The Elara Handbook

The Project Elara contributors

March 16, 2026

Contents

0.1 The basics

0.1.1 A basic guide to Project Elara

In this chapter, we will introduce Project Elara from its very basics. You'll learn the details of what it's about, why we do it, and *how* we do what we do. In addition, we will go over the general math and physics to understand the Project's research. By reading through this chapter, you will have taken the first step along the Project Elara journey. We hope it will be a very fulfilling one.

Project Elara overview

Project Elara is a broad effort to to push frontiers in science and engineering and make world-changing technologies possible. We have a research program covering a wide variety of areas, but our current focus is on **space-based solar power**, a technology that promises to deliver sustainable energy to nearly everywhere around the world. It is a very ambitious and challenging task, but it is one we choose to tackle, because we believe that developing the technology that can create an energy-resilient world, freed from power scarcity and inequity, is more than worth the effort.

Project Elara actively supports open-access research, in which we freely exchange our knowledge and expertise. Our team believes in the strength of collaboration, and is keen to participate in research partnerships with other groups and institutions. We hope to form strong alliances and bring together teams from around the world, to build a future worthy of our future generations.

> "Some see things as they are and ask why; [We] dream things that never were, and say why not."
> **George Bernard Shaw**

A tourist's guide to the project We are a research organization based in Rensselaer Polytechnic institute, focused on developing **space-based solar power technology**. This technology collects energy from our Sun in space and beams it back to Earth. When fully-realized, this technology would generate power orders of magnitude above nearly any means of power generation we could conceive of, even above nuclear fission and fusion. This technology could very well **power the world one day** with clean energy, one that can last for hundreds of thousands of years, and one that protects the Earth's natural environment and keeps it pristine. It also allows us to send that energy to **nearly anywhere on Earth** in essentially **all weather conditions**, allowing us to provide power to the people that **need it most**. That means providing power to hospitals cut off from electricity, so their emergency wards can keep on running; providing power to rural schools, so kids stay warm inside in freezing winters; and providing power to emergency workers in disaster zones, so their equipment can operate around the clock and help save countless lives.

Further, the same technology used in space-based solar power could pave the way to creating a true spacefaring civilization, and achieving **interplanetary and interstellar spaceflight** with lightsail-equipped starships. Realizing these capabilities would protect our future generations, ensuring that even if disaster strikes on Earth, they can find a home among the stars. And as a power source with immense energy-generation capabilities, it could even pave the way for highly-advanced technologies that are likely centuries away, including advanced interstellar propulsion and possibly even spacetime engineering. At the same time, closer to home, it could bring about improvements in fields as far-ranging as laser surgery, cancer treatment, and gravitational-wave research, due to its broad technology-transfer potential. For all these reasons, and more, we believe that space-based solar power technology is not only *beneficial* to develop, but *essential* to develop for securing our future.

As part of our pledge to future generations, we are committed into ensuring our technology and research will remain openly-available to everyone, forever. The entire project is open-source, and many components are dedicated to the public domain, meaning they are copyright-free and patent-free. In addition, we will make sure that the services we provide are as low-cost as possible, and eventually, can be given completely for free.

We realize that working on research and technology this ambitious is not possible to complete within any short time frame, especially for an organization determined to share its technologies without regard for profit. For this reason, we have written this Handbook to document our work, so that others can easily pick up our work from where we left off. We hope that our vision of a hopeful future inspires solidarity and unity that will act as a positive force against cynicism and division around the world. Whether we succeed or not, we believe that our work can inspire others to carry forward our ideals and our mission, and that, alone, is worth fighting for.

> "Give me a lever long enough and a fulcrum on which to place it, and I shall move the world."
> **Archimedes**

What is Elara? In a constructed language devised by the Project's founder, *Elara* (anglicized: *elara*) is the word for *hope*. Project Elara is a choice to have hope, despite knowing of the nearly impossible odds in succeeding in our mission. We hope that it can represent the best of us, the good we can accomplish when we put our efforts towards the good of the entire world.

> "Alone, we can do so little; together, we can do so much."
> **Helen Keller**

Open Philosophy

> "As we enjoy great advantages from the inventions of others, we should be glad of an opportunity to serve others by any invention of ours, and this we should do freely and generously."
> **Benjamin Franklin**

Project Elara's mission is one that cannot be accomplished without a philosophy of openness. After all, how would there be much benefit from our work if its knowledge is confined to only a few individuals, or only available for exorbitant sums? Rather, science should be open and should benefit everyone. For this reason, we release our work under the principles of **open science** and **open knowledge**:

- The project is a collaborative **open effort**, with contributions from all being accepted
- All code written as part of the project is **open-source** and **dedicated to the public domain**
- All creative resources of the project are licensed under the **CC0 public domain license**
- All hardware produced in the project will be **open hardware** with documentation and design specifications freely available

By dedicating Project Elara to the public domain, we have greater confidence that no one can take control of the technology for themselves, and everyone will be ensured free and unlimited access to the research.

An ethical research approach It is an unfortunate that while the sciences are meant to be a quest to push the bounds of human understanding and knowledge, it easily becomes a chase for fame and recognition. We are *not* the only team working on space-based solar and related high-precision beam focusing; among the many other teams include a brilliant team from Caltech, the LISA team working on fantastic laser interferometery, and researchers at the OTPS at NASA (the list is most certainly non-exhaustive and too short to list the contributions of every team and scientist).

Another common side-effect of the ills of scientific competition is the omission of credit to those who made a discovery or an innovation possible. The individual egos of scientists (or nations funding the work of scientists) makes it tempting to downplay the work of others - especially of scientists who came from disadvantaged backgrounds - and make it seem like it was the work of a single individual or group. To us, this is **unacceptable**.

At Project Elara, we have **no interest** in competing with any other groups working on the same technology. To the highest extent we possibly can, we will freely extend offers of collaboration, freely share our research and technological breakthroughs (the Project is structured around this principle) and assist our colleagues in whichever way we can. When we use the work of others, this **must** be acknowledged as their work and not our own work. In doing so, we hope to build others up instead of tearing others down, because, to put down *even one scientist* amounts to a gigantic loss to the field as well as a human trajedy.

In addition, we **never** want to start a priority dispute nor pass over the contributions of any scientist. Internally, we have a firm policy that **fair credit must be given to all who make contributions within the Project**. In our "Credits" section in the Appendix of the Handbook, we have a list that attempts to chronicle every contribution to the Project. This list is certainly imperfect - realistic constraints mean it will never be fully accurate nor exhaustive - but we hope that this builds trust in our team in that no contribution is forgotten.

Our standards regarding AI-based tools In our quest for scientific identity, we cannot afford to not consider the circumstances of this new era (at the time of writing) of the widespread use of AI-based tools. These tools are powerful, and we do indeed use these tools in the course of our research: to learn, discover new ideas, and investigate new approaches to solving problems. We may even ask these tools to suggest ideas for how to solve a particular physical or engineering problem.

But crucially, there are several guidelines that we need to follow in doing so. First, *we must do our homework.* After learning something new from an AI-based tool, it is imperative that we **research it, learn about it,** and do our best to **understand it** and record down that knowledge in this

Handbook for posterity. Simply taking ideas from an AI-based tool as if it were our own work, or copy-and-pasting its responses verbatim, is **unacceptable**.

Second, *every* idea that was **not novel**, whether it came from an AI-based tool or from another scientist's (or team's) paper or article must be traced back to its source and credited, as far as is realistically possible. After all, AI-generated tools rarely make something *new* as much as piece together information that was fed to them during the training process. The ideas came from another scientist (or team of researchers). While it is not always possible to either automatically or manually find the sources used by an AI-based tool in one of its responses, we will do our best to try.

And third, we are well-aware of the ethical implications of using an AI-based tool. We see the exploitation of human beings by leading AI companies such as OpenAI as well as the widespread abuses committed by prominent members of such companies as **unacceptable**. While we recognize the convenience of using their technology, we have **no interest** in supporting their abuses of power and people by continuing to use their technology (even if their services are nominally free). We instead highly encourage each of our members **to use AI models that we train ourselves, or not at all**. We also highly encourage using our Elara ML machine-learning library and other tools we build, as these libraries were created with our ideals in mind, rather than to serve (and be abused by) the interests of a corporate entity.

Scientific integrity While it is essential that science is performed in a manner that gives honest results - even if those results aren't necessarily groundbreaking - this is very often tossed aside out of personal greed. Faking results, attempting to inflate poor results, or data manipulation to support a certain claim are common violations of scientific integrity. The highly-competitive research environment, often characterized by the term *publish-or-perish*, exacerbates and encourages this (mis)behavior. We are aware that simply because we stand for high ideals **does not make us immune to this behavior**, and thus we must take steps to avoid it to the furthest extent possible.

First, just as we mentioned earlier, we want to work together with other scientists working on the same (or similar) things so that we are not trying to gain a "competitive edge", so to speak, over another group of scientists. We will extend out offers of collaboration in good faith and willingly encourage the participation of others. In this way, we can mitigate one of the big drivers of scientific dishonesty - the fear that someone will "get to it first".

Second, we will identify any flaws we can find in our research and continually test and re-test, fixing any issues we find and noting this process, before publishing. Yes, realistic considerations may force us to shorten this process, but we will always aim to perform an internal review process. We will also encourage **mentioning null results or unexpected results**. While others may regard it as embarassing, we believe that sharing such results allows others to not repeat a flawed process we used and to know the difficulties and the dead-ends of a particular research pathway.

At the end of the day, we are imperfect human beings who make mistakes. However, we belive that establishing these guidelines is our best hope of preventing the most egregious ethical violations and encouraging high scientific ethical standards and paving the way for a future based in collaboration and fellowship for humankind.

Guiding Principles

At Project Elara, our mission is not purely scientific research. It is to pave the path to a better world for all humanity, to give future generations something to hope for. At present, we may be small and our research is only in its infancy. But we want to build a firm foundation of **respect for fundamental principles** we strive to uphold.

For this reason, we have an official **Project Elara Charter**, the fundamental instrument that establishes, regulates, and is the source of supreme authority over the Project. While the Project has leaders, its leaders do **not** have authority on their own regard, but only on that which is granted to them by the Charter for the period the Charter allows. At the time of writing, the Charter is provisional, as the formal process of ratification does require time, but one day, the Charter will be ratified, at which point it will be **enforced**.

While the Charter itself is rather lengthy and technical, and not an exceptionally fun read, we believe it is important to outline its general principles. We will elaborate on several important principles encoded in (or will eventually be in) the Charter in the following paragraphs.

> **Note**
> Much of what is discussed within this section is difficult to enforce at the present stage of the Project. We ask that they be considered **provisional** at present - that is, they are subject to change - although we will still not change them lightly. Upon the promogulation and ratification of the Elara Charter, these standards will be fixed, at which point they **cannot** be changed except through the formal amendment process.

Cost-free and universally-available technology Putting up any paywalls, charging for our software, or hiding source code behind proprietary walls is not our way. All of the source code relating to Project Elara will be open-sourced and provided freely to anyone who wishes to obtain a copy of it. The project's source code may **not** be re-licensed under a nonfree software license. Financial considerations will not be taken into account, and all technologies developed directly or indirectly from the project will be unpatented, to ensure that the project remains universally open and accessible, because improving others' lives shouldn't be motivated by money.

In addition, we will freely share our technology and openly disclose our research advances. We will not hide anything to prevent someone else from "doing it first" - we want *everyone* to succeed and we will collaborate as much as we can. We want to unite people together instead of perpetuating divisions caused by toxic competition. The crux of our position is this: if we at Project Elara have a good idea, why **not** share it, so that we can all move forward and collaborate together to achieve what we want to achieve? In this way, instead of having one "winner" and countless "losers", we **all win** and we **all benefit**.

Leading by conscience One may ask, if our software is completely free and open-source, what prevents malicious individuals and bad actors from using it to harm people? We are well-aware of this issue and have given it much consideration. Our answer is that while we cannot *stop* others from abusing our software - although realistically, even a well-intended license to prevent misuse would do little against those who are determined to perpetuate horrific crimes - we *can* stand up against them. We will use every peaceful means available to us to do so, including petition and protest alongside others, as well as harsh condemnations in our public addresses and assisting in efforts to bring said malicious individuals and bad actors to justice.

In addition, we are against any organization that attempts to turn our software into spyware, to terrorize, maim, and kill human beings, or to inflict environmental devastation. We collect **no telemetry** and try to design our software to work offline-first. While we use GitHub for code hosting, we have plans to move to something more secure and self-hosted, like Gitea, and use GitHub purely as a mirror (that is, a synchronized read-only copy of our code/handbook).

Finally, we **will not** seek to profit from this project in any way, and this extends to gaining fame or recognition. We further encourage (though do not mandate) that any awards and recognitions be

given to the **entire Project Elara organization** instead of a single member. We hope for us to do what we do out of the noble purpose for which it was intended, rather than as a personal avenue for success and fame.

Open to all, benefit all The project is open to contributions - that is, pull requests, participation in the organization, collaborations, or research partnerships - from anyone, with very few exceptions. We want to open our gates and invite everyone. Discrimmination within the Project is taken seriously and violators will have consequences. If contributions are taken, credit will be given where it is due. No one will be allowed to claim singular or substantial credit.

The innovations brought by the project will not be left in the hands of a powerful few. The innovations will belong to everyone. The primary motivation for the project will not be for profit, but instead, to progress our civilization forward and leave a better world for our future generations. We want to make sure it stays this way.

No outside influences We have already emphasized that the project should not fall into the hands of one specific person, nation, or company. The project will not accept monetary funding from persons or groups seeking to influence the project, nor conduct *quid pro quo* deals where funding or assistance is exchanged for political influence over the project. We will take a cautious approach: if we have suspicions of some donor or backer, we will launch an investigation and we will **reject** the funding or assistance by default, unless we have **confidence beyond a reasonable doubt** that the project will not be compromised. The same goes for potential conflicts of interest.

We ask that the project **never be sold or disbanded**, no matter how lucrative the fees or pressures put on us. Individual members may enter and leave the Project at will, but the Project itself **must** always stay and **will not** be merged into another organization.

Staying above politics Project Elara is meant to be an organization that stands for the advancement of all peoples and improving the lives of all peoples across the globe, without showing preferential treatment towards any individual, group, nationality, or organization. Being dragged into politics is an existential threat to the Project's independence and gravely endangers trust and confidence in the Project. For this reason, we ask that our individual members express any political views, sentiments, or support for political movements **separately from the Project**, and not try to bring politics into the Project. While politics may be discussed in informal conversations, this must always be in a **non-official capacity**. The Project must avoid morphing into a political entity, which means that we will **not** endorse political candidates, fund politicians, or ally with a particular political party, or conduct political activities within the Project. In addition, we **will not** promote a particular political position, and we **do not** choose sides in a political debate. We are **not associated** with any particular government, any particular domestic or foreign policy, or any particular social movement, nor will endorse any such group or movement. We are to be true to our Charter and our Charter alone, and we are a **politically-independent** organization.

Full accountability and transparency Even with all of these rules in place, the skeptic may wonder how we will hold ourselves to account when we have to self-enforce. We are reminded of the famous saying, *"who will guard the guards?"*, originating from the Roman poet Juvenal.

For this reason, we operate an internal whistleblower protection program in which the identities of whistleblowers who report a violation of our rules will be kept anonymous. Whistleblowers may email elaraproject.sci@gmail.com using a throwaway email such as https://temp-mail.org/en/ to inform us, attach any important information (such as screenshots, conversations, pictures, and so forth). Further, any whistleblowers who publicly speak out **must not be harmed in any way** and we will make arrangements to see that this is enforced.

Project organization

We always believe that our mission at Project Elara is **first and foremost** our concern for humanity and determination in building a hopeful future for our descendants, not one that can be reduced to pure research or technology. However, there is no denying that there is *a lot* of technology and research as well as logistics involved in carrying out our mission. Project Elara is a complex project, with many, many components. We will go over each of these components.

The Elara Handbook The Handbook is perhaps the most recognizable component of Project Elara - after all, you are reading it right now! It serves as Project Elara's central documentation and reference work and should be regarded as the most up-to-date reference for the Project. In addition, it serves as a guided learning resource, and we strongly recommend reading it for new members. We intend it to be as comprehensive as possible - it is well over 100 pages long at the moment of writing, so do not expect you need to read all of it!

The Elara Codeberg Beyond simply documentation, we also need a safe place to store our code, software, and hardware. This is where the Project Elara Codeberg comes in. Codeberg is an open-source code hosting platform developed by a nonprofit, unlike GitHub, which is the industry-standard; we have chosen Codeberg since we trust it to keep our code safe and is free from corporate influence. It is also where we coordinate our work, have technical discussions, and backup our files.

The Elara Community Last but not least, Project Elara is a team! We have active discussions on our Discord server and prospective members can fill out the new members form to quickly get started with the Project. Lastly, we have a newsletter on Substack and YouTube channel that provide coverage of our work and raise awareness of our mission.

The Handbook's purpose

Before we move into the mathematics and physics that will form the bulk of the rest of the chapter, we feel it is important to discuss *what* exactly this Handbook is for. This is because the Handbook is not only a book - it is an integral part of Project Elara.

First, the Handbook is our **learning guide**. In our belief that all knowledge should be free and accessible to anyone, we put everything we know into the Handbook, teaching even the most technical topics in a step-by-step fashion. It should be conceivably possible for anyone to self-learn everything about the project from minimal basics using this Handbook. A future goal is that this Handbook will be translated into multiple languages, so anyone can read this Handbook in their native language.

This is why it contains a multitude of topics, making it more of a small library than one book. The currently-complete sections are primarily math and science-focused. We hope that with more active development, errors and issues in the Handbook can be quickly fixed, ensuring accuracy. As a (somewhat intentional) side-effect this also means that the Elara Handbook can be used as an (unconventional) physics and engineering textbook or complement to a self-learning course in a STEM field.

Second, the Handbook is our **avenue to share our research**. We present our research in the Handbook, and in-progress research as well, again, out of our belief in free and open knowledge. The Handbook is not meant to be a place for research papers, but for long and detailed explanations of our research that would not fit in a research paper (and that honestly, we'd prefer instead of writing papers). We want our research to be as fully-accessible as possible, but research papers inherently force research to be condensed and often difficult-to-read, so the Handbook is where we put the full details and explain things in a gentler fashion.

In pursuit of this goal, the Handbook aspires to follow a very specific style. It seeks to use simple, unambiguous language, and never assume something is "obvious" to the reader. In addition, when complete, it is meant to include many, many fully worked-out practice problems to reinforce understanding.

> **Note**
> You may notice that this is not necessarily the current style! Indeed, we have much to work on, but due to inherent time constraints, we've had to rush through a lot in the Handbook. This is a *temporary situation* and will be rectified as soon as we are able to.

Finally, the Handbook has one unique purpose that is perhaps its most important: a **safeguard for the project**. It holds the Project's collective knowledge, experience, and preserves the Project from the tumultuous turns of the world. That is, no matter what disasters may happen to the project itself, copies of the Handbook will allow it to live on, and the knowledge contained within is preserved for future generations. Even if somehow the Project is destroyed, anyone should be able to self-learn from the Handbook and rebuilt the project from scratch if need be.

The Handbook is specifically designed to be able to be viewed in multiple formats, including online, offline on a local web server, as plaintext markdown source, in PDF format, and (eventually) printed paperback and hardcover books: multiple formats prevent data loss and are better for long-term preservation of information. We will not chase after the fanciest new formats in this digital age; we will stick with what is proven to last.

What you are reading right now is not just a rather long book on idealistic technology, but a book truly intended to be a gift for all future generations: an archive for humanity.

How to get started in the project

Project Elara is not the work of one person, but the work of a team - and we welcome anyone who wants to work with us to join us. If you're new to the Project, or interested in participating in it, this part of the Handbook is for you!

If you're just starting with us, we want to make sure that joining Project Elara is as seamless as possible. And we know it can be quite confusing to start work on any research or open-source project, with bewildering documentation and a lack of a clear beginner's guide to follow. This is a step-by-step guide to quickly get up to speed and becoming part of Elara's mission!

Step 1: Get in touch First, we want to make sure that you can get in touch with us. This is especially important so that we can explain things and troubleshoot in case you run into any trouble with the onboarding process! If you're on Discord, you can join our Discord server by copying our server invite link https://discord.gg/Zr37GyxzDd into your web browser. You can also send an email to either one of the address below:

- elaraproject.sci@gmail.com, which is our official organization email (this is the recommended email for official inquiries)
- songy14@rpi.edu, which is the email of Jacky Song, our organization head and chief administrator (this is the **recommended email for fast responses**)

Step 2: Understand the basic onboarding tasks Before working on the Project, we want to get to know you a little more, and make sure you have access to the main platforms we use below.

New Member Form: There is a New Member Form available to anyone who wants to join. This simply is used so that we can get to know you a little more, and we thank you for providing a little bit about yourself!

Sign up for Codeberg: Project Elara operates much of its database on Codeberg, a repository of all our progress so far. We require that members set up an account so that they have easy access to our work. Start by using the link provided - make sure that you use an email that you feel comfortable with being publicly-visible.

Installing Git: You can also check out our Git Reference Guide, an introduction to the work we upload on Git.

Signing the Licensing Policy: Lastly, we have a Licensing Policy that we would like members to look at. Not only does it help you understand our values and goals, but it also helps us in knowing that you are well-committed and understand our policies.

Signing the Charter (Optional): Lastly, there is a Project Charter that is open to signing, but is completely optional. If you are interested, feel free to read through the link above.

Step 3: Understand our roles and ideals Second, Project Elara is part of RCOS, Rensselaer's Center for Open Source. Whether you are a student currently attending Renssealer Polytechnic Institute (RPI) contributing to the Project for credit, or if you are simply interested in working outside of RCOS, we have a spot for you!

Understand that Project Elara is open to all experience levels - in fact, you don't even need to have experience working on open-source research or have a strong grasp in the type of work we do. Other than our more hands-on, technical, and mathematical aspects, we are also open to article writing, music composition, digital/conceptual design, and uploading content pertaining to our progress. Regardless of how you want to contribute to the Project and the amount of experience you have, we are more than ready to welcome you, helping you settle in and adjust to work that you enjoy doing.

Step 4: Familiarize yourself with general information about the Project Of course, we want to make sure you know what we do, and what our mission is! First, make sure you read our mission statement, which we've put down below:

> **Our mission statement**
> Project Elara is a nonprofit dedicated to being part of creating a future powered by plentiful space-based solar power. We conduct open-source research dedicated to the public domain that is not driven by profit, because our singular goal is to help create the hopeful future we envision. We puvlish our work openly and explain it in detail, so that even if we don't succeed, others can pick up where we left off and being our work to completion.

We believe in **open science** = science without paywalls, done for the sake of advancing science and public good. And we believe in the value of **open source** - releasing our code and hardware into the public domain, making sure it's available to everyone, and that no one will be able to establish a monopoly for its use.

In terms of the platofmrs we use, we've provided brief descriptions of each below.

As mentioned above, Codeberg is our main repository of work and progress, including commits, tasks, trackers, and more. We are also on OpenCollective, where we keep track of our purchases for materials meant for practice and experimentation. Additionally, we are on Weblate, where we keep track of the langauges that have been translated completely for our website so far. Lastly, we use Substack for article writing and personal stories, in the hopes of raising awareness for the Project and also allowing opportunities for members to express their creative freedom via writing and peer review.

Last but not least, we believe in working as a team, treating each other with kindness and understanding, and making sure everyone feels comfortable and safe within our team. A big part of this is that we carefully track contributions so **everyone gets credit where credit is due**. Working with us is contingent upon you affirming these ideals. If you choose to espouse, disseminate, or take action based on ideas that are hostile to our mission, you will **not** be allowed to further participate in our work. We will trust that you will act in good faith - please do not break that trust.

Optional: Gain a basic understanding of our research We also want to make sure you have a basic grasp of our research and our design! For this, we *highly* recommend watching our trailer aimed at anyone new to the project.

Reading the first chapter of this Handbook is also recommended, but feel free to skip that if you don't have the time.

Step 5: Pursue your interests Having read our mission statement and our introductory poster, it's time to choose the role(s) you want to take in the Project! We want to make sure everyone has a role that fits them. At the moment, we have four (major) divisions open to new members:

- The research division, which focuses on physics and engineering analysis of space-based power systems using analytical and computational methods
- The build division, which focuses on construction and testing of power system prototypes, with plenty of hands-on work!
- The community & outreach division, which focuses engaging with the community, helping out, and spreading the awareness of Project Elara and encouraging more people to join us. This is especially suited to anyone with experience or interest in the visual arts (in particular digital art, photography, film production, motion graphics, and graphic design), volunteering, fundraising, or public speaking.
- The developer division, which focuses on developing and maintaining our open-source software, documentation, and web content (including our website). Experience in physics/engineering is a bonus, but not required. If you're a developer, this might be the place for you!

> **Note**
> In this Handbook, we explain (much of) the physics, engineering, and math involved in our work, and we require no background in any of our areas of work, though experience is of course welcomed. We should note that the research-division is the most physics/math-heavy, the build division is much less so, and the outreach division has minimal scientific/math-based work.

We strongly recommend exploring our roles document, which details a list of roles within the Project. You can choose more than one role, but please don't take more than you can handle. Please also make sure that you let us know how you'd like to keep in touch with us, so that we can work together more easily - again, our work is highly collaborative in nature. If you try a role and find that you don't like it, that's okay! Just let us know, we are happy to help you find something you enjoy doing!

Step 6: Next steps Once you've done all the previous steps, you're all set to start working with us! What to do next depends on which division you wish to join. We'll go through the things to do for each of our divisions, one-by-one.

If you're planning to join the build division or research division, we post tasks to work on in our official work repository, Elara Labs, or our general project tracker. Just let us know which task(s) you'd like to work on by adding a comment under the task(s): you can take whichever tasks interest you! In each task, we list out the work description in detail, as well as relevant sections in the Handbook for any background knowledge required.

If you're planning to join the developer division, we post tasks to work on in specific issues for our repositories, such as the issue trackers for `elara-math`, `elara-gfx`, and `elara-array`. You are also welcome to work on improving this Handbook itself! Note that not all of these are programming/code-based, so if you're planning to work as a developer, just focus on the ones that are to your liking.

If you're planning to join the community & outreach division, please get in touch with us, either by emailing elaraproject.sci@gmail.com or our Discord (via the invite link mentioned previously). We'll guide you through participating in the community and help you find a way to contribute.

0.1.2 Introductory mathematics

Up to this point, we've gone through the details of the Project and explained its work in broad terms. But to gain a *deep understanding* of how everything works, we do have to use a reasonable amount of mathematics (in particular, calculus) and physics, which, as you'll soon see, are very closely-related fields. We will start everything at basic algebra, and build up to calculus-based physics. These can be difficult topics; take it at your own pace. We hope it will be an enjoyable read.

Mathematical notation

Unless otherwise indicated, mathematical symbols will be represented by the following notational conventions:

Function: $f(x)$
Function composition: $f(g(x))$
Limit: $\lim_{x \to x_0} f(x)$
Vector quantity: $\mathbf{E}$ or $\vec{E}$
Derivative: $\dfrac{df}{dx}$ (preferred), $f'(x)$ (alternative); time derivative only: $\dot{x} = \dfrac{df}{dt}, \ddot{x} = \dfrac{d^2f}{dt^2}$.
Nth-derivative: $\dfrac{d^nf}{dx^n}$ (preferred), $f^{(n)}(x)$ (alternative)
Derivative operator: $\dfrac{d}{dx}$
Partial derivative: $\dfrac{\partial f}{\partial x}$
Partial derivative operator: $\dfrac{\partial}{\partial x}$
Gradient: ∇f
Divergence: $\nabla \cdot \mathbf{F}$
Curl: $\nabla \times \mathbf{F}$
Laplacian: $\nabla^2 f$
Integrals:

Integral type	Symbol	Alternative notation
Indefinite integral	$\int_0^x f(k)\ dk$	Integral without bounds (but not recommended)
Definite integral	$\int_a^b f(x)dx$	Limits can be placed directly on integral sign
Line integral (scalar)	$\int_C f(x,y,z)d\ell$	Yes, use dr or ds as differential
Closed line integral (scalar)	$\oint_C f(x,y,z)d\ell$	Yes, use dr or ds as differential
Line integral (vector)	$\int_C \mathbf{F}\cdot\mathbf{d}\ell$	Yes, use $\mathbf{dr}$ or $\mathbf{ds}$ as differential
Closed line integral (vector)	$\oint_C \mathbf{F}\cdot\mathbf{d}\ell$	Yes, use $\mathbf{dr}$ or $\mathbf{ds}$ as differential
Surface integral (scalar)	$\iint_\Sigma f(x,y,z)\ dS$	Yes, $\int_\Sigma f(x,y,z)\ dS$
Closed surface integral (scalar)	${}_\Sigma f(x,y,z)\ dS$	Yes, $\oint_\Sigma f(x,y,z)\ dS$
Surface integral (vector)	$\iint_\Sigma \mathbf{F}\cdot\mathbf{dS}$	Yes, $\int_\Sigma \mathbf{F}\cdot\mathbf{dS}$
Closed surface integral (vector)	${}_\Sigma\mathbf{F}\cdot\mathbf{dS}$	Yes, $\oint_\Sigma \mathbf{F}\cdot\mathbf{dS}$
Double integral	$\iint_R f(x,y)\ dA$	Not recommended
Area integral	$\iint_R dA$	Not recommended
Triple integral	$\iiint_\Omega f(x,y,z)\ dV$	Not recommended
Volume integral	$\iiint_\Omega dV$	Not recommended
Spacetime integral	$\int_M \sqrt{-g}\ d^4x$	None

Note
For all multivariable integrals, the precise subscript of the integral, whether C or Ω or M or Σ, isn't really that important. A subscript should be placed but should also be elaborated on in the text describing the integral. It is the integrating differential that is most important.

Algebra

We will begin by going through a review of basic algebra. If seems too easy, feel free to skip to the next section; otherwise, continue here.

We will assume you know what negative numbers and fractions are, and how to do basic arithmetic. If not, review at https://www.khanacademy.org/math/pre-algebra.

The fundamentals of algebra Algebra is a system of mathematics where we do computations using symbols rather than numbers. Symbols are usually denoted with roman or greek letters.

For instance, consider the following equation:

$$\square - 3 = 5 \tag{1}$$

It's clear that the "missing number" in the box has to be 8 for the equation to be true. We can now replace the box with a symbol called x:

$$x - 3 = 5 \tag{2}$$

Where:

$$x = 8 \tag{3}$$

The rules of algebra Every unknown in algebra is denoted by a **different symbol** such as x, y, z, etc. For example, we could write:

$$x + y = 3 \tag{4}$$

Which would be true if $x = 1$ and $y = 2$.

Multiplication in algebra is written like this:

$$x \times 3 = 3x = x + x + x \tag{5}$$

This means that:

$$1x = x \times 1 = x \tag{6}$$

And:

$$0x = x \times 0 = 0 \tag{7}$$

Division in algebra is almost always written as a fraction:

$$\frac{x}{4} = x \div 4 \tag{8}$$

We can move the x to the left of a fraction and it would be equivalent:

$$\frac{1}{4}x = \frac{x}{4} \tag{9}$$

If we multiply a fraction by the same value as the bottom of the fraction, we get rid of the fraction:

$$\frac{a}{b} \times b = a \tag{10}$$

If we have the same thing on the top and on the bottom of the fraction, we can remove the same thing - this is called "cancelling":

$$\frac{ax}{ay} = \frac{x}{y} \tag{11}$$

The fraction of anything over one is itself:

$$\frac{a}{1} = a \tag{12}$$

We can move the bottom part of a fraction out of the fraction to get rid of the fraction:

$$x = \frac{a}{b} \Rightarrow xb = a \tag{13}$$

We can have negative symbols as well as positive symbols:

$$-x = -1 \times x \tag{14}$$

A negative sign to the left of a fraction is equal to a negative numerator or a negative denominator:

$$-\frac{2}{5} = \frac{-2}{5} = \frac{2}{-5} \tag{15}$$

Algebraic expressions are collections of numbers and symbols:

$$3x^2 + 5xy + 6 \tag{16}$$

Equations are expressions that are related by an equal sign:

$$xyz = 5 \tag{17}$$

Parts of algebraic expressions and equations separated by operators $(+ - \times \div)$ or placed inside brackets are called **terms**. For example, $5x + 3y + 2z = 5$ has the terms $5x$, $3y$ and $2z$.

If we change the order of an equation, the equation remains the same:

$$x + 1 = 1 + x \tag{18}$$

Brackets are equivalent to multiplication when a number or symbol is placed to the left of a bracket:

$$3(x + 1) = 3 \times (x + 1) \tag{19}$$

We can **expand** brackets by multiplying the number (or symbol) in front of the bracket by everything inside the bracket:

$$3(x + 1) = 3 \times x + 3 \times 1 = 3x + 3 \tag{20}$$

And in the same way, we can **factorize** an expression into one that has brackets:

$$ab + ac = a(b + c) \tag{21}$$

For more complicated brackets expansions, we use this formula:

$$(a + b)(c + d) = ac + ad + bc + bd \tag{22}$$

To factorize this, we'll need a tool called the quadratic formula, which we'll explore later in this chapter.

We can write powers like this:

$$x^3 = x \times x \times x \tag{23}$$

Where x^3 is equal to x multiplied by x 3 times.

Putting a negative sign in front of brackets is equal to multiplying everything inside the brackets by -1:

$$-(a + b) = -a + -b \tag{24}$$

A double negative is a positive:

$$-(-a) = a \tag{25}$$

Sometimes, we use symbols to represent a number rather than an unknown - these are called **constants**. To avoid confusing symbols that represent numbers and symbols that represent unknowns, we usually use special letters (especially greek letters) to denote constants, and we will always say beforehand that the expression or equation contains a constant. For example, a common constant is π, which is a number that starts with the digits 3.14159. This means that:

$$\pi x = 3.14159 \times x \tag{26}$$

Solving algebraic equations To be able to solve algebraic equations, the key rule is that **we do to the left-hand side of the equals sign as we do to the right**.

For example, take the example:

$$x + 5 = 10 - x \tag{27}$$

Here, we first need to add x to both sides of the equation, giving us:

$$x + 5(+x) = 10 - x(+x) \tag{28}$$

This will become:

$$x + x + 5 = 10 \tag{29}$$

Which simplifies to:

$$2x + 5 = 10 \tag{30}$$

We then subtract 5 from both sides of the equation:

$$2x = 5 \tag{31}$$

And then divide both sides by 2 to get the answer:

$$\frac{2x}{2} = \frac{5}{2} \tag{32}$$

$$\frac{2}{2}x = \frac{5}{2} \tag{33}$$

Because $\frac{2}{2} = 1$:

$$x = \frac{5}{2} \tag{34}$$

Sometimes, we need to solve an equation that uses entirely symbols. Yes, that can seem scary, but going slowly, every step becomes easier. For example, presume we have the equation:

$$\frac{p(a + 2x)}{q^2} = b \tag{35}$$

The first step is to multiply by q^2 on both sides:

$$\frac{p(a + 2x)}{q^2} \times q^2 = b \times q^2 \tag{36}$$

Since we know that we can remove the $\times$ sign and just write symbols next to each other to show multiplication, this becomes:

$$\frac{p(a + 2x)}{q^2} q^2 = bq^2 \tag{37}$$

We use one of the rules of algebra, which says that $\frac{a}{b} \times b = a$, to simplify:

$$\frac{p(a+2x)}{q^2} \times q^2 = p(a+2x) \tag{38}$$

This means our original equation becomes:

$$p(a+2x) = bq^2 \tag{39}$$

Now, we need to divide p from both sides. Recall that we write divisions as fractions, so we have:

$$\frac{p(a+2x)}{p} = \frac{bq^2}{p} \tag{40}$$

Using the rule that $\frac{px}{p} = x$, we can cancel the p:

$$(a+2x) = \frac{bq^2}{p} \tag{41}$$

Now, we can subtract a from both sides to get rid of it:

$$(a+2x) - a = \frac{bq^2}{p} - a \tag{42}$$

Which gives:

$$2x = \frac{bq^2}{p} - a \tag{43}$$

We still unfortunately have the 2, which is more annoying to deal with. We have to divide the 2 from both sides:

$$\frac{2x}{2} = \frac{\frac{bq^2}{p} - a}{2} \tag{44}$$

Because $\frac{2x}{2} = x$ (it makes sense; multiplying something by a number then diving by that same number should give you that original something), we can simplify to:

$$x = \frac{\frac{bq^2}{p} - a}{2} \tag{45}$$

Remember that anything multiplied by 1 is itself:

$$x = \frac{1 \times \left(\frac{bq^2}{p} - a\right)}{2} \tag{46}$$

And that we can move parts of a fraction off the fraction:

$$x = \frac{1}{2}\left(\frac{bq^2}{p} - a\right) \tag{47}$$

Congratulations, we've solved it!

Graphs and coordinates A graph is a visual way of representing data. To plot a graph, need two values: an x value, and a y value. We represent these values as **ordered pairs** like this:

$$(x, y) \tag{48}$$

For example, if we want to plot the value $x = 1, y = 3$, then $(x, y) = (1, 3)$. We can plot this on a graph by traveling 1 unit to the right, then 3 units up, like this:

```
import matplotlib.pyplot as plt
import numpy as np
# matplotlib -specific customizations
%matplotlib inline
plt.rcParams["font.family"] = "serif"
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams["axes.grid"] = True

plt.plot(1, 3, "ro")
plt.title("The point $(1, 3)$")
plt.xlim(0, 4)
plt.ylim(0, 4)
plt.show()
```

Functions Functions take in a number and output another number. We denote a function with a symbol with the input as another symbol - for instance, a function f with the input x is written as $f(x)$. A simple function could be:

$$f(x) = 2x \tag{49}$$

This means that any input x value would have an output value of twice that x value. We can make a table for values of this function:

x (input)	$f(x)$ (output)
0	0
1	2
2	4
3	6
4	8
5	10

We can visualize this table of values by plotting it as (x, y) pairs, where $y = f(x)$:

```
x = np.linspace(0, 5)
plt.plot(x, 2 * x)
plt.title("$f(x) = 2x$")
plt.show()
```

Since we plot the value of $f(x)$ on the y-axis, we say that $y = f(x)$. This means we could've called the above function $f(x) = 2x$, or $y = 2x$ - it doesn't really matter.

Polynomial functions Linear functions are in the form $f(x) = ax+b$, such as $f(x) = 5x$, where a and b can be equal to zero. They are called linear because they are "line-like" - they are straight lines. We've already seen this.

Quadratic functions are in the form $f(x) = ax^2 + bx + c$, such as $f(x) = x^2 + 2x + 3$, where b and c can be equal to 0. They look like this:

```
x = np.linspace( -5, 5)
plt.plot(x, x ** 2)
plt.title("$f(x) = x^2$")
plt.show()
```

We can solve any quadratic function in the form $ax^2 + bx + c = 0$ using the quadratic formula:

$$x = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \text{ and } x = \frac{b + \sqrt{b^2 - 4ac}}{2a} \tag{50}$$

Cubic functions are similar, but they are just in the form $f(x) = ax^3 + bx^2 + cx + d$, such as $f(x) = 3x^3 + 5x^2 + 4x + 6$. They look like this:

```
x = np.linspace( -5, 5)
plt.plot(x, x ** 3 + 3 * x ** 2)
plt.title("$f(x) = x^3 + 3x^2$")
plt.show()
```

The absolute value function $f(x) = |x|$ takes in any negative or positive value and returns a positive value. It looks like this:

```
x = np.linspace( -5, 5)
plt.plot(x, np.abs(x))
plt.title("$f(x) = |x|$")
plt.show()
```

Rational functions The rational function is in the form $f(x) = \frac{1}{x}$. It looks like this:

```
def f(x):
    with np.errstate(divide='ignore', invalid='ignore'):
        return 1/x
fx_name = r'$f(x)=\frac{1}{x}$'

x=np.linspace( -10,10,101)
y=f(x)
plt.plot(x, y, label=fx_name)
plt.legend(loc='upper left')
plt.show()
```

Trigonometry

Angles An **angle** is the space between two lines that meet. We often represent angles using the symbol θ, pronounced "theta".

There are 2 main measurement systems we can use to define angles. The first measurement system is called degrees, denoted with the degree symbol ř, where you can think of 360° as one full circle. Thus, one angle of 1 degree is like slicing a circle into 360 pieces, and just taking one of those pieces.

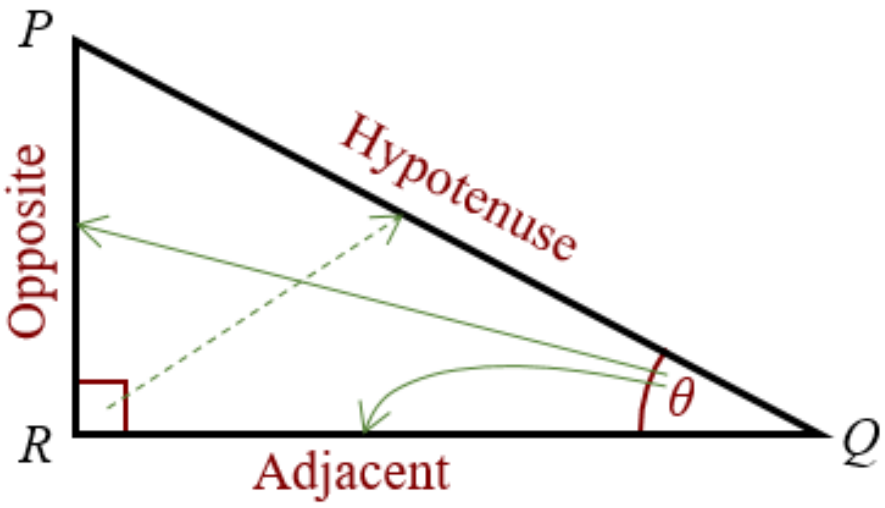
The second measurement system is called radians, where one full circle is 2π. This is a bit more abstract to understand, but you can use the same way to imagine: imagine taking a circle and cutting it into 2π (that's around 6.283) pieces. What? Who would *ever* cut a circle into 6.283 pieces? Yes, it doesn't make a lot of sense, but if you just think of one of those pieces, you would have an angle of 1 radian. We can convert between degrees and radians with:

$$1 \text{ radian} = 1^\circ \cdot \left(\frac{\pi}{180}\right) \tag{51}$$

$$1^\circ = 1 \text{ radian} \cdot \left(\frac{\pi}{180}\right) \tag{52}$$

Right-angled triangles Trigonometry is the study of **right-angled triangles** - triangles that have one right angle and one other angle θ.

The longest side of a right-angled triangle is called the **hypotenuse**. The **opposite** side is the the side that, well, is *opposite* the angle θ. Finally, the **adjacent** side is the side between the right angle and the angle θ.



But wait, doesn't a triangle have three angles? Yes, that is true, and so the choice of which angle you call θ doesn't matter. You can define either one of the two other angles (the non-right-angle angles) as your θ angle, you just have to stick with one angle in your calculations.

Every right-angled triangle obeys the rule that:

$$\text{Adjacent}^2 + \text{Opposite}^2 = \text{Hypotenuse}^2 \tag{53}$$

Trigonometric functions Trigonometric functions are oscillating functions that form a constant repeating pattern. The main, sine (sin), cosine (cos), and tangent (tan), are defined by:

$$\sin\theta = \frac{\text{opposite}}{\text{adjacent}} \tag{54}$$

$$\cos\theta = \frac{\text{adjacent}}{\text{hypotenuse}} \tag{55}$$

$$\tan\theta = \frac{\text{opposite}}{\text{hypotenuse}} = \frac{\sin\theta}{\cos\theta} \tag{56}$$

The *reciprocal trigonometric functions* are the normal trig functions but "flipped over", and they look like this:

$$\csc\theta = \frac{1}{\sin} \tag{57}$$

$$\sec\theta = \frac{1}{\cos} \tag{58}$$

$$\cot\theta = \frac{1}{\tan} \tag{59}$$

You can imagine taking a right-angled triangle and slowly making its angle θ bigger and bigger and bigger. The ratio between the sides is the value of the trigonometric functions.

The graph of $f(\theta) = \sin(\theta)$, where θ is in units of radians, looks like this:

```
from matplotlib.ticker import FuncFormatter, MultipleLocator

f, ax = plt.subplots()
x = np.linspace(0, 2 * np.pi)
ax.plot(x, np.sin(x))
ax.set_ylim( -1, 1)
ax.grid(True)
ax.set_title(r"$f(x) = \sin(x)$")
```

```
ax.xaxis.set_major_formatter(FuncFormatter(
   lambda val, pos: f"{val/np.pi:.0g}" + r"$\pi$" if val !=0 else '0'
))
ax.xaxis.set_major_locator(MultipleLocator(base=np.pi))

plt.show()
```

The values of $f(x) = \sin(x)$ can be deduced by reading off the graph:

$$\sin(0) = 0 \sin(\pi/2) = \sin(3\pi/2) = 1 \sin(\pi) = \sin(2\pi) = 0 \tag{60}$$

Where $\pi \approx 3.14159$.

The graph of $f(x) = \cos x$ looks like the a shifted sine graph:

```
f, ax = plt.subplots()
x = np.linspace(0, 2 * np.pi)
ax.plot(x, np.cos(x))
ax.set_ylim( -1, 1)
ax.grid(True)
ax.set_title(r"$f(x) = \cos(x)$")
ax.xaxis.set_major_formatter(FuncFormatter(
   lambda val, pos: f'{val/np.pi:.0g}' + r"$\pi$" if val !=0 else '0'
))
ax.xaxis.set_major_locator(MultipleLocator(base=np.pi))

plt.show()
```

Whereas the graph of $f(x) = \tan x$ looks a little different:

```
def f_tan(x):
    with np.errstate(divide='ignore', invalid='ignore'):
        return np.tan(x)

f, ax = plt.subplots()
x = np.linspace(0, 2 * np.pi, 1000)
y = np.sin(x) / np.cos(x)
tol = 100
y[y > tol] = np.nan
y[y < -tol] = np.nan
ax.plot(x, y)
ax.grid(True)
ax.set_ylim( -10, 10)
ax.set_title(r"$f(x) = \tan(x)$")
ax.xaxis.set_major_formatter(FuncFormatter(
   lambda val, pos: f'{val/np.pi:.0g}' + r"$\pi$" if val !=0 else '0'
))
ax.xaxis.set_major_locator(MultipleLocator(base=np.pi))

plt.show()
```

Radicals If we know a number, say, 4, can we find two equal numbers that multiply to form that number? Actually, we can! In the case of 4, we know that:

$$4 = 2 \times 2 \tag{61}$$

So we can say that the "square root" of 4 is 2, and we use the $\sqrt{}$ symbol to show this:

$$\sqrt{4} = 2 \tag{62}$$

We can expand this idea. If we knew a number, say 8, can we find three equal numbers that multiply to form that number? Yes, we can! In the case of 8, we know that:

$$8 = 2 \times 2 \times 2 \tag{63}$$

So we can say that the "cube root" of 8 is 2, and we use the $\sqrt[3]{}$ symbol to show this:

$$\sqrt[3]{8} = 2 \tag{64}$$

The same goes for 4 equal numbers that multiply to form a number, 5, 6, and so on. These roots, from the square root and cube root to that 4th and 5th and 6th root, are called **radicals**. Radicals follow these rules:

$$\sqrt{1} = 1 \tag{65}$$
$$\sqrt{0} = 0 \tag{66}$$
$$\sqrt{a} \times \sqrt{a} = a \tag{67}$$
$$\sqrt{ab} = \sqrt{a} \times \sqrt{b} \tag{68}$$

Exponentials and Logarithms

The rules of exponents Exponents are every time we multiply one number by itself a certain number of times. For example, 2^2, pronounced "2 squared" or "2 to the power of 2" is two multiplied by itself, two times - that is 2×2. 2^3, pronounced "2 cubed" or "2 to the power of 3" is two multiplied by itself, three times - that is $2 \times 2 \times 2$. Exponents follow these rules:

$$1^a = 1 \tag{69}$$
$$0^a = 0 \tag{70}$$
$$a^0 = 1 \tag{71}$$
$$a^1 = a \tag{72}$$
$$a^m + a^n = a^{(m+n)} \tag{73}$$
$$a^m / a^n = a^{(m-n)} \tag{74}$$
$$(a^b)^c = a^{(b \times c)} \tag{75}$$

Fractional exponents are the same as a radical:

$$a^{\frac{1}{2}} = \sqrt{a} \tag{76}$$
$$a^{\frac{1}{3}} = \sqrt[3]{a} \tag{77}$$

Finally, negative exponents are equal to one over the positive exponent of that number:

$$a^{-n} = \frac{1}{a^n} \tag{78}$$

Exponential and logarithmic functions All exponential functions are in the form $f(x) = b^x$ where $b > 0$ and $b \neq 1$. Note that if x is negative, then $b^{-x} = \left(\frac{1}{b}\right)^x$. The most commonly-used exponential function is $f(x) = e^x$, also confusingly called "the exponential function", where e is a special number that is around 2.718:

```
x = np.linspace( -5, 5)
plt.grid(True)
plt.plot(x, np.exp(x))
plt.title(r"$f(x) = e^x$")
plt.show()
```

What if we want to "undo" the exponential function? We would need to find an **inverse function**, a function that does the opposite thing as another function. In the case of an exponential function, the inverse is called a **logarithmic function**. The basic logarithmic function is given by:

$$y = \log_b x \tag{79}$$

where $b^y = x$. To remember this mapping between exponential functions, you can remember that b is the "basement" (because of its subscript), and it's raised to the "answer" of y to get x. Several common logarithmic functions have shorthand notations:

$$\ln x = \log_e x \tag{80}$$
$$\log x = \log_{10} x \tag{81}$$

The ln function is also called the "natural logarithm", and it looks like this:

```
x = np.linspace(0.01, 5)
plt.grid(True)
plt.plot(x, np.log(x))
plt.title(r"$f(x) = \ln x$")
plt.show()
```

Just as with exponents, logarithms follow certain rules:

$$\log_b(xy) = \log_b x + \log_b y \tag{82}$$
$$\log_b(x^n) = n \log_b x \tag{83}$$
$$\log_b\left(\frac{x}{y}\right) = \log_b x - \log_b y \tag{84}$$
$$\log_b x = \frac{\ln x}{\ln b} \tag{85}$$

> **Important**
> Be careful! Note that $(\log_b x)^r \neq r \log_b x$!!!

Factorials A factorial is when you multiply a number by all whole numbers smaller than it, and is denoted with !. For example, $2! = 2 \times 1$, $3! = 3 \times 2 \times 1$, and $4! = 4 \times 3 \times 2 \times 1$.

Sums Sometimes, we want to add up a lot of numbers:

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + \dots \tag{86}$$

How do we compactly write this out? One way is to assign each number a symbol, with its position in the list of numbers to add as an index on the bottom. So 1, which is the first number, would be a_1. This becomes:

$$a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + \dots \tag{87}$$

A compact way of writing this out is with the **summation symbol**:

$$\sum_{i=1}^{n} a_i = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 \tag{88}$$

where i is the index on the bottom, $i = 1$ means the indices start from 1, and n means the indices end at n. n can be infinity sometimes:

$$\sum_{i=1}^{\infty} a_i = a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + \dots \tag{89}$$

Summation is very useful for shortening long formulas. For example, the **binomial expansion formula**:

$$(a + b)^n = \sum_{i=0}^{n} \frac{n!}{(n-i)!i!} a^{n-i} b^i \tag{90}$$

Introduction to calculus

> Calculus is the most powerful weapon of thought yet devised by the wit of man.
> **Wallace B. Smith**

Calculus is the study of continuous change: a simple set of concepts that can be applied to the most complex of systems. Nearly all of modern-day science is built on calculus, especially physics; it is quite the tool to have in your engineer's toolbox!

Over its nearly 400-year history, calculus has evolved into perhaps the most versatile and broad field of mathematics. There is multivariable calculus, vector calculus, stochastic calculus, the calculus of variations, tensor calculus, fractional calculus...we could go on and on...

However, a lifelong journey through calculus begins with monovariable calculus (or just "calculus"). In monovariable calculus, we're concerned with developing the fundamental ideas of calculus. These ideas will carry over to all the more advanced forms of calculus. Through them, you're realize how calculus has such versatility and power.

Limits The **limit** is the value a function approaches as x approaches a given value.

For instance:

$$f(x) = \frac{x^2 - 1}{x - 1} \tag{91}$$

While the function is not defined at $x = 1$, it approaches 2 as x approaches 1, thus we can say:

$$\lim_{x \to 1} f(x) = 2 \tag{92}$$

Sometimes, limits are the same regardless of the direction x approaches c. At other times, the limit is different, and dependent on the direction. For instance, take:

$$f(x) = \frac{1}{x - 3} \tag{93}$$

As x approaches 3 from the right-hand side, $f(x)$ approaches infinity, and as x approaches 3 from the left-hand side, $f(x)$ approaches **negative** infinity. Thus, we can say there are separate left-hand and right-hand limits, where:

$$\lim_{x \to 3^+} f(x) = \infty = DNE \tag{94}$$

$$\lim_{x \to 3^-} f(x) = -\infty = DNE \tag{95}$$

(DNE means "does not exist")

Notice that as the limit is infinite, and infinity is not a real number, we do not say that the limit of the function is infinity. Rather, we say that the limit does not exist.

The limit of a function does not exist in one of three cases:

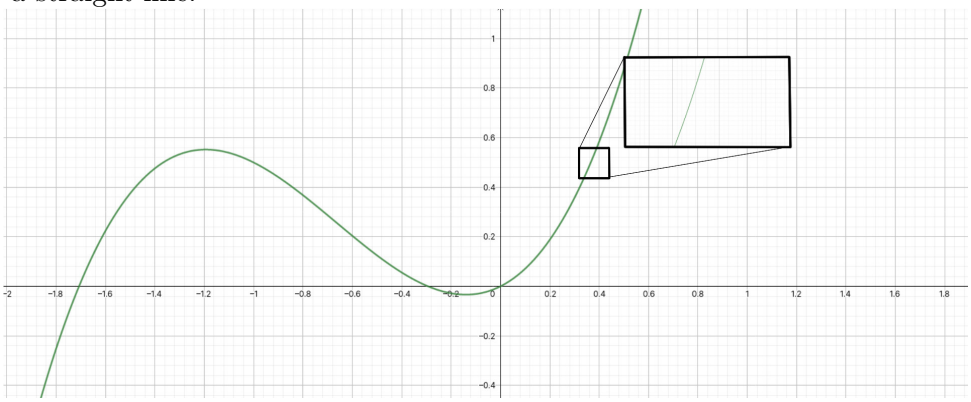
- If the function's left- and right-hand limits are not equal at that x-value
- If the function oscillates in value around an x-value
- If the function becomes indefinitely large around an x-value

Derivatives The equation for the slope of a line is defined as:

$$m = \frac{\Delta f(x)}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1} \tag{96}$$

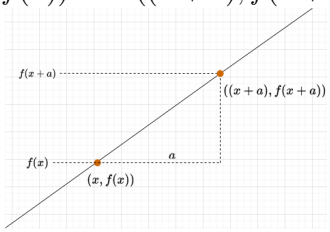
However, the slope equation only works for straight lines. How, then, could we find the slope of a curve?

Well, we can first take advantage of the fact that if you zoom *really* close in to a curve, it looks like a straight line:



Notice how, as we zoom into the curve, the curve looks more and more like a straight line, and the curvature becomes less and less noticeable.

The **derivative** is a function that tells you the slope of another function at *any* point. You can think of it as an "upgraded" version of the slope formula. We find the derivative by taking two points, $(x, f(x))$ and $((x+a), f(x+a))$, and calculating the slope from them:



As we shrink a and make it smaller and smaller, a will approach zero, and the slope becomes:

$$m = \frac{y_2 - y_1}{x_2 - x_1} = \frac{f(x+a) - f(x)}{(x+a) - x} = \frac{f(x+a) - f(x)}{a} \tag{97}$$

So, for a function f, the derivative $\frac{df}{dx}$ is defined as the limit by taking a smaller and smaller:

$$\frac{df}{dx} = \lim_{a \to 0} \frac{f(x+a) - f(x)}{a} \tag{98}$$

An alternative understanding of the derivative A more intuitive, but less mathematically rigorous, definition of the derivative is to look at the slope formula once again:

$$m = \frac{\Delta y}{\Delta x} \tag{99}$$

Now, let's imagine making the change Δ smaller and smaller. The eventual result is that Δy, a small change, becomes a tiny change dy, and Δx, a small change, becomes a tiny change dx:

$$\lim_{\Delta \to 0} \frac{\Delta y}{\Delta x} = \frac{dy}{dx} \tag{100}$$

(Opinionated) derivative notation Calculus was invented at roughly the same time by two brilliant mathematicians - Gottlieb Leibniz and Isaac Newton. Unfortunately, each of them published their work at the same time with differing notations. Leibniz wrote the derivative with the notation $\frac{df}{dx}$; Newton used the notation $\dot{f}$. Lagrange and Euler, not long after, came up with the notation $f'(x)$. In any case, Leibniz and Newton...got into a fight, which became a political controversy, and other mathematicians decided to develop *other* notations as well. So, sadly, there is not a unified notation around calculus.

The most common form of notation is Leibniz's notation, where the derivative of $f(x)$ is written like this:

$$\frac{df}{dx} \tag{101}$$

The *nth*-derivative is written as:

$$\frac{d^n f}{dx^n} \tag{102}$$

Important
Note that even though this looks like a fraction, and can be manipulated similar to fractions, the derivative in Leibniz's notation is **absolutely not** the same as a fraction!

The second most common notation is Langrange's notation, where the derivative of $f(x)$ is written like this:

$$f'(x) \tag{103}$$

Here, the *nth*-derivative is written as:

$$f^n(x) \tag{104}$$

In Project Elara, derivatives of a single-variable function use Leibniz notation most commonly to minimize confusion. The derivative of a function $f(x)$ in Project Elara is *preferred* to be written as:

$$\frac{df}{dx} \tag{105}$$

If we have a function $v(t)$, the derivative with respect to t would be written as:

$$\frac{dv}{dt} \tag{106}$$

A higher order derivative (e.g. second derivative) is written like this:

$$\frac{d^2 f}{dx^2} \tag{107}$$

The derivative evaluated at a certain point $x = a$ is written as:

$$\left.\frac{df}{dx}\right|_{x=a} \tag{108}$$

However, there are some alternate notations that will sometimes be used. Note that these are **all** equivalent:

$$\frac{d}{dx} f(x) = f'(x) \tag{109}$$

You will also see $\dot{f}$ and $\ddot{f}$ sometimes for first and second time derivatives respectively. This is recommended not to be used as it is easily notationally confused.

Differentiation The derivative is a very powerful function, but finding the derivative of a function unfortunately requires a bit of time and patience. This is because there is no universal formula for finding the derivative of a certain function - instead, we have general *rules* for finding the derivatives of a certain type of function, which we use in the process of **differentiation**.

Let's start with the easiest derivative - the derivative of any constant function is zero. Why? Because the slope of any constant function is always zero, and remember, the derivative is a function that tells you the slope at every point. So if the slope at every point is zero, the derivative will always be zero.

We call this the **constant rule**, and we write it out like this:

$$\frac{dC}{dx} = 0 \tag{110}$$

Where n can be any constant. For instance, the derivative of 2 with respect to x (the same as finding the rate of change of $f(x) = 2$) would be:

$$\frac{d(2)}{dx} = 0 \tag{111}$$

This also means that if you have a function $f(x) = c$, where c is a constant, then:

$$\frac{df}{dx} = c \tag{112}$$

That should be simple enough, right?

Now, let's do the second-easiest derivative. The derivative of the exponential function $f(x) = e^x$ is itself. We call this the **exponential rule**, and we write it out like this:

$$\frac{d(e^x)}{dx} = e^x \tag{113}$$

The exponential rule can also be more generally written as this:

$$\frac{d(a^x)}{dx} = \ln(a)a^x \tag{114}$$

For trigonometric functions, the derivatives unfortunately have to be memorized, but you just have to memorize two of them to find the derivatives of all trigonometric functions:

$$\frac{d(\sin x)}{dx} = \cos x \tag{115}$$

$$\frac{d(\cos x)}{dx} = -\sin x \tag{116}$$

And for polynomial functions, we can use the **power rule**:

$$\frac{d(x^n)}{dx} = nx^{n-1} \tag{117}$$

The power rule applies to linear functions in the form $y = mx + c$:

$$\frac{df}{dx} = \frac{d(mx+c)}{dx} = \frac{d(mx)}{dx} = 1m\left(x^{1-0}\right) = m \tag{118}$$

As well as for rational functions in the form $f(x) = \frac{1}{x^n}$:

$$\frac{d\left(\frac{1}{x^n}\right)}{dx} = \frac{d(x^{-n})}{dx} = -nx^{-n-1} = -\frac{n}{x^{n+1}} \tag{119}$$

And nth root functions (e.g. square root, cube root, etc.) in the form $f(x) = \sqrt[n]{x}$:

$$\frac{d\left(\sqrt[n]{x}\right)}{dx} = \frac{d\left(x^{\frac{1}{n}}\right)}{dx} = \frac{1}{n}x^{\frac{1-n}{n}} \tag{120}$$

Combining the power rule and exponential rule gives us the derivatives of logarithms:

$$\frac{d(\ln x)}{dx} = \frac{1}{x} \tag{121}$$

$$\frac{d(\log_a x)}{dx} = \frac{1}{x\ln a} \tag{122}$$

However, most functions are made from a *combination* of these functions. For instance, the function $f(x) = 2x^2 + 3x + 5$ is a combination of a constant function, linear function, and power function. To find the derivatives of combinations of functions, we have a few more rules to help us.

First, we have the **sum rule**:

$$\frac{d(f(x) + g(x))}{x} = \frac{df}{dx} + \frac{dg}{dx} \tag{123}$$

Then, the **constant coefficient rule:**

$$\frac{d(c \cdot f(x))}{dx} = c \cdot \frac{df}{dx} \tag{124}$$

Then, the **product rule**:

$$\frac{d(f(x)g(x)}{dx} = \frac{df}{dx}g(x) + \frac{dg}{dx}f(x) \tag{125}$$

From the product rule, we can derive the **quotient rule**:

$$\frac{d\left(\frac{f(x)}{g(x)}\right)}{dx} = \frac{\frac{df}{dx}g(x) - \frac{dg}{dx}f(x)}{(g(x))^2} \tag{126}$$

And, most importantly, we have the **chain rule**. The chain rule is used for *composite functions* - functions that have been nested into each other. For instance, $h(x) = \sin x^2$ is made by nesting the function $g(x) = x^2$ *inside* of the function $f(x) = \sin x$. So, we can say that $h(x) = f(g(x))$. This is a **composition of functions**.

With that in mind, the **chain rule** is written like this:

$$\frac{df(g(x))}{dx} = \frac{df}{du}\frac{du}{dx} = \frac{df}{dx}\{g(x)\}\frac{dg}{dx} \tag{127}$$

This means we nest $g(x)$ in the derivative of $f(x)$ and multiply that by the derivative of $g(x)$. The other rules here are mostly self-explanatory, but I'll go through a worked example with the chain rule: let's try to find the derivative of $h(x) = \cos x^2$.

Practicing the chain rule We use the chain rule for *composite* functions, like our example, $h(x) = \cos x^2$. We know that we can rewrite $h(x)$ as a composite function $f(g(x))$, where:

$$h = \cos(u) \tag{128}$$

$$u = x^2 \tag{129}$$

We can now use the chain rule. In the first step, we find $\frac{dh}{du}$:

$$\frac{dh}{du} = -\sin(u) = -\sin(x^2) \tag{130}$$

Note that we substituted in x^2 for u. Now, we find $\frac{du}{dx}$:

$$\frac{du}{dx} = 2x \tag{131}$$

We now just need to multiply them together:

$$\frac{dh}{dx} = \frac{dh}{du}\frac{du}{dx} \tag{132}$$

$$\frac{dh}{dx} = -2x \sin x^2 \tag{133}$$

That's our answer!

Reciprocal derivatives In monovariable calculus *only*, derivatives follow the **reciprocal rule**:

$$\frac{df}{dx} = \frac{1}{\frac{dx}{df}} \tag{134}$$

$$\frac{dx}{df} = \frac{1}{\frac{df}{dx}} \tag{135}$$

Tangent to a curve The tangent to the function $f(x)$ at $x = a$ is given by the function $T(x)$, where:

$$\sigma(x) = \frac{df}{dx} \tag{136}$$

$$T(x) = \sigma\left(a\right)\left(x - a\right) + f\left(a\right) \tag{137}$$

This is also called the **linear approximation** of a function.

Higher-order derivatives Taking a derivative *nth* times gives you the *nth* derivative of a function. For example, taking the derivative of the derivative is the second derivative, the derivative of the second derivative is the third derivative, and so on. Going in the other direction, the 0th derivative is not taking the derivative at all - the same as the original function.

The second derivative is the most common higher-order derivative, and it is given by:

$$\frac{d^2 f}{dx^2} = \frac{d\left(\frac{df}{dx}\right)}{dx} \tag{138}$$

The *order* of the derivative is the number of times you take the derivative: for instance, the 7th derivative involves taking the derivative of a function 7 times! (don't do that, please...)

Finding maxima and minima We can find the critical points (maxima and minima) of a function $f(x)$ by finding its derivative and setting it to zero:

$$\frac{df}{dx} = 0 \tag{139}$$

For instance, for the function $f(x) = 2x^2 + 1$:

$$\frac{df}{dx} = 4x \tag{140}$$

$$4x = 0 \tag{141}$$

$$x = 0 \tag{142}$$

Then, plugging in that x value into the original f(x), we can find the y value of the maximum/minimum point:

$$f(0) = 2(0^2) + 1 = 1 \tag{143}$$

So the minimum of $f(x)$ is the point $(0, 1)$. How do we know if it's a maximum or minimum though? To find out, we use the **second derivative**, which measures how the slope changes. If a point is a maximum, the slope will change from positive to negative around that point; if a point is a minimum, vice-versa. So:

$$\begin{cases} \text{if} & \frac{d^2 f}{dx^2} < 0 \quad x = \text{max.} \\ \text{if} & \frac{d^2 f}{dx^2} > 0 \quad x = \text{min.} \end{cases} \tag{144}$$

The second derivative of $f(x)$ is the derivative of its derivative, which we can find like this:

$$\frac{d^2f}{dx^2} = \frac{d(4x)}{dx} = 4 \tag{145}$$

Since $\frac{d^2f}{dx^2} > 0$, we know that the point $(0, 1)$ has to be a **minimum**.

Implicit differentiation Implicit differentiation is when we differentiate with respect to an intermediary variable, often used when the object being differentiated is not a function. For instance, consider the equation of a circle:

$$x^2 + y^2 = 1 \tag{146}$$

To implicitly differentiate it, we have:

$$\frac{d}{dx}(x^2 + y^2) = \frac{d}{dx}(1) \tag{147}$$

The right-hand side is straightforward:

$$\frac{d}{dx}(x^2 + y^2) = 0 \tag{148}$$

The sum rule can be used for the left-hand side:

$$\frac{d}{dx}x^2 + \frac{d}{dx}y^2 = 0 \tag{149}$$

$$2x + \frac{d}{dx}y^2 = 0 \tag{150}$$

To implicitly differentiate y^2, we can use the chain rule:

$$\frac{d}{dx}y^2 = 2y\frac{dy}{dx} \tag{151}$$

So:

$$2x + 2y\frac{dy}{dx} = 0 \tag{152}$$

Rearranging the terms, we find that:

$$2y\frac{dy}{dx} = -2x \tag{153}$$

$$\frac{dy}{dx} = -\frac{x}{y} \tag{154}$$

Problem: Suppose that $y = x^2 + 3$. Find $\frac{dy}{dt}$ when $x = 1$, $\frac{dx}{dt} = 2$.
We implicitly differentiate y to find:

$$\frac{dy}{dt} = 2x\frac{dx}{dt} \tag{155}$$

Then we plug in the values for x and $\frac{dx}{dt}$ to find that:

$$\frac{dy}{dt} = 2(1)(2) = 4 \tag{156}$$

Antiderivatives The antiderivative is the function $F(x)$ that is the original function of a derivative $f(x)$. For instance, the antiderivative of $2x$ is x^2, because if $2x$ is the derivative, then the original function is x^2:

$$\frac{d(x^2)}{dx} = 2x \tag{157}$$

The **indefinite integral** is another name for the antiderivative. The indefinite integral is written as:

$$F(x) + C = \int f(x)dx \tag{158}$$

The C is due to the fact that the integral returns a family of antiderivatives, as any derivative has infinitely many antiderivatives.

Just like derivatives, indefinite integrals follow certain rules:

$$\int dx = x + C \tag{159}$$

Constant rule:

$$\int k\,dx = kx + C \tag{160}$$

Inverse power rule:

$$\int x^n dx = \frac{x^{n+1}}{n+1} \tag{161}$$

Sum and difference rule:

$$\int [f(x) \pm g(x)]dx = \int f(x)dx \pm \int g(x)dx \tag{162}$$

Constant factor rule:

$$\int kf(x)dx = k\int f(x)dx \tag{163}$$

An intuitive understanding Suppose we have the derivative of a function, and that derivative is $f(x)$. We now want to find the original function $F(x)$ that derivative came from. To do so, we notice that:

$$f(x) = \frac{dF}{dx} \tag{164}$$

So if we rearrange the terms, we get:

$$f(x)dx = dF \tag{165}$$

Now, dF is similar to saying "a tiny tiny change in F". If we add lots of dF's together, it would seem reasonable that we would get F:

$$F(x) = \sum_i dF \tag{166}$$

Only in integral calculus, we replace the summation symbol with the integral symbol $\int$, so it becomes:

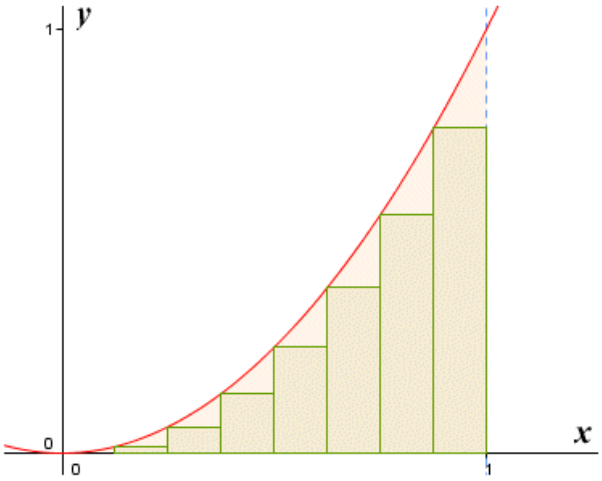
$$F(x) = \int dF = \int f(x)dx \tag{167}$$

Integration

Area under a curve Up to this point, we've only seen one type of integral - the indefinite integral, used for finding the original function given its derivative. Now, we will explore another type of integral, the definite integral, which returns an area rather than a function. The definite integral looks like this:

$$A = \int_a^b f(x)dx \tag{168}$$

Intuitively, the definite integral gives the area underneath a curve. It does this by creating tiny rectangles under that curve:



The area under a curve can be approximated by summing the areas of tiny tiny rectangles under the curve.

We can find the area by summing the areas of the rectangles we place under the curve. As more rectangles are used, the approximation becomes closer to the true area. Taking the limit as the number of rectangles approaches infinity, we find the true area underneath the curve:

$$A = \lim_{n\to\infty} \sum_{i=1}^{n} f(c_i)\Delta x = \int_a^b f(x)dx \tag{169}$$

Several properties of the definite integral include:

$$\int_a^a f(x)dx = 0 \tag{170}$$

$$\int_b^a f(x)dx = -\int_a^b f(x)dx \tag{171}$$

$$\int_a^c f(x)dx = \int_a^b f(x)dx + \int_b^c f(x)dx \tag{172}$$

Average value of a function The average (mean) value of a function on $[a, b]$ is given by:

$$\frac{1}{b-a}\int_a^b f(x)dx \tag{173}$$

The fundamental theorem of calculus Derivatives, indefinite integrals, and definite integrals are related by the **fundamental theorem of calculus (FTC)**. The theorem consists of two parts.

The first part of the FTC states that if you have an original function $F(x)$ whose derivative is $f(x)$, then taking the indefinite integral of the derivative gives you the original function:

$$f(x) = \frac{dF}{dx} \Rightarrow F(x) = \int f(x)dx \tag{174}$$

The FTC #1 establishes that differentiation and integration are **inverse operations**. The derivative is the *rate of change*, whereas the integral finds the *accumulated net change*.

Then the second part of the FTC states that the definite integral, which is the area underneath the curve of $f(x)$, is given by evaluating the indefinite integral between the bounds of the curve:

$$A = \int_a^b f(x)dx = F(b) - F(a) \tag{175}$$

This means that when we know the indefinite integral of a function, we can use the FTC as a shortcut to evaluating the definite integral.

For instance, let's try to evaluate the area under the curve of $f(x) = x^2$ from $x = 0$ to $x = 3$:

$$\int_0^3 x^2 dx \tag{176}$$

We first find the indefinite integral of $f(x)$ - the original function whose derivative is x^2. We can use the inverse power rule for this:

$$\int x^n dx = \frac{1}{n+1}x^{n+1} + C \tag{177}$$

Thus the indefinite integral of $f(x)$ is equal to:

$$F(x) = \frac{x^3}{3} + C \tag{178}$$

Now, we just evaluate $F(3) - F(0)$:

$$F(3) - F(0) = \left(\frac{3^3}{3} + C\right) - \left(\frac{0^3}{3} + C\right) \tag{179}$$

The two C's cancel and we are simply left with:

$$F(3) - F(0) = 9 \tag{180}$$

Thus:

$$\int_0^3 x^2 dx = 9 \tag{181}$$

U-substitution Just as we have the chain rule for derivatives, we have a sort of "reverse" chain rule for evaluating indefinite integrals. We begin with a typical integral:

$$\int f(x)dx \tag{182}$$

And we want to express the integrand $f(x)$ in terms of another function $g(u)$. To do this, we define:

$$x = g(u) \tag{183}$$

$$f(x) = f(g(u)) \tag{184}$$

$$\frac{dx}{du} = g'(u) \tag{185}$$

$$dx = g'(u)du \tag{186}$$

Therefore:

$$\int f(x)dx = \int f(g(u))g'(u)du \tag{187}$$

which is essentially a reverse chain rule. Why would we do this though? Doesn't it make integrals more complicated? While the formula may seem scary, the reality is that with clever cancelling, u-substitution can make integrals a lot easier to solve. Take, for instance:

$$\int \frac{x^3}{\sqrt{16-x^4}}dx \tag{188}$$

Here, we identify the part of the integral that we want to make our u. Usually, this is an annoyingly complex part of the integral that we want to make simpler with u-substitution. In this case, it would be $u = 16 - x^4$. So:

$$u = 16 - x^4 \tag{189}$$

$$\frac{du}{dx} = -4x^3 \tag{190}$$

$$dx = \frac{du}{-4x^3} \tag{191}$$

Now, we can substitute $16 - x^4$ for u:

$$\int \frac{x^3}{\sqrt{16-x^4}}dx = \int \frac{x^3}{\sqrt{u}}dx \tag{192}$$

And substitute $\frac{du}{-4x^3}$ for dx:

$$\int \frac{x^3}{\sqrt{u}}dx = \int \frac{x^3}{\sqrt{u}}\frac{du}{-4x^3} \tag{193}$$

And magically we find that the x^3 and $-4x^3$ cancel, leaving us with an integrand completely in terms of u:

$$\int \frac{x^3}{\sqrt{u}}\frac{du}{-4x^3} = \int -\frac{1}{4}\frac{1}{\sqrt{u}}du \tag{194}$$

We can make the integral even simpler by moving the constant factor outside the integral:

$$\int -\frac{1}{4}\frac{1}{\sqrt{u}}du = -\frac{1}{4}\int \frac{1}{\sqrt{u}}du \tag{195}$$

We've successfully greatly simplified the integral!

$$\int \frac{x^3}{\sqrt{16-x^4}}dx \Rightarrow -\frac{1}{4}\int \frac{1}{\sqrt{u}}du \tag{196}$$

Solving this integral in terms of u is easy:

$$-\frac{1}{4}\int \frac{1}{\sqrt{u}}du = -\frac{1}{2}\sqrt{u} + C \tag{197}$$

We just need to remember to now replace u with its value in terms of x:

$$-\frac{1}{2}\sqrt{u}+C=-\frac{1}{2}\sqrt{16-x^4}+C \tag{198}$$

So we evaluated a difficult integral using u-substitution to find its answer!

Integration by parts Integration by parts is given by:

$$\int udv=uv-\int vdu \tag{199}$$

It is the integral form of the product rule for derivatives. Unsurprisingly, it is often used for integrating products of functions. To illustrate this, let's consider integrating $f(x)=x\sin(x)$:

$$\int x\sin(x)dx \tag{200}$$

To integrate, we first pick our u and dv. We generally want u to be a function that is easy to differentiate, and dv to be a function that is easy to integrate. Here, we will pick $u=x$, $dv=\sin(x)$. So:

$$u=x \tag{201}$$

$$\frac{du}{dx}=1 \tag{202}$$

$$du=1dx=dx \tag{203}$$

$$dv=\sin x \tag{204}$$

$$v=\int dv=-\cos x \tag{205}$$

Using the integration by parts formula, we have:

$$\int x\sin(x)dx=-x\cos x-\int(-\cos x)dx \tag{206}$$

Which simplifies to:

$$\int x\sin xdx=-x\cos x+\sin x+C \tag{207}$$

Other integration rules There are several other notable integration tricks such as integrating by partial fractions, integrating by trigonometric substitutions, using trig identities, long division, and other algebraic methods for solving indefinite integrals. A full list of them can be found at https://brilliant.org/wiki/integration-tricks/. Another way to solve complicated integrals is to refer to integral tables, which contain precomputed tables of common integrals - see Lists of integrals.

Integral calculators When integrals are too difficult to solve using any known method, consult an online integral calculator! Some very good ones are https://mathdf.com/int/ and https://www.wolframalpha.com. Additionally, for relatively simple expressions (to type in), consult https://gamma.sympy.org/. The SymPy Python library is also well worth learning to do symbolic integration on a computer.

Taylor series Imagine we had a certain function $f(x)$. We want to approximate that function with a polynomial $T(x)$. So we write the general equation of a polynomial centered at $x = a$:

$$T(x) = c_0 + c_1(x-a) + c_2(x-a)^2 + c_3(x-a)^3 + c_4(x-a)^4 + \dots c_n(x-a)^n \tag{208}$$

Now, for this function to be equal to $f(x)$, then $T(a) = f(a)$. So we compute $T(a)$:

$$T(a) = c_0 + c_1(a-a) + c_2(a-a)^2 + c_3(a-a)^3 + c_4(a-a)^4 + \dots c_n(a-a)^n \tag{209}$$

Since $a - a$ cancels to zero, every term in the equation cancels to zero other than c_0, so:

$$T(a) = c_0 \tag{210}$$

But we know that $T(a) = f(a)$. So we can rewrite $T(x)$ as:

$$T(x) = f(a) + c_1(x-a) + c_2(x-a)^2 + c_3(x-a)^3 + c_4(x-a)^4 + \dots c_n(x-a)^n \tag{211}$$

Alright, but just having two functions be equal at a single point isn't enough to make sure that these two functions are equal. To be more sure that the two functions are equal, we demand $T'(a) = f'(a)$ must be true as well. If we compute $T'(x)$, we'd get:

$$T'(x) = c_1 + 2c_2(x-a) + 3c_3(x-a)^2 + 4c_4(x-a)^3 + \dots nc_n(x-a)^{n-1} \tag{212}$$

Once again, if we compute $T'(a)$, the terms all cancel other than the first, so we have:

$$T'(a) = c_1 + 2c_2(a-a) + 3c_3(a-a)^2 + 4c_4(a-a)^3 + \dots nc_n(a-a)^{n-1} \tag{213}$$

Again, since all the $a - a$ terms cancel to zero, we are left with just one term, where:

$$T'(a) = c_1 \tag{214}$$

But again, we know that $T'(a) = f'(a)$, so we can rewrite $T(x)$ as:

$$T(x) = f(a) + f'(a)(x-a) + c_2(x-a)^2 + c_3(x-a)^3 + c_4(x-a)^4 + \dots c_n(x-a)^n \tag{215}$$

Still, we can't be certain that $T(x) = f(x)$. We don't just want their values and their first derivatives to match at $x = a$ - we want their second derivatives to match too! So we do the same exercise - take the second derivative of $T(x)$. We get:

$$T''(x) = 2c_2 + 3(2)c_3(x-a) + 4(3)c_4(x-a)^2 + \dots n(n-1)c_n(x-a)^{n-2} \tag{216}$$

Once again, if we solve for $T''(a)$, every term except for the first cancel to yield zero, so we get:

$$T''(a) = 2c_2 \tag{217}$$

But we said that $f''(a) = T''(a)$, so:

$$c_2 = \frac{1}{2}f''(a) \tag{218}$$

So $T(x)$ is now:

$$T(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2 + c_3(x-a)^3 + c_4(x-a)^4 + \dots c_n(x-a)^n \tag{219}$$

You might be noticing a pattern here! In fact, if we take the general expression of $T(x)$:

$$T(x) = \sum_{n=0}^{\infty} c_n(x-a)^n \tag{220}$$

Then the nth-term of $T(x)$ would be:

$$c_n(x-a)^n \tag{221}$$

If we take the derivative of the nth term, using the power rule, we get:

$$c_n(n)(n-1)(n-2)\dots(3)(2)(1)(x-a)^{n-n} \tag{222}$$

Here we can write $n(n-1)(n-2)(n-3)\dots(3)(2)(1)$ as $n!$ (we call that "n-factorial"). For example, $3! = 3 \times 2 \times 1 = 6$. We can also rewrite $n - n = 0$, and anything raised to the power of zero is just one, so we get this expression for the nth-derivative of T:

$$c_n n! \tag{223}$$

And if we demand that the nth-derivative of $f(x)$ is equal to the nth-derivative of $T(x)$ at $x = a$, then:

$$f^{(n)}(a) = c_n n! \tag{224}$$

If we solve for c_n, we get:

$$c_n = \frac{f^{(n)}(a)}{n!} \tag{225}$$

Then if we plug this back in to the infinite sum representation of $T(x)$, replacing c_n with what we derived:

$$T(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n \tag{226}$$

This is the formula for the **Taylor series** of a function. For practical purposes, we usually don't let the sum range from 0 to infinity, and instead cap the sum at some number, which we call the **order** of the resulting polynomial. For example, the 7th-order Taylor polynomial is a Taylor series capped at 7 terms.

We can use the Taylor series to compute the values of transcendental functions (the exponential function, logarithmic function, and trigonometric functions), given known values of a function. For example, using a 7th-order Taylor approximation at $a = 0$ and range reduction techniques to rescale any value of the sine function to between $-\frac{\pi}{4}$ and $\frac{\pi}{4}$, you can calculate the value of the sine function to within an error of $\pm 3.6 \times 10^{-6}$ of the correct value.

The Taylor series is also a method of calculating the precise value of e, Euler's number, to use in calculations of the exponential function. To do so, we first need to recognize that $e^0 = 1$, as with any other number raised to the power of zero. We also know that because $\frac{d}{dx}e^x = e^x$, the nth-derivative of e^x will always be e^x, and thus the nth-derivative $f^{(n)}(0)$ is always going to be $e^0 = 1$. Plugging that into our Taylor series, and setting $a = 0$ (as our point of reference is $x = 0$), we have:

$$e^x = T(x) = \sum_{n=0}^{\infty} \frac{1}{n!}(x-0)^n \tag{227}$$

Which becomes:

$$T(x) = \sum_{n=0}^{\infty} \frac{x^n}{n!} \tag{228}$$

Now, given also that we know that $e^1 = e$, then $T(1) = e$, so:

$$e = T(1) = \sum_{n=0}^{\infty} \frac{1^n}{n!} \tag{229}$$

And since $1^n = 1$, we have:

$$e = \sum_{n=0}^{\infty} \frac{1}{n!} = 1 + 1 + \frac{1}{2} + \frac{1}{6} + \frac{1}{24} + \dots \tag{230}$$

Which results in:

$$e \approx 2.718 \tag{231}$$

You can also find π using Taylor series in a similar manner (just with a Taylor approximation of the arctangent or arcsine function instead).

Taylor series are also good for approximating complex functions. For example, the Taylor series for $\sin(x)$ starts with:

$$\sin(x) \approx x - \frac{1}{3!}x^3 + \frac{1}{5!}x^5 - \frac{1}{7!}x^7 + \dots \tag{232}$$

Which means the first-order Taylor polynomial for $\sin(x)$ is just:

$$\sin(x) \approx x \tag{233}$$

This is called the **small-angle approximation** to the sine function, and it is very powerful - as we'll see later, it allows many complicated differential equations to be simplified and made solvable.

Vectors and matrices

> "The beauty of mathematics only shows itself to more patient followers."
> **Maryam Mirzakhani**

The fundamental building blocks of physics and engineering are the humble **vector** and **matrix**. Indeed, they are such indispensable mathematical tools that we could scarcely describe our world without them! We will now give an overview of what they are, how to use them, and why they feature so prominently in the sciences.

Vectors To begin, what exactly is a vector? Well, a vector, at least in the physical sense, is a mathematical object used to represent a quantity that has *direction*. For instance, we describe the position of objects with vectors (even if we don't realize it) when we say "that store is a block away to your right" or "I'm on your left-hand side". In a similar way, we can describe the velocity of objects (which is *not* the same thing as speed) with vectors.

To represent a vector mathematically, we write them as a list of items (usually numbers) which represent coordinates in a certain coordinate system. For example, the vector $(4, 2)$ is a line that moves 4 in the x-direction, and 2 in the y-direction. We draw vectors as arrows in space.

A scalar is a number that "scales" (that is, shrinks or stretches) a vector. More generally, a scalar is any singular number.

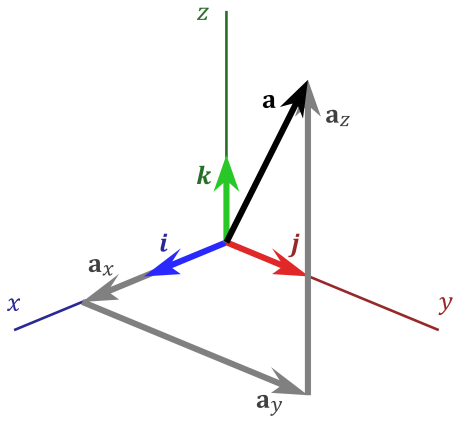
To write a vector (say, $\vec{a}$), we have several notations:

$$\vec{a} = (4, 2) = \begin{bmatrix} 4 \\ 2 \end{bmatrix} = 4\hat{i} + 2\hat{j} \tag{234}$$

The first (point notation) is used for both points and vectors and is the simplest method. The second (matrix notation) is the same thing, just written out vertically in a column. The third (basis vector notation) is slightly different. It defines a vector as a sum of transformed basis vectors, where:

$$\hat{i} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \hat{j} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \hat{k} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{235}$$

The basis vectors are then used to "construct" a vector like this:



A vector as a superposition of Cartesian basis vectors. (source: Wikipedia)

Note
A basis vector can be thought of as a vector with a "tail" at the origin and a "head" at 1 unit from the origin along one of the axes.

We can then express any vector as a linear combination of the basis vectors:

$$\vec{a} = s_1\hat{i} + s_2\hat{j} + s_3\hat{k} \tag{236}$$

Note
A linear combination of basis vectors simply means that we stretch or shrink each basis vector by a certain factor (s being that factor), and then add those stretched/shrinked vectors together to form the final vector.

We use the basis vectors $\hat{i}, \hat{j}, \hat{k}$ in 2D and 3D space, but for nth-dimensional spaces, we use the more general basis vectors $e_1, e_2, e_3, \dots e_n$, so linear combinations work like this:

$$\vec{a} = s_1\vec{e}_1 + s_2\vec{e}_2 + s_3\vec{e}_3 + \cdots + s_n\vec{e}_n \tag{237}$$

Which we often write as:

$$\vec{a} = \sum_{i=1}^{n} s_i\vec{e}_i \tag{238}$$

Important
Remember this notation! This will be very important later on.

To add vectors of any dimension, we add their corresponding components:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_1 + b_1 \\ a_2 + b_2 \\ a_3 + b_3 \end{bmatrix} \tag{239}$$

To multiply a vector by a scalar, we multiply the scalar by each of the components:

$$k\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} k(a_1) \\ k(a_2) \\ k(a_3) \end{bmatrix} \tag{240}$$

To find the length of a vector, we take its magnitude, like this:

$$\text{magn}(\vec{a}) \tag{241}$$

A note on notation
You will also see the magnitude of a vector written as $||\vec{a}||$ but I highly recommend *not* using this alternate notation, because it is so easily confused with the absolute value symbol.

The magnitude of a vector can be found using the Pythagorean theorem, so the magnitude is the square of the sum of each component:

$$\text{magn}\begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \dots \\ a_n \end{bmatrix} = \sqrt{(a_1)^2 + (a_2)^2 + (a_3)^2 + \cdots + (a_n)^2} \tag{242}$$

The **dot product** (or inner product) between vectors tells us how much 2 vectors point in the same direction. The dot product is usually defined like this:

$$\vec{a} \cdot \vec{b} = \text{magn}(\vec{a}) \, \text{magn}(\vec{b}) \cos(\theta) \tag{243}$$

The more two vectors point in the same direction, the larger the dot product, and vice-versa. If the dot product is zero, that means the vectors are perpendicular to each other, and if the dot product is negative, that means the vectors are pointing in opposite directions.

The dot product returns a **scalar**, and there is a simpler formula for computing it:

$$\begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} a_1 \cdot b_1 \\ a_2 \cdot b_2 \\ a_3 \cdot b_3 \end{bmatrix} \tag{244}$$

The **cross product** between vectors returns a new vector perpendicular to both vectors with a magnitude proportional to how far apart the two vectors are pointing. The cross product is usually written like this:

$$\vec{a} \times \vec{b} \tag{245}$$

And the magnitude of the cross product is this:

$$\text{mag}(\vec{a} \times \vec{b}) = \text{magn}(\vec{a}) \, \text{magn}(\vec{b}) \sin(\theta) \tag{246}$$

The cross product can be calculated by writing a special matrix that looks like this:

$$\begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & a_2 & a_3 \\ b_1 & b_2 & b_3 \end{vmatrix} \tag{247}$$

Now, we compute each component of the cross product vector by blocking out the column under that component's basis vector in that matrix. For example, for $\hat{i}$, we block out the first column under it:

$$\begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \square & a_2 & a_3 \\ \square & b_2 & b_3 \end{vmatrix} \tag{248}$$

This leaves us with a square matrix underneath the basis vectors. We just need to compute a special value called the determinant of that matrix, which is given by:

$$\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{249}$$

In our case, we have:

$$\begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix} = a_2 b_3 - a_3 b_2 \tag{250}$$

We do the same for all three basis vectors, which gives us this:

$$\vec{a} \times \vec{b} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \square & a_2 & a_3 \\ \square & b_2 & b_3 \end{vmatrix} - \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & \square & a_3 \\ b_1 & \square & b_3 \end{vmatrix} + \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ a_1 & a_2 & \square \\ b_1 & b_2 & \square \end{vmatrix} \tag{251}$$

$$\vec{a} \times \vec{b} = \begin{vmatrix} a_2 & a_3 \\ b_2 & b_3 \end{vmatrix} \hat{i} - \begin{vmatrix} a_1 & a_3 \\ b_1 & b_3 \end{vmatrix} \hat{j} + \begin{vmatrix} a_1 & a_2 \\ b_1 & b_2 \end{vmatrix} \hat{k} \tag{252}$$

$$\vec{a} \times \vec{b} = (a_2 b_3 - a_3 b_2)\hat{i} - (a_1 b_3 - a_3 b_1)\hat{j} + (a_1 b_2 - a_2 b_1)\hat{k} \tag{253}$$

Note
Be careful, remember that it is the first determinant times $\hat{i}$, then **minus** the second determinant times $\hat{j}$, and then plus the third determinant times $\hat{k}$.

A **matrix** is an array of items (usually numbers), arranged in rows and columns. The dimension of a matrix is how many rows and columns it has. A two-by-three (2×3) matrix, for example, has two rows and 3 columns, like this:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 2 & 1 \end{pmatrix} \tag{254}$$

We denote the rows of the matrix typically with the letter i, and the columns with the letter j, and we can refer to the entries of a matrix with their column and row number. For example, A_{23} is the entry on the 2nd row, 3rd column of the matrix A.

A matrix can be n-dimensional. A higher-dimensional matrix with i rows and j columns would be written like this:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} & \dots & A_{1j} \\ A_{21} & A_{22} & A_{23} & \dots & A_{2j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ A_{i1} & A_{i2} & A_{i3} & \dots & A_{ij} \end{pmatrix} \tag{255}$$

The transpose of a matrix is the matrix with its rows and columns switched, such that:

$$(A_{ij})^T = A_{ji} \tag{256}$$

We can write out a vector as a row vector or column vector, which are respectively matrices with just one row or column:

$$\vec{a} = \begin{bmatrix} a_1 & a_2 & a_3 \end{bmatrix}^T = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{257}$$

Important
Note that while a vector can be arbitrarily written as a row or column vector, a row vector is **not** the same thing as a column vector!

A matrix is a description of a *linear transformation* of the grid. To understand what this means, let's go back to our basis vectors for a moment. Remember that (in 2D space), the basis vectors $\hat{i}$ and $\hat{j}$ are:

$$\hat{i} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \hat{j} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{258}$$

Now, suppose we had a matrix A, as follows:

$$A = \begin{bmatrix} 2 & 1 \\ 5 & 1 \end{bmatrix} \tag{259}$$

This matrix tells us to move the basis vectors $\hat{i}$ and $\hat{j}$ like this:

$$\hat{i} : \begin{bmatrix} 1 \\ 0 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 5 \end{bmatrix} \tag{260}$$

$$\hat{j} : \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 1 \end{bmatrix} \tag{261}$$

By moving the basis vectors, every other point on our original grid moves too, so every vector on the original grid moves with it. So, a matrix is a way to *transform* vectors into new positions by changing the entire grid.

Matrix Operations We can add matrices together if they have the *same* dimensions:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} + \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} A_{11} + B_{11} & A_{12} + B_{12} \\ A_{21} + B_{21} & A_{22} + B_{22} \end{bmatrix} \tag{262}$$

We can multiply any matrix by a scalar:

$$k \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} k(A_{11}) & k(A_{12}) \\ k(A_{21}) & k(A_{22}) \end{bmatrix} \tag{263}$$

And we can multiply matrices with other matrices. To do so, we calculate the dot product of each row vector of matrix A with each column vector of matrix B.

In general, when finding the entry C_{ij} where C is the matrix product of A and B, we take the dot product of the ith row of A with the jth column of B.

This is a bit of a more multi-step process, so let's see how this works, given our two example matrices A and B with the matrix product C:

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \tag{264}$$

First, we take the *first* row vector of A and multiply it by the *first* column vector of B:

$$C_{11} = \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \cdot \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix} = A_{11}B_{11} + A_{12}B_{12} \tag{265}$$

Then, we take the *second* row vector of A and multiply it by the same first column vector of B:

$$C_{21} = \begin{bmatrix} A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix} = A_{21}B_{11} + A_{22}B_{12} \tag{266}$$

If the first matrix had more rows, we'd keep doing this, until we've multiplied every row of the first matrix by the first column of the second matrix.

Now, we take the *first* row vector of A and multiply it by the *second* column vector of B:

$$C_{12} = \begin{bmatrix} A_{11} & A_{12} \end{bmatrix} \cdot \begin{bmatrix} B_{12} \\ B_{22} \end{bmatrix} = A_{11}B_{12} + A_{12}B_{22} \tag{267}$$

Then, we take the *second* row vector of A and multiply it by the same second column vector of B:

$$C_{22} = \begin{bmatrix} A_{21} & A_{22} \end{bmatrix} \cdot \begin{bmatrix} B_{11} \\ B_{12} \end{bmatrix} = A_{21}B_{11} + A_{22}B_{12} \tag{268}$$

We'd continue this process, going through all the columns of the second matrix. Putting it all together, we find the final matrix product.

Since matrix multiplication is dependent on the dot product, we can only multiply two matrices A and B if A has the same number of columns as B has rows. So we can find the matrix product of two 3×3 matrices, or a 5×2 matrix with a 2×5 matrix, but not a 5×2 and 3×3 matrix.

Why is matrix multiplication useful? Well, because remember matrices are transformations that can be applied to vectors. If we want to apply two matrices, A and B, to a vector $\vec{a}$, that's the same as transforming $\vec{a}$ by the matrix product AB.

Using matrices to transform vectors We can use matrices to transform vectors as well, using a modified version of matrix multiplication. The general formula for matrix-vector multiplication is given by:

$$A\vec{x} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}. \tag{269}$$

Consider the following matrix and vector:

$$S = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}, \vec{v} = \begin{bmatrix} 5 \\ 2 \end{bmatrix} \tag{270}$$

If we apply the matrix S on a vector $\vec{v}$, then we end up with a new vector $\vec{v}_2$:

$$S\vec{v} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} 5 \\ 2 \end{bmatrix} = \begin{bmatrix} 10 \\ 4 \end{bmatrix} \tag{271}$$

We can check our answer using NumPy's `dot()`, which performs matrix-vector multiplication when given a matrix and a vector:

```
import numpy as np

a = np.array([[2, 0],
              [0, 2]])
b = np.array([[5],
              [2]])
a.dot(b)
```

Note that the new vector $\vec{v}_2$ is equal to the original vector $\vec{v}_1$ scaled by 2! So the matrix S actually encodes the **linear transformation** of scaling by 2. In fact, all matrices can be thought of as *linear maps* that map vectors onto their transformed versions. Common transformations, such as rotation, shearing, stretching, translation, and countless others can all be encoded through matrices! This is why matrices are useful!

Linearity Linear algebra might seem an unrelated mess of mathematical objects, problems, and techniques. But there is one theme underlying linear algebra - **linearity**. Anything that is linear is acted on by **linear operators**, to which the following rule applies:

$$f(ax + by) = af(x) + bf(y) \tag{272}$$

For example, one linear operator would be vector addition, and another linear operator would be matrix multiplication - both follow this rule. Therefore vectors and matrices are linear as well.

What's the point? Linear algebra has many arbitrary rules - but why learn them? What does linear algebra have to do with the real world, when it feels like you're just moving around a bunch of numbers in columns and rows? The answer is - a **lot**.

For example, matrices are used to solve complicated systems of equations, and find optimal solutions to many problems. Vectors are used to model physical quantities, like force, position, and acceleration. Almost all computer graphics and machine learning uses both, and physics uses both in combination with other mathematics. In fact, the whole reason why linear algebra is called **linear** algebra is that it provides a consistent set of rules that apply to linear problems. Put it another way, any linear problem can be solved with linear algebra!

Extra: motivation for tensors Consider a 3D vector:

$$\vec{A} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 3 \\ -5 \\ 2 \end{bmatrix} \tag{273}$$

This vector is written in Cartesian coordinates (x, y, z coordinates). The same vector would be written like this in spherical coordinates (θ and ϕ are given in radians):

$$\vec{A} = \begin{bmatrix} r \\ \theta \\ \phi \end{bmatrix} = \begin{bmatrix} (x^2 + y^2 + z^2)^{1/2} \\ \arctan(y/x) \\ \arccos(z^2/(x^2 + y^2 + z^2)) \end{bmatrix} = \begin{bmatrix} 6.164 \\ -1.03 \\ 0.324 \end{bmatrix} \tag{274}$$

Notice that these two are the *same* vector, but in different coordinate systems, the components of the vector change. So, a vector's components are **not** invariant. They depend on the coordinate system you use, and converting between coordinate systems is really annoying!

This is where tensors come in. Tensors are generalizations of n-dimensional arrays - including vectors and matrices - which are invariant. This means, we can rewrite a scalar, vector, or matrix as an equivalent tensor that remains the same in whichever coordinate system we put it in. How cool is that! We'll explore tensors some more a few sections later.

Multivariable calculus

> "Give me a lever long enough and a fulcrum on which to place it, and I shall move the world."
> **Archimedes**

Within the first hundred years of the invention of calculus, Newton and Leibnitz had already pushed single-variable calculus to its limit. A new generation of mathematicians had cropped up, ones who yearned to push the envelope of calculus, extending it to multivariable functions. Multivariable calculus allowed for physics, then in its infancy, to finally flourish, bringing with it our modern understanding of the world. And so it is invaluable that we explore this critical tool in the physicist's toolbox, and follow in the footsteps of Lagrange and Euler; seeing further, "standing on the shoulders of giants".

Prerequisites for multivariable calculus

Parametric Equations Parametric equations are equations where x and y are functions of a *parameter* t. For instance, the following is a parametric equation:

$$\begin{cases} x = \sin t \\ y = \cos t \end{cases} \tag{275}$$

Note that $x^2 + y^2 = \sin^2(t) + \cos^2(t) = 1$, so this is the parametric equation of the unit circle, as we will confirm below:

```
import sympy as sp
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
plt.rcParams["font.family"] = "serif"
plt.rcParams['mathtext.fontset'] = 'stix'
plt.rcParams["axes.grid"] = True
from sympy.plotting.plot import *
sp.init_printing()

def plot_p2d_example():
    t = sp.Symbol("t")
    plt = plot_parametric(sp.sin(t), sp.cos(t), show=False, aspect_ratio=(1.0, 1.0))
    plt.show()

plot_p2d_example()
```

The same concept can be generalized with a 3D parametric equation:

```
def plot_p3d_example():
    t = sp.Symbol("t")
    plt = plot3d_parametric_line(sp.sin(t), sp.cos(t), t, show=False)
    plt.show()

plot_p3d_example()
```

Monovariable calculus The derivative of a monovariable function is defined as:

$$\frac{dy}{dx} = \lim_{x \to h} \frac{f(x+h) - f(x)}{h} \tag{276}$$

The indefinite integral of a function is the inverse operation of the derivative:

$$\int \frac{dy}{dx} dx = f(x) \tag{277}$$

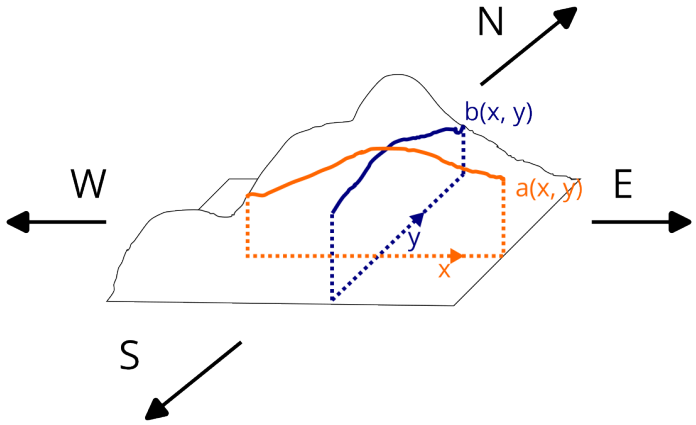
And is defined by:

$$\int f(x)dx = \lim_{\Delta x_a \to 0} \sum_{a=1}^{n} f(x_a)\Delta x_a \tag{278}$$

Definite integrals are the (signed) area under the curve of a function, and can be evaluated by the fundamental theorem of calculus:

$$\frac{dF}{dx} = f(x) \Rightarrow \int_a^b f(x)dx = F(b) - F(a) \tag{279}$$

Partial Derivatives If you were travelling along a mountain, you would notice something interesting. Let's assume you take two journeys across the mountain, measuring the height of the mountain as you go. If you walk strictly from east to west of the mountain, you find that the rate of change of the height is given by a specific function $a(x)$. If you then change directions and walk strictly from north to south of the mountain, then you find that the rate of change of the height is given by a *different* specific function $b(y)$. Clearly these two rates of change are different.



If we call the height of the mountain a function $h(x, y)$, we would call the rate of change of the height in strictly the east-west direction (along x) is the **partial derivative** of the height h with respect to x, which we notate with:

$$a(x, y) = \frac{\partial h(x, y)}{\partial x} \tag{280}$$

And the rate of change of the height in strictly the north-south direction (along y) is the **partial derivative** of the height h with respect to y, which we notate with:

$$b(x, y) = \frac{\partial h(x, y)}{\partial y} \tag{281}$$

Essentially, partial derivatives take the derivative of a multivariable function with respect to one variable and leave all other variables constant. For example, take $f(x, y) = 2x^2y$:

$$\frac{\partial(2x^2y)}{\partial x} = y \cdot \frac{d(2x^2)}{dx} = y \cdot 4x = 4xy \tag{282}$$

Here, we treat y as a constant, allowing us to factor it out of the equation as a constant, and then we can just take the ordinary derivative of $2x^2$. Similarly:

$$\frac{\partial(2x^2y)}{\partial y} = 2x^2 \cdot \frac{d(1y)}{dy} = 2x^2 \cdot 1 = 2x^2 \tag{283}$$

Here, we treat x as a constant, allowing us to factor everything in terms of x (that being $2x^2$) out of the equation as a constant, and then we can just take the ordinary derivative of $1y$.

Formally, the partial derivative of a multivariable function $f(x_1, x_2, x_3, \ldots, x_i, \ldots, x_n)$ is defined by:

$$\frac{\partial f}{\partial x^i} = \lim_{a \to 0} \frac{f(x_1, \ldots, (x_i + a), \ldots, x_n) + f(x_1, \ldots, (x_i), \ldots, x_n)}{a} \tag{284}$$

For a two-variable function, such as $f(x, y)$, the partial derivative with respect to x is given by:

$$\frac{\partial f}{\partial x} = \lim_{a \to 0} \frac{f(x + a, y) - f(x, y)}{a} \tag{285}$$

And the partial derivative with respect to y is given by:

$$\frac{\partial f}{\partial y} = \lim_{a \to 0} \frac{f(x, y + a) - f(x, y)}{a} \tag{286}$$

The **multivariable chain rule** for derivatives of a composite function is based on the single-variable version. For a function $f(x, y) = (x(t), y(t))$, the derivative with respect to t is given by:

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} + \frac{\partial f}{\partial y}\frac{dy}{dt} \tag{287}$$

For instance, we could calculate the partial derivative of $f(x, y) = 2x^2y$ with respect to x. Let's say that $x(t) = 3t^5 + 5$ and $y(t) = 7\sqrt{t}$. We can then calculate our derivative with respect to t like so:

$$\frac{df}{dt} = \frac{\partial(2x^2y)}{\partial x}\frac{d(3t^5 + 5)}{dt} + \frac{\partial(2x^2y)}{\partial y}\frac{d(7\sqrt{t})}{dt} \tag{288}$$

$$= 4xy \cdot 15t^4 + \frac{2x^2}{2\sqrt{t}} \tag{289}$$

$$= \frac{x\left(60t^{\frac{9}{2}}y + x\right)}{\sqrt{t}} \tag{290}$$

The more general form of the chain rule with a multivariable function $f(q_1, q_2, q_3, \ldots, q_n) = (q_1(t), q_2(t), q_3(t), \ldots, q_n(t))$ is given by:

$$\frac{df}{dt} = \sum_i^n \frac{\partial f}{\partial q^i}\frac{dq^i}{dt} \tag{291}$$

And if both the outer and inner functions are multivariable:

$$\frac{\partial f}{\partial t} = \sum_i^n \frac{\partial f}{\partial q^i}\frac{\partial q^i}{\partial t} \tag{292}$$

Mixed partial derivatives Second partial derivatives are denoted by $\frac{\partial^2 f}{\partial x^2}$ or f_{xx}. When using the "del" notation (with the ∂ symbol) you take derivatives from right to left order.

Second partial derivatives **commute**, which means the order you take them doesn't matter. Thus $\frac{\partial f}{\partial x \partial y} = \frac{\partial f}{\partial y \partial x}$.

Scalar-valued functions A scalar-valued function is a function that always outputs a number for each input, not a vector, such as $f(x, y) = 2x + 3y^2$.

Scalar field A scalar field is when a scalar-valued function gives the value of every point in space. An example of a scalar field could be a temperature field; then the temperature at each point (x, y, z) in space is a number given by the function $T(x, y, z)$.

Vector-valued functions Vector-valued functions produce a vector for each input, for instance:

$$\vec{f}(t) = \begin{bmatrix} t^2 + 2t \\ \sin(2t) + t \end{bmatrix} \tag{293}$$

Note that vector-valued functions can have components that are functions of x, y, z, or functions of t, in which case its components are **parametric functions**.

The derivative of a vector-valued function is another vector in all cases. For example, velocity is often given by a vector-valued function where:

$$\vec{v}(t) = \begin{bmatrix} v_x(t) \\ v_y(t) \\ v_z(t) \end{bmatrix} = \begin{bmatrix} x'(t) \\ y'(t) \\ z'(t) \end{bmatrix} \tag{294}$$

Speed is given by the norm of the velocity function:

$$v = \sqrt{v_x^2 + v_y^2 + v_z^2} \tag{295}$$

If the vector-valued function is of one variable, e.g. $\vec{v}(t)$, then its derivative is a regular derivative $\frac{d\vec{v}}{dt}$ that is a vector. If the vector-valued function is of several variables, e.g. $\vec{v}(s,t)$, then it has one partial derivative for each variable, e.g. $\frac{\partial \vec{v}}{\partial s}$ and $\frac{\partial \vec{v}}{\partial t}$, and each of those partial derivatives is also a vector.

Vector field A vector field is when a vector-valued function gives the value of every point in space. An example of a vector field could be a wind velocity field; then the wind velocity vector at each point (x, y, z) in space is a number given by the function $\vec{W}_v(x, y, z)$.

Div, Grad, Curl The **gradient** of a multivariable function takes scalar-valued function (or more generally scalar field) and produces a vector field. The vector produced follows 2 attributes:

- Its direction is in the direction of greatest increase
- Its magnitude is proportional to the steepness (rate of increase)

In Cartesian coordinates, the gradient of a function $f(x, y)$ is defined using the nabla (∇) symbol as follows:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \tag{296}$$

For example, let's take the gradient of $f(x, y) = x^2 - y^2$. The gradient at each point (x, y, z) would be then given by:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} = \begin{bmatrix} 2x \\ -2y \end{bmatrix} \tag{297}$$

Or more generally, of a function $f(q_1, q_2, q_3, \ldots, q_n)$:

$$\nabla f = \begin{bmatrix} \frac{\partial f}{\partial q_1} \\ \frac{\partial f}{\partial q_2} \\ \frac{\partial f}{\partial q_3} \\ \vdots \\ \frac{\partial f}{\partial q_n} \end{bmatrix} \tag{298}$$

To intuitively understand the gradient, take a look at the graph of $f(x, y)$:

```
def plot_fxy():
    x, y = sp.symbols("x y")
    f = x ** 2 - y ** 2
    plot3d(f)
```

```
plot_fxy()
```

Let's now take a look at its gradient:

```
def gradient(f):
    x, y = sp.symbols("x y")
    return (f.diff(x), f.diff(y))

def plot_vecfield(g, description):
    x, y = sp.symbols("x y")
    xrange = np.linspace( -3,3,15)
    yrange = np.linspace( -3,3,15)
    X,Y = np.meshgrid(xrange, yrange)

    U=X
    V=Y

    for i in range(len(xrange)):
        for j in range(len(yrange)):
            x1 = X[i,j]
            y1 = Y[i,j]
            U[i,j] = g[0].subs({x:x1, y:y1})
            V[i,j] = g[1].subs({x:x1, y:y1})

    plt.quiver(X,Y,U,V, linewidth=1)
    plt.title(description)
    plt.show()

def fxy_plt_vecfield():
    x, y = sp.symbols("x y")
    f = x ** 2 - y ** 2
    g = gradient(f)
    plot_vecfield(g, "Gradient of $f(x, y) = x^2 - y^2$")

fxy_plt_vecfield()
```

Note how the vectors all point outwards from the center towards the "hills" of the function, the direction of greatest increase.

The **directional derivative** is built on the gradient and gives the rate of change of a function with respect to any direction $\vec{v}$, rather than just the x, y, z directions. It is defined as follows:

$$\nabla_{\vec{v}} f(x,y) = \frac{\partial f}{\partial \vec{v}} = \nabla f \cdot \vec{v} \tag{299}$$

> **Note**
> Remember that this is a **dot product**, not just multiplication! The output of the directional derivative has always to be a scalar function!

This means that given a vector $\langle a, b \rangle$, then the directional derivative at a point (x, y) is given by:

$$\nabla_{\vec{v}} f(x,y) = a\frac{\partial f}{\partial x} + b\frac{\partial f}{\partial y} \tag{300}$$

We cannot take the gradient of a vector field. However, there are two other operations we can do on a vector field - they are the **curl** and **divergence**, notated as follows:

$$\text{grad}\,(f) = \nabla f \tag{301}$$
$$\text{div}\,(f) = \nabla \cdot f \tag{302}$$
$$\text{curl}\,(f) = \nabla \times f \tag{303}$$

The divergence is defined as follows:

$$\nabla \cdot F = \left\langle \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right\rangle \cdot \langle \vec{F}_x, \vec{F}_y, F_z \rangle = \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} \tag{304}$$

The divergence can be thought of as describing fluid flow - if the divergence is positive, fluid is flowing outwards, whereas if the divergence is negative, fluid is flowing inwards.

Meanwhile, the curl, a measure of fluid rotation, is defined much like the typical cross product:

$$\nabla \times F = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ F_x & F_y & F_z \end{vmatrix} \tag{305}$$

Which, when expanded out, results in:

$$\nabla \times F = \left\langle \frac{\partial F_z}{\partial y} - \frac{\partial F_y}{\partial z}, \frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}, \frac{\partial F_y}{\partial x} - \frac{\partial F_x}{\partial y} \right\rangle \tag{306}$$

> **Note**
> Notice that the center term of the curl using the normal cross product formula is actually $-\left(\frac{\partial F_z}{\partial x} - \frac{\partial F_x}{\partial z}\right)$, which we just distributed the negative sign to get $\frac{\partial F_x}{\partial z} - \frac{\partial F_z}{\partial x}$

The Laplacian operator is the gradient of the divergence of a scalar field, given as follows:

$$\text{lapl}\,(f) = \nabla \cdot \nabla f = \nabla^2 f = \frac{\partial^2 F_x}{\partial x^2} + \frac{\partial^2 F_y}{\partial y^2} + \frac{\partial^2 F_z}{\partial z^2} \tag{307}$$

Multiple integrals We are already familiar with the idea of an integral along a single-variable function $f(x)$:

$$\int_a^b f(x)dx = \lim_{\Delta \to 0} \sum f(x_i)\Delta x \tag{308}$$

Here, we are adding thin slices of width dx and height $f(x_i)$ to find the area under the curve of $f(x)$. We can generalize this to two-variable functions $f(x, y)$:

$$\iint_R f\,(x, y)\; dA = \lim_{\Delta \to 0} \sum f(x_i, y_i)\Delta A \tag{309}$$

This is called a **double integral**, and it finds the volume under the *surface* $f(x, y)$ by adding thin *cubes* of area dA and height $f(x_i, y_i)$ within the bounds R.

To evaluate double integrals, we use a similar process as partial derivatives: we integrate first with respect to y, then with respect to x:

$$\iint_R f(x, y)dA = \int_{x_0}^{x_1} \int_{y_0}^{y_1} f(x, y)\, dy\, dx \tag{310}$$

For example, let's evaluate:

$$\iint_R 6xy^2 dA, x \in [2, 4], y \in [1, 3] \tag{311}$$

This becomes:

$$\int_2^4 \int_1^2 6xy^2 \, dy \, dx \tag{312}$$

Let's first evaluate the inner integral. Treating everything that doesn't contain y as constant, we have:

$$\int_1^2 6xy^2 dy = (2xy^3)\Big|_1^2 = 2x(2^3) - 2x(1^3) = 16x - 2x = 14x \tag{313}$$

Here we integrated $6xy^2$ with respect to y, then substituted the resulting y's in the answer for the bounds of integration. Now, let's plug this into our second integral:

$$\int_2^4 14x dx = (7x^2)\Big|_2^4 = 84 \tag{314}$$

The same procedure is true for **triple integrals** over a region $f(x, y, z)$, where we find the volume under a 4-dimensional surface by adding thin 4D regions of volume dV and height $f(x_i, y_i, z_i)$:

$$\iiint_E f(x, y, z) dV = \int_{x_0}^{x_1} \int_{y_0}^{y_1} \int_{z_0}^{z_1} f(x, y, z) \, dz \, dy \, dx \tag{315}$$

> **Note**
> In physical applications, when double and triple integrals are integrated over space, a triple integral is often called a *volume integral* and a double integral is often called an *area integral*. We will be using this terminology from this point on.

Line integrals Similar to integrals over a 1D line, 2D area, or 3D volume, we can take integrals over a curve - these are called **line integrals**. There are two types of line integrals - ones over scalar fields, and ones over vector fields. Line integrals over **scalar fields** take the form:

$$\int_C f(r) \, ds \tag{316}$$

They can be evaluated with:

$$\int_C f(x, y) \, ds = \int_a^b f(x(t), y(t)) \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} dt \tag{317}$$

Or in the 3D scalar field case:

$$\int_C f(x, y, z) \, ds = \int_a^b f(x(t), y(t), z(t)) \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2 + \left(\frac{dz}{dt}\right)^2} dt \tag{318}$$

For example, we can evaluate the line integral, with a curve C with endpoints $(1, 2)$, and $(4, 7)$ (source):

$$\int_C y e^x ds \tag{319}$$

To do this, we must parameterize C with two parametric equations:

$$x(t) = 1 + 3t \tag{320}$$
$$y(t) = 2 + 5t \tag{321}$$

This means that at $t = 0$, $C = (1, 2)$, and at $t = 1$, $C = (4, 7)$. We now take their derivatives:

$$\frac{dx}{dt} = 3\frac{dy}{dt} \qquad = 5 \tag{322}$$

We now subsitute into the equation for scalar 2D line integrals:

$$\int_C ye^x ds = \int_0^1 (2+5t)e^(1+3t)\sqrt{3^2+5^2}dt = \frac{16}{9}\sqrt{34}e^4 - \frac{1}{9}\sqrt{34}\,e \tag{323}$$

The main application for scalar line integrals is to calculate the mass, moment of inertia, and center of mass of a wire. However, vector line integrals are far more common, and in many cases, far more useful.

Line integrals over **vector fields** take a different form. They look like this:

$$\int_C \vec{F} \cdot d\vec{r} \tag{324}$$

And are evaluated with:

$$\int_C \vec{F} \cdot d\vec{r} = \int_a^b \vec{F}(\vec{r}(t)) \cdot \vec{r}'(t)dt \tag{325}$$

Note that the dot is not multiplication, but a **dot product**. In evaluating a line integral over a vector field, both $\vec{F}$ and $\vec{r}(t)$ *must* be known to solve.

The main application of vector line integrals is in various physical laws - for example, the **work** done by a vector field $\vec{F}$ (such as a gravitational or magnetic field) on a particle that travels along the curve C through that field is given by a vector line integral:

$$W = \int_C F \cdot d\vec{r} \tag{326}$$

Meanwhile, the integral forms of two of **Maxwell's equations** all use vector line integrals. For example, Ampère's law states that the line integral of a magnetic field $\vec{B}$ is proportional to the enclosed current:

$$\oint_C \vec{B} \cdot d\vec{r} = \mu_0 I \tag{327}$$

Note that here, we use a special symbol to denote that the curve C enclosing the magnetic field is *closed*. Additionally, the more generalized version of Ampère's law is slightly more complex:

$$\oint_C \vec{B} \cdot d\vec{r} = \mu_0 \left(I + \epsilon_0 \frac{\partial \Phi_E}{\partial t} \right) \tag{328}$$

Additionally, Faraday's law states that the line integral of an electric field $\vec{E}$ is proportional to the rate of change of the magnetic flux:

$$\oint_C \vec{E} \cdot d\vec{r} = -\frac{\partial \Phi_B}{\partial t} \tag{329}$$

Surface integrals Surface integrals generalize the idea of a line integral to surfaces. They can be defined over scalar fields or vector fields. I will add more to this section later. For now though, just note that they look very similar to double integrals, just the integration is of an infinitesimal portion of surface dS:

$$\iint_\Sigma f(x,y)dS \tag{330}$$

The most important application of surface integrals is in the integral form of **Maxwell's equations** - the first two equations extensively utilize surface integrals. Gauss's law states that the surface

integral of the electric field over a closed surface (often, a sphere) is proportional to the enclosed charge:

$$\Phi_E = \iint\limits_{\Sigma_{\text{closed}}} \vec{E} \cdot d\vec{S} = \frac{Q}{\epsilon_0} \tag{331}$$

And Gauss's law for magnetism states that the surface integral of the magnetic field over a closed surface is zero:

$$\iint\limits_{\Sigma_{\text{closed}}} \vec{B} \cdot d\vec{S} = 0 \tag{332}$$

An aside on integral notation The multivariable integrals we have encountered have a variety of different notations, and it is easy to get confused and think that different notations of the same type of integral are actually different types of integral.

To show what we're talking about, consider a *volume* integral:

$$\iiint \rho \, dV = \iiint \rho(x, y, z) \, dx \, dy \, dz \tag{333}$$

This integral could *also* be written as $\int \rho \, dV$ or (in some texts) $\int \rho \, d^3x$ or even $\int \rho \, dv$. These notations are *completely equivalent*. The three-integral-sign notation is just for conceptual clarity; it reminds you that we are integrating over 3 dimensions when explicitly evaluating the volume integral. The same applies for area integrals - you can write it with one or two integral signs, both are equivalent.

Similarly, it is common to write surface integrals using just a *single* integral sign, e.g. as $\int \mathbf{F} \cdot d\mathbf{S}$ or $\oint \mathbf{F} \cdot d\mathbf{S}$ for a closed surface integral. It is *also* common to write $\iint \mathbf{F} \cdot d\mathbf{A}$ or $\iint \mathbf{F} \cdot d\mathbf{a}$, "a" for (surface) "area". The difference is only notational; there is no true difference between these different ways of writing the same surface integral. In addition, using $\vec{F}$ or $d\vec{A}$ is also perfectly acceptable. It simply matters that a surface vector is notated with the *integrand* and the *differential (surface) element* as **both vectors**.

The same goes for line integrals. Scalar line integrals are often just written $\int f(x, y, z) \, ds$, with our notation $\int_C f \, ds$ to illustrate that the line integral is over a particular curve, or $\oint_C f ds$ to illustrate that the line integral is over a *closed* curve. Vector line integrals are variously notated $\int \mathbf{F} \cdot d\mathbf{s}$ or $\int \mathbf{F} \cdot d\mathbf{r}$, and their equivalents using $\vec{F}$ and $d\vec{s}$ or $d\vec{r}$, with $\oint \mathbf{F} \cdot d\mathbf{s}$ or $\oint \mathbf{F} \cdot d\mathbf{r}$ used for the closed curve equivalents.

The takeaway from this aside is that *notation is not consistent* and this is a fact to be aware of when exploring external resources. The idea is to be familiar with a wide variety of different vector calculus notations, so that you don't get mystified when seeing a new and unconventional notation.

Vector calculus theorems There are several important vector calculus theorems. First, the divergence theorem, which relates surface and triple integrals:

$$\iiint\limits_{\Omega} (\nabla \cdot \vec{F}) \, dV = \iint\limits_{\Sigma} \vec{F} \cdot d\vec{S} \tag{334}$$

Then, Stokes' theorem, which relates surface integrals around a region and line integrals enclosing that region:

$$\iint\limits_{\Sigma} (\nabla \times \vec{F}) \, dS = \oint_C \vec{F} \cdot d\vec{r} \tag{335}$$

Differential equations

> "Science is a differential equation. Religion is a boundary condition."
> **Alan Turing**

Differential equations are simultaneously often regarded as some of the coolest and strangest objects in physics. On one hand, they're ubiquitous, and nearly every physics theory is expressed using them. On the other, they have a tendency to be unsolvable and difficult to understand. It is hoped that this chapter will bring all their positive qualities into the limelight, and make differential equations no longer scary or intimidating, but descriptions of nature unrivaled in their beauty.

What is a differential equation? Differential equations are any equations that describe a function in terms of how it changes through space or time. For instance, we could have:

$$\frac{dy}{dt} = ky \tag{336}$$

The interpretation of this equation is that y increases proportional to its derivative, so as y increases, the rate of change of y increases by a proportional amount.

Differential equations can be of a single-variable function and its derivatives, or a multivariable function and its partial derivatives. For instance, the wave equation is given by:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2} \tag{337}$$

Here, $u = u(x, t)$, and is the function describing the wave. The interpretation of the wave equation is that the rate of change of the rate of change of u moving along the x direction is proportional to the rate of change of the rate of change of u as time passes.

Differential equations that only involve single-variable functions are called **ordinary differential equations** (ODEs), while those that involve multivariable functions (and their partial derivatives) are called **partial differential equation** (PDEs).

Initial-value and boundary-value problems We are interested in solving differential equations to be able to perform further mathematical analysis of the physics of a given system. There are a myriad number of ways to solve differential equations, and we will cover just a few in the following sections. In general, however, finding an *exact* solution to a particular system cannot be done with knowledge of just the differential equation; typically, *some* data must be provided before a solution can be found.

In the case of ODEs in the form $\frac{dy}{dt} = f(t, y)$, this data is called the *initial condition*, the state of $y(t)$ at $t = 0$. Some physical examples of initial conditions are the initial velocity or initial position. We often denote such an initial condition y_0. Once we know that, we can calculate the values of y that the differential equation predicts for all future times t. The combination of an ODE and an initial value is known as an **initial-value problem (IVP)**.

Boundary-value problems can be thought of as an extension of initial value problems to partial differential equations (PDEs). Unlike initial counditions, boundary value problems usually specify the *function* that the solution to the PDE takes at the boundaries. Such functions are called *boundary conditions* (BCs). A physical example may be some water sloshing in a water tank; a PDE may be solved to find the function describing the distribution of water within the tank. The boundary condition would then be the height of the water at the walls of the tank.

The three main types of boundary conditions are Dirichlet, Neumann, and Robin. You can mix and use different boundary conditions together, especially for boundary-value problems that have different types of boundaries, but individually they are generally in one of these forms:

Type of BC	Example mathematical form
Dirichlet	$y = f$
Neumann	$\frac{\partial y}{\partial x} = f$
Robin	$Ay + B\frac{\partial y}{\partial x} = f$
Cauchy	Combination of Dirichlet and Neumann

Physical significance of different types of boundary conditions The physical intepretation of a Dirichlet boundary condition is that the physical quantity is fixed or constrained at the boundary, that is, it takes a specific constant value. In the special case where $u = 0$ at the boundary, the physical quantity vanishes at the boundary.

Meanwhile, the physical interpretation of a Neumann boundary condition is that the physical quantity is kept within the boundary. This corresponds to insulating or reflecting boundaries that prevent the physical quantity from flowing or radiating outwards.

Robin boundary conditions have more flexible physical interpretations. One specific type of robin boundaries is an *open* boundary. The physical interpretation of an open boundary condition is that the physical quantity is allowed to flow undisturbed outwards beyond the boundary. This corresponds to boundaries that allow for outward radiation or free propagation through them. It is a specific type of Robin boundary condition.

Some supplementary information
Boundary conditions generally do not involve specifying more than the value(s) of the first partial derivatives and the function at the boundaries. This is because the majority of PDEs that govern the universe don't have any higher derivatives of higher than second-order. The main equations we will be solving are either first-order or second-order.

Solving initial- and boundary-value problems Initial-value problems can be solved by hand in some cases, but for cases where they cannot be solved by hand, computational methods can be used to solve them numerically (this results in an approximate solution). We will explore how to do so in the numerical methods section in Chapter 3. In addition, there exist online calculators that numerically solve differential equations: see Bluffton University's free tool at this link for solving IVPs.

Boundary-value problems can be solved by hand in vastly fewer cases, and even when a solution is possible to find by hand, many simplifying assumptions must be used. For this reason, numerical methods are required for the majority of boundary-value problems. Again, we will explore this further in Chapter 3, but for those interested, the web app Visual PDE provides an easy-to-use graphical interface for solving PDEs numerically. We recommend you try it out!

Solving differential equations When we are told to solve a differential equation, what we are doing is to figure out y from the differential equation. How we can do this depends on the type of differential equation. For example, consider an ODE in the form:

$$\frac{dy}{dt} = f(y)g(t) \tag{338}$$

If we can put any ODE in this form, it is considered **separable**. What we can then do is to make sure each side of an equation is expressed only in terms of one variable:

$$\frac{1}{f(y)}dy = g(t)dt \tag{339}$$

And finally, to integrate both sides to solve:

$$\int \frac{1}{f(y)}dy = \int g(t)dt \tag{340}$$

Partial differential equations are considered separable by a similar criteria: if you can express each side of them only in terms of one variable, then they are separable.

But enough theory! Let's actually try solving two differential equations, one ODE and one PDE. The ODE we will be solving is the exponential growth equation - the reason for that name will become apparent very soon. It is given by:

$$\frac{dy}{dx} = ky \tag{341}$$

Here, k is just a constant. By multiplying dx to both sides of the equation, we have:

$$dy = kydx \tag{342}$$

And now, by dividing y from both sides, we have:

$$\frac{1}{y}dy = kdx \tag{343}$$

We can now integrate both sides to solve:

$$\int \frac{1}{y}dy = \int kdx \tag{344}$$

Which results in:

$$\ln(y) = kx + C \tag{345}$$

We raise both with e^x, to get:

$$e^{\ln(y)} = e^{kx+C} \tag{346}$$

Which becomes:

$$y = e^{kx+C} \tag{347}$$

Just one more step! Now that:

$$e^{kx+C} = e^C e^{kx} \tag{348}$$

If we define a new constant $C_2 = e^C$, then:

$$y = C_2 e^{kx} \tag{349}$$

This is called the **general solution** to the exponential growth equation, because C_2 can be any number, and so this general solution encodes all possible solutions each with their own value of C_2.

To actually solve it for a value, we need an **initial value**. For instance, we may be told that $y(0) = 1$. If we plug that in:

$$1 = C_2 e^{k(0)} \tag{350}$$

$$C_2 = 1 \tag{351}$$

So our unique solution given our initial value is now:

$$y = e^{kx} \tag{352}$$

We have now solved the **initial value problem** - finding the solution to the differential equation given the provided initial condition.

Note that C_2 was the same number as $y(0)$. Therefore, we have a new interpretation of C_2 - note that C_2 is the **initial value** of a function, so $C_2 = y_0$, and we can rewrite the general solution as:

$$y = y_0 e^{kx} \tag{353}$$

The PDE we will be solving is the 1D **heat equation**, given by:

$$\frac{\partial u}{\partial t} = \alpha^2 \frac{\partial^2 u}{\partial x^2} \tag{354}$$

This equation models the temperature function $u(x,t)$ of a long, thin rod, where the value of u at a given position x and a given time t is the temperature. Note that α is just a constant. Third nice fact about the equation: it's separable! To separate, we first write $u(x,t)$ as a product of two functions:

$$u(x,t) = f(x)g(t) \tag{355}$$

Therefore, using this definition, we can compute the partial derivatives of u:

$$\frac{\partial u}{\partial t} = f(x)g'(t) \tag{356}$$

$$\frac{\partial^2 u}{\partial x^2} = f''(x)g(t) \tag{357}$$

So we now have:

$$fg' = \alpha^2 f''g \tag{358}$$

If we rearrange this equation, we get:

$$\frac{g'}{g\alpha^2} = \frac{f''}{f} \tag{359}$$

Now, note that the left hand side has derivatives and functions dependent on t, but the right hand side has derivatives and functions dependent on f. If derivatives with respect to different variables are constant, then the two sides of the equation must be equal to a constant, which we'll call λ:

$$\frac{g'}{g\alpha^2} = \frac{f''}{f} = -\lambda \tag{360}$$

Why the negative sign? Since a negative sign applied to a constant makes it still a constant, so we can technically do what we want to λ - scale it, add another constant to it, make it positive or negative, the math still works out. The only difference is that $-\lambda$ makes the resulting differential equations way easier to solve, which is why we're adding a negative sign.

So, we have two ODEs instead of our original PDE to solve:

$$g' = -\lambda\alpha^2 g \tag{361}$$

$$f'' = -\lambda f \tag{362}$$

The first equation is separable, and, after going through the steps to manually solve the differential equation, the result is:

$$g(t) = C_1 e^{-\lambda a^2 t} \tag{363}$$

The second equation requires a little bit more work. Which functions are equal to negative of their second derivative? On inspection, we can guess that the function $\sin x$ would work, and indeed it does work when we compute its second derivative:

$$\frac{d^2}{dx^2}\sin x = -\sin x \tag{364}$$

However, we want the second derivative to not be equal to the function, but **proportional** to it. Thus, perhaps a good guess would be $\sin(\lambda x)$. But note that:

$$\frac{d^2}{dx^2}\sin(\lambda x) = -\lambda^2 \sin(\lambda x) \tag{365}$$

So to make the proportionality constant agree with our differential equation, we want to square root λ, instead, so:

$$\frac{d^2}{dx^2}\sin(\sqrt{\lambda}x) = -\lambda \sin(\sqrt{\lambda}x) \tag{366}$$

To make this equation more general, since multiplying by a constant does not affect proportionality, we can scale $\sin(\sqrt{\lambda}x)$ by an arbitrary constant C_2. So we have solved our second differential equation:

$$f(x) = C_2 \sin(\sqrt{\lambda}x) \tag{367}$$

Now, we simply need to combine the two solutions together:

$$u(x,t) = f(x)g(t) = C_2 \sin(\sqrt{\lambda}x) C_1 e^{-\lambda\alpha^2 t} \tag{368}$$

Since we have two multiplied constants, we can rewrite them as a third constant where $C_3 = C_2 C_1$:

$$u(x,t) = C_3 \sin(\sqrt{\lambda}x) e^{-\lambda\alpha^2 t} \tag{369}$$

This is our general solution of the 1D heat equation!

Laplace and Fourier transforms The next category of methods to solve differential equations involves Laplace and Fourier transforms. These are two transforms that take an expression of one variable to be expressed in terms of a different variable. For a given function $f(t)$, the Laplace transform results in a new function $g(s)$:

$$g(s) = \int_0^\infty f(t) e^{-st} dt \tag{370}$$

And the Fourier transform also results in a new function $h(s)$:

$$h(s) = \int_{-\infty}^\infty f(t) e^{-2\pi i s t} dt \tag{371}$$

The main idea behind these transforms is that they allow ordinary differential equations in terms of t to become algebraic equations in terms of s. They also allow partial differential equations in terms of t to become algebraic or ordinary differential equations in terms of s. Then, the unknown function can be algebraically solved for, and an inverse transform can be taken to find the solution in terms of t.

Numerical methods and approximations While many differential equations can be solved exactly, not all differential equations are so straightforward to solve. Instead, most differential equations are typically not solved directly.

There are three common alternatives if a differential equation is resistant to separation of variables or the Laplace and Fourier transforms:

- "Guess and check"
- Taylor series approximation
- Numerical solving (often using computers)

The "guess and check" approach, also known as the "method of inspired guessing", is literally that - given knowledge of functions and their derivatives, guess a solution to the differential equation. For instance, suppose we had the differential equation:

$$\frac{d^2x}{dt^2} = 0 \tag{372}$$

From basic analysis of this differential equation, we know that the original function must be of a degree less than 2, because otherwise its second derivative wouldn't vanish to zero. Thus, we can guess that it is some type of linear function. And indeed, if we take the second derivative of a linear function, it does yield zero. So the solution is:

$$x(t) = at + b \tag{373}$$

We can also use a Taylor series to approximate solutions to a differential equation. For instance, we could use it to approximate $r(t)$ from Newton's gravitational force equation:

$$\frac{d^2r}{dt^2} = -\frac{GM}{r^2} \tag{374}$$

We aim to solve this with a 4th-order Taylor series for the Earth and the Sun. We have the initial conditions $r(0) = r_0$ and $r'(0) = v_0$, where r_0 is the mean distance from the Earth to the Sun (1 AU), and $r'(0)$ is equal to $\frac{2\pi r_0}{T}$, where T is the Earth's period. A 4th-order Taylor polynomial is given by:

$$r(t) = r(0) + r'(0)t + \frac{r''(0)}{2}t^2 + \frac{r'''(0)}{6}t^3 \tag{375}$$

We know that $r(0) = r_0$, and $r'(0) = v_0$. We also know that $r''(0) - \frac{GM}{r^2} = -\frac{GM}{r_0^2}$. Finally, $r''' = \frac{d}{dt}r''$, such that:

$$r''' = \frac{d}{dt}\left(-\frac{GM}{r^2}\right) = \frac{2GM}{r^3}\frac{dr}{dt} \tag{376}$$

Thus, $r'''(0) = \frac{2GM}{r_0^3}v_0$, so the final Taylor polynomial is:

$$r(t) = r_0 + v_0t - \frac{GM}{2r_0^2}t^2 + \frac{2v_0GM}{r_0^3}t^3 \tag{377}$$

This is 4th-order Taylor expansion for the solution $r(t)$ of the differential equation, and will successfully approximate the solution, so long as t is close to zero.

The final method of solving differential equations that cannot be easily solved using any other means is by using **numerical methods**. Numerical methods take the initial conditions of a differential equation and calculate a tiny change dy in a function caused by a small step along the function's input dx. Then, they add that little change dy to the existing value of y. By doing this process many, many times, over and over, an entire solution to a differential equation can be approximated.

To find the formula for one type of numerical method, the Euler method, consider the definition of the derivative, just with the limit removed:

$$f'(x) = \frac{f(x+h) - f(x)}{h} \tag{378}$$

Let's solve for $f(x+h)$ in this equation:

$$f(x+h) = hf'(x) + f(x) \tag{379}$$

This means that given a current value of a function f, the next value of f is given by:

$$f_{n+1} = hf'(x) + f_n \tag{380}$$

This method is tedious to do by hand, but computers can do it very quickly. More accurate types of numerical methods, including the very popular Runge-Kutta methods, are similar in nature, only they break each step into smaller steps for more precision.

Summary of Differential Equations The great paradox of differential equations is that they can be ridiculously easy to solve, or ridiculously hard to solve. Using just pen-and-paper techniques, differential equations require lots of creativity and imaginative approaches to be solved, and often require simplifying the problem or special cases of problems. But with brute-force computer solving, differential equations can be simplified into much easier problems, albeit problems that require a lot of steps and computing power. In Project Elara, the majority of differential equation solving will be done numerically, but knowing the analytic techniques will certainly be helpful as well. That said, enough on differential equations - let's get back into physics!

0.1.3 Introductory physics

Previously, we have reviewed the mathematical concepts that are crucial to our next area of study: physics. Physics is **not** simply applied math, because it also relies on observation, scientific reasoning, and systematic analysis. Nearly every part of the project requires, to some extent, applying physics and physical principles.

To have a good grasp of physics is essential, but it can certainly be a very intimidating subject. We will go through it step-by-step, and we will limit ourselves (in this section) to physics at a basic level. It may be helpful to re-read sections that are confusing or hard to understand. We hope that this section will prepare you well for more advanced topics ahead.

Newtonian mechanics

> "The important thing is to never stop questioning."
> **Albert Einstein**

For over 200 years, before the advent of 20th century relativistic and quantum physics, our view of the world was dominated by classical mechanics. From the foundation laid by Newton to the work of Lagrange and Euler and the masterminds of Faraday and Maxwell, classical mechanics set out to describe the world around us through concrete, physical laws, and still does so brilliantly. This guide aims to explain classical mechanics in as simple and elegant a way as possible, hopefully allowing you to soon have a working knowledge of mechanics with which to do physics.

Kinematics We can describe the location of a point through what is known as a position function. The position function $s(t)$ describes the location of an object through space at any given time. For example, imagine a position function as follows:

$$s(t) = 3t^3 - 5t + 1 \tag{381}$$

We adopt the convention in physics that positive numbers are used for positions to the top or right, and negative numbers are used for positions to the left or bottom.

At $t = 0$, $s(t) = 1$, which means that the object is 1 meter to the right of the origin at that time. At $t = 1$, $s(t) = -1$, which means the object is 1 meter to the left of the origin. At $t = 2$, $s(t) = 15$, which means the object is 15 meters to the right of the origin.

Sometimes, we like to specify that $s(t)$ is moving along the x-direction, so we call it $x(t)$ instead. We can also specify that $s(t)$ is moving along the y-direction by calling it $y(t)$. The position function remains the same, just the direction of travel becomes different.

Velocity is the derivative of position with respect to time. That is:

$$v = \frac{ds}{dt} \tag{382}$$

For example, given that $s(t) = 3t^3 - 5t + 1$, $v(t) = 9t^2 - 5$, as it is the derivative of $s(t)$.

Acceleration is the derivative of velocity, which is the second derivative of position. That is:

$$a = \frac{dv}{dt} = \frac{d^2s}{dt^2} \tag{383}$$

From the same definitions, velocity is the integral of acceleration, and position is the integral of velocity:

$$v = \int adts \qquad\qquad = \int vdt \tag{384}$$

Lastly, **displacement**, denoted by Δs (or Δx or Δy) is the change in position between 2 times:

$$\Delta s = s(b) - s(a) \tag{385}$$

1D projectile motion The **kinematic equations** describe the motion of objects under constant acceleration. They are given as follows:

$$v = v_0 + at \tag{386}$$

$$\Delta s = \left(\frac{v + v_0}{2}\right) \tag{387}$$

$$\Delta s = v_0 t + \frac{1}{2}at^2 \tag{388}$$

$$v^2 = (v_0)^2 + 2a\Delta x \tag{389}$$

The kinematic equations are especially helpful when analyzing objects in free-fall, as all objects in free-fall - called projectiles - have a constant acceleration of $|g| = 9.81$ m/s^2. For example, suppose that a rock was thrown off a cliff from rest and took 5 seconds to hit the ground. We can write:

$$\Delta y = v_0(5) + \frac{1}{2}a(5^2) \tag{390}$$

We know that the rock was thrown from rest, so $v_0 = 0$, allowing us to simplify to:

$$\Delta y = \frac{1}{2}a(25) \tag{391}$$

We know that the acceleration is $g = 9.81$ m/s^2, and as the acceleration is downwards, we write it as negative:

$$\Delta y = -\frac{1}{2}(9.81)(25) \tag{392}$$

Which gives $\Delta y = -122.625$ m, meaning that the rock dropped in the downwards direction 122 meters, so the cliff's height is also 122 meters.

2D projectile motion To analyze projectile motion in two dimensions, we analyze the components of vertical and horizontal motion separately. To do this, we write the velocity as a 2D vector:

$$\vec{v} = \begin{bmatrix} v_x \\ v_y \end{bmatrix} \tag{393}$$

Using basic trigonometry, we can find that:

$$v_x = \vec{v}\cos\theta \tag{394}$$

$$v_y = \vec{v}\sin\theta \tag{395}$$

Where θ is the angle of the vector from the horizontal.

For a 2D projectile, one unique fact is that its horizontal velocity will always be the same as its initial horizontal velcity. That is:

$$v_x = v_{0_x} \tag{396}$$

Meanwhile, its vertical velocity follows the kinematic equation of freefall, with constant acceleration g. This means that:

$$v_y = v_{0_y}t + \frac{1}{2}at^2 \tag{397}$$

And as g is acting in the negative direction, we can write it as:

$$v_y = v_{0_y}t - \frac{1}{2}gt^2 \tag{398}$$

Lastly, remember that we can rewrite v_x and v_y in terms of $\vec{v}$ using trigonometry. Thus, the general equation of projectile motion of an object launched at angle θ_0 and initial velocity $\vec{v}_0$ is:

$$x(t) = \vec{v}_0\cos\theta_0 t \tag{399}$$

$$y(t) = \vec{v}_0\sin\theta_0 t - \frac{1}{2}gt^2 \tag{400}$$

A plot of the equation shows that it is correct:

```
import matplotlib.pyplot as plt
import numpy as np
from matplotlib.colors import LinearSegmentedColormap

t = np.linspace(0, 10)
v0 = 50
theta0 = np.pi / 4 # 45 degrees
g = 9.81
x = v0 * np.cos(theta0) * t
y = v0 * np.sin(theta0) * t - (1/2) * 9.81 * (t ** 2)

plt.plot(x, y)
plt.title("Projectile 2D motion")
plt.xlim(0, 250)
plt.ylim(0, 100)
plt.show()
```

Newton's Laws In the late-1600s, Isaac Newton came up with a series of 3 laws to describe a more general set of moving objects, not limited to objects in freefall. These form the foundation of Newtonian physics.

Newton's 1st law states that objects in constant motion remain in motion unless acted on by a net force. A net force occurs when the forces acting on an object are not balanced. This means that objects in constant motion can have forces acting on them, but no *net* force, because all the forces are balanced.

Newton's 2nd law states that a net force acts on an object to accelerate it, or mathematically:

$$F_{net} = \sum F = ma \tag{401}$$

Newton's 2nd law is often called the **equation of motion** for an object. To see why, notice that the law can be rewritten as:

$$a = \frac{F_{net}}{m} \tag{402}$$

And since acceleration is the second derivative of position, this can be written:

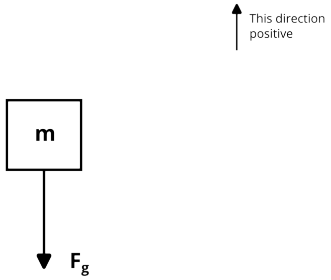
$$\frac{d^2x}{dt^2} = \frac{F_{net}}{m} \tag{403}$$

Thus, solving for Newton's second law gives the function $x(t)$ that describes the motion of an object through space and time.

Newton's 3rd law states that for every action force, there is an equal and opposite reaction force opposing it.

Of all of Newton's laws, it is the second that is the most typically helpful.

To utilize Newton's 2nd law, we often like to draw free-body diagrams to illustrate all the forces on an object. Forces that are in the positive direction (usually up and right) are positive, and forces in the negative direction (usually down and left) are negative. For instance, consider a free-falling object, influenced only by the downward force of gravity:



To apply Newton's 2nd law, we first sum all the forces to find the net force:

$$F_{net} = \sum F = 0 + (-F_g) = -F_g \tag{404}$$

Then, we equate the net force to ma:

$$F_{net} = ma \tag{405}$$

$$-F_g = ma \tag{406}$$

Given that the force of gravity close to Earth is given by $F_g = mg$, we can rewrite the last equation as:

$$-mg = ma \tag{407}$$

Which we can simplify to:

$$-g = a \tag{408}$$

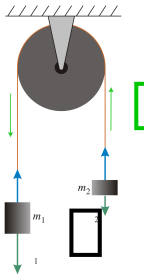
To find the motion of the object, we need to recall that the acceleration is the second derivative of position, so:

$$-g = \frac{d^2s}{dt^2} \tag{409}$$

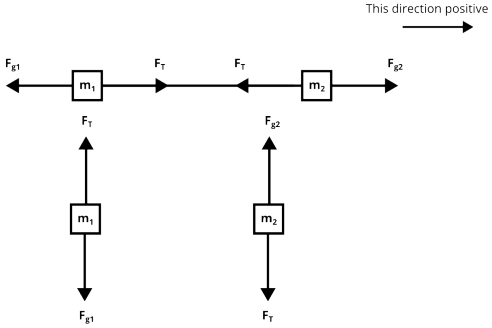
We can solve this differential equation by integrating both sides twice, to get:

$$s(t) = v_0 t - \frac{1}{2}gt^2 + h_0 \tag{410}$$

A more difficult example is with the following system, where F_T is the tension force, transmitted by the rope:



Here, to draw a free-body diagram, we "unwrap" and stretch out the system, then draw 2 separate free-body diagrams for each sub-system:



From there we can apply the 2nd law individually to each system. For the first system:

$$\sum F = F_T + (-F_{g_1}) = m_1 a \tag{411}$$

Which we simplify to:

$$F_T - F_{g_1} = m_1 a \tag{412}$$

And recalling $F_g = mg$, we can further simplify with:

$$F_T - m_1 g = m_1 a \tag{413}$$

And for the second system:

$$\sum F = F_{g_2} + (-F_T) = m_2 a \tag{414}$$

Which we can simplify to:

$$F_{g_2} - F_T = m_2 a \tag{415}$$

Again, we can use $F_g = mg$ to further simplify with:

$$m_2 g - F_T = m_2 a \tag{416}$$

We now have a simultaneous series of equations:

$$F_T - m_1 g = m_1 a \tag{417}$$
$$m_2 g - F_T = m_2 a \tag{418}$$

Solving them allows us to find the net acceleration of both systems:

$$m_2 g - F_T = m_2 a \tag{419}$$
$$m_2 g = F_T + m_2 a \tag{420}$$
$$F_T = m_2 g - m_2 a \tag{421}$$
$$F_T = m_1 a + m_1 g \tag{422}$$
$$m_2 g - m_2 a = m_1 a + m_1 g \tag{423}$$
$$m_2 g - m_1 g = m_1 a + m_2 a \tag{424}$$
$$g(m_2 - m_1) = a(m_1 + m_2) \tag{425}$$
$$a = \frac{g(m_2 - m_1)}{m_1 + m_2} \tag{426}$$

We can also use the same equations to derive a value of the tension force:

$$F_T = \frac{2 m_1 m_2 g}{m_1 + m_2} \tag{427}$$

And the equation of motion for the entire system can be found by solving the differential equation, which is in turn found by substituting a for the second derivative of position:

$$\frac{d^2 s}{dt^2} = \frac{g(m_2 - m_1)}{m_1 + m_2} \tag{428}$$

Using Newton's laws usually results in a second-order differential equation for a system - one typically not easily solved, and usually only solvable via computer.

Finally, it is worth mentioning **Newton's law of universal gravitation**, which states the gravitational force between two *arbitrary* distant masses (as opposed to close to Earth for $F_g = mg$) is given by:

$$F_g = -\frac{GMm}{r^2} \tag{429}$$

Where G is the universal gravitational constant, and $G \approx 6.67 \times 10^{-11}$

Work, Energy, and Power Analyzing forces is one very good approach to finding the equations of motion for a system, but it has its drawbacks - the vector equation $F_{net} = ma$ is often long and tedious to solve for. Another approach - the work-energy approach - is often much easier and more helpful for arriving at a solution when analyzing a given system.

At its simplest, **work** is simply force exerted over a distance:

$$W = \int_{x_1}^{x_2} F dx \tag{430}$$

Kinetic energy K, the energy of moving objects, is equal to the work done to accelerate an object from rest to a velocity v. That is:

$$K = \int_{x_1}^{x_2} F dx = \int_{x_1}^{x_2} ma dx = \int_{x_1}^{x_2} m\frac{dv}{dt} dx \tag{431}$$

Using the chain rule to expand out $\frac{dv}{dt} = \frac{dv}{dx}\frac{dx}{dt}$, we have:

$$\int_{x_1}^{x_2} m\frac{dv}{dx}\frac{dx}{dt} dx \tag{432}$$

If we move around the terms and rewrite $\frac{dx}{dt} = v$, we get:

$$\int_{x_1}^{x_2} mv\frac{dv}{dx} dx \tag{433}$$

The two dx's cancel and as part of our substitution we replace our endpoints in x with endpoints in v to get:

$$\int_0^v mv dv = \frac{1}{2}mv^2 \tag{434}$$

So:

$$K = \frac{1}{2}mv^2 \tag{435}$$

For instance, a 1 kg object moving at 10 m/s will have a kinetic energy of 50 Joules.

Potential energy U, the energy possessed by objects due to their position or configuration, is given by:

$$U(x) = -W = -\int F dx \tag{436}$$

> **Attention**
> Note that the second equation (the integral of force with respect to position) only holds true for **conservative forces** in the single-variable case, where the work done is independent of the path taken.

For gravity, this results in:

$$U_g(r) = -\int -\frac{Gm_1m_2}{r^2} dr = -\frac{GMm}{r} \tag{437}$$

Note that an approximation for gravitational potential energy is $U_g = mgh$, which works close to Earth.

For springs, where the elastic force is given by $F = -kx$, this results in:

$$U_s(x) = -\int -kx dx = \frac{1}{2}kx^2 \tag{438}$$

For electrostatics, where the electrostatic force is given by $F = \frac{kq_1q_2}{r^2}$, this results in:

$$U_e(r) = -\int \frac{kq_1q_2}{r^2}dr = \frac{kq_1q_2}{r} \tag{439}$$

The potential energy, when expressed as a quantity, is always taken *relative some reference point.* For a falling object, a common reference point is the surface of the Earth. This means that on the Earth's surface, an object has zero potential energy, while far from the Earth's surface, the object has maximal potential energy.

The **conservation of energy** states that the sum of potential and kinetic energy is constant - that is, the sum of inital potential and kinetic energy is equal to the sum of final kinetic potential and kinetic energy:

$$K_i + U_i = K_f + U_f \tag{440}$$

For example, let's use the same cliff example as earlier, with the only difference being that we know that the final velocity of the falling object is 48.9 m/s.

Using the conservation of energy, we know that $K_i = 0$ (as the object is thrown from rest), and $U_f = 0$ (as the object is on the surface of the Earth, our reference point). So:

$$U_i = K_f \tag{441}$$

$$mgh = \frac{1}{2}m(v_f)^2 \tag{442}$$

$$h = \frac{1}{2}\frac{m(v_f)^2}{g} \tag{443}$$

Plugging in the values, this gives us the same answer of the height of the cliff: 122 meters! However, note that we didn't need to use any kinematic formulas, just an understanding of work and energy!

Momentum **Momentum** is the product of mass and velocity:

$$p = mv \tag{444}$$

Which also makes the derivative of the momentum force:

$$F = \frac{dp}{dt} \tag{445}$$

And the (time) integral of force momentum:

$$p = \int F dt \tag{446}$$

Do not confuse this with potential energy, which is the position integral of force!

Momentum cannot be created nor destroyed; it can only be transferred between objects. This is the principle of the **conservation of momentum**:

$$P_{A_i} + P_{B_i} = P_{A_i} + P_{B_f} \tag{447}$$

Or more generally:

$$\sum P_i = \sum P_f \tag{448}$$

Impulse is the change in momentum:

$$J = \Delta P = \int_{t_1}^{t_2} F dt \tag{449}$$

Fields A vector field is an object that spans all of space and assigns a value to each point in space. For example, take the gravitational field, which gives each point a vector that is given by:

$$\vec{g}(r) = \frac{GM}{r^2} \tag{450}$$

Here is a visualization of the gravitational field $\vec{g}(r)$:

```
def plot_gfield():
    """
    Plot vector field of gravity in polar coordinates
    """
    G = 1
    M = 1
    radii = np.linspace(1, 3, 5)
    thetas = np.linspace(0, 2 * np.pi, 20)
    theta, r = np.meshgrid(thetas, radii)
    R = -(G * M) / (r ** 2)

    f = plt.figure()
    ax = f.add_subplot(polar=True)
    ax.quiver(theta, r, R * np.cos(theta), R * np.sin(theta))
    plt.title("Gravitational vector field")
    plt.show()

plot_gfield()
```

The gravitational field is related to the gravitational force by:

$$\vec{g} = \frac{F}{m} \tag{451}$$

That means that at each point in the field, an object of mass m will feel a force of $\vec{g}m$.

A potential field, similar to a vector field, is an object that spans all of space and assigns a scalar to each point in space. Potentials have the special property that their slopes are equal to another vector field. For example, the gravitational (vector) field $\vec{g}$ is related to the gravitational potential ϕ by:

$$\vec{g} = -\nabla\phi = -\left(\frac{\partial\phi}{\partial x}, \frac{\partial\phi}{\partial y}, \frac{\partial\phi}{\partial z}\right) \tag{452}$$

From this, we can derive that the gravitational potential is given by:

$$\phi = \frac{GM}{r} \tag{453}$$

> **Note**
> This applies for all $r > R$ for a spherical gravitating body (e.g. planet) of radius R, that is, *outside* the spherical body. Inside the spherical body, the gravitational potential is given by $\phi = -\frac{GM}{2R^3}(3R^2 - r^2)$.

A visualization of the gravitational potential is shown below:

```
def gravitational_potential(r, radius=1.5, G=1, M=0.5):
 # slightly more complex than
 # the formula to accomodate
 # r < R where R is radius of body
```

```
 return np.where(r > radius, -G*M/r, -G*M*(3*radius**2 - r**2)/(2*radius**3))

def calc_div_grav():
    %matplotlib inline
    radii = np.linspace(0, 7, 50)
    thetas = np.linspace(0, 2 * np.pi, 50)
    R, T = np.meshgrid(radii, thetas)
    Phi = gravitational_potential(R)

    # convert r, theta coordinates to
    # cartesian x, y coordinates
    X, Y = R*np.cos(T), R*np.sin(T)

    f = plt.figure(figsize=(7, 7))
    ax = f.add_subplot(projection="3d")
    ax.set_zlim( -0.75, 0.15)
    # Set the angle of the camera
    ax.view_init(25, -45)

    # Colormap
    cmap = LinearSegmentedColormap.from_list("", ["#D54C90", "#A37DF8", "#B3E6FF"])
    ax.plot_surface(X, Y, Phi,linewidth=0.1 ,
                    cmap=cmap,
                    alpha=1,
                    cstride=2,
                    rstride=2,
                    edgecolors="black")
    plt.title(r"Gravitational potential field $\Phi$")
    plt.grid()
    plt.rcParams["figure.autolayout"]
    plt.show()

calc_div_grav()
```

The gravitational potential is related to the gravitational potential energy by:

$$\phi = \frac{U}{m} \tag{454}$$

That means that at each point in the potential field, an object of mass m will have a gravitational potential energy of $U = \phi m$ relative to infinity.

The divergence of the gravitational field is given by:

$$\nabla \cdot \vec{g} = -4\pi G\rho \tag{455}$$

Combining the two equations, we can find the gravitational potential in terms of its mass density:

$$\vec{g} = -\nabla\Phi \tag{456}$$

$$\nabla \cdot -\nabla\Phi = \nabla^2\Phi = 4\pi G\rho \tag{457}$$

This is **Poisson's equation**:

$$\nabla^2\Phi = 4\pi G\rho \tag{458}$$

In empty space, the equation reduces to:

$$\nabla^2 \Phi = 0 \tag{459}$$

Electricity and magnetism

> "Science is an ongoing process. It never ends. There is no single ultimate truth to be achieved...because this is so, the world is far more interesting."
> **Carl Sagan**

Physics as we know today may have started with Newton, but it most certainly did not end with him. Newton's contemporaries (and sometimes rivals) continued to build on his work. In the following centuries, advancements in physics resulted in the successful theories explaining everything from sound, to mechanical vibrations, to heat flow, to even the tides. But among the most long-lasting - and most successful - achievements of classical physics is the classical theory of **electricity and magnetism**.

Electrostatic force The classical theory of electricity and magnetism concerns the behavior of **charged objects**. Objects are **charged** due to an imbalance in their number of protons and electrons. As protons are immobile, the *movement of electrons* causes changes in charge. Objects gaining electrons become negatively-charged, and objects losing electrons become positively charged. We measure charge using units of **coulombs** and typically use the symbol q for charge.

Like charges attract, and opposing charges repel, causing a force between any two charges placed together. The magnitude of this attraction (or repulsion) can be found from **Coulomb's force law**, expressed as follows:

$$F_e = k\frac{q_1 q_2}{r^2} \tag{460}$$

Here, r is the separation between the charges, and k is the Coulomb constant, equivalent to about $8.99 \times 10^9\ \mathrm{N \cdot m^2/C^2}$. It is common to call $\vec{F}_e$ the *electrostatic force* - here, we'll use the term *electric force*, which is more intuitive (though less accurate). *Electrostatic* means that charges are moving so slowly that we may consider them essentially stationary (static). In many cases, we can *assume* that this approximation holds true, even if charges are technically moving, as long as the charges are relatively slow-moving.

> **Note**
> It is common to write $k = \dfrac{1}{4\pi\varepsilon_0}$ in equations involving electrostatics, where ε_0 is the **electric constant**, but here we have chosen to just use k for simplicity.

The **vector form** of Coulomb's force law is found by taking the scalar form and adding a unit vector $\hat{r}_{12}$ pointing between the two objects. This, however, is not as simple as it may seem, because a force must be the action of *one* object on *another* object. Thus the force of charge q_1 acting on q_2, which is a vector, is *not* the same vector as the force of charge q_2 acting on q_1 (in fact they are opposite in direction, by Newton's third law). Therefore, we must define two different forces, $\vec{F}_{12}$ for the force *exerted by* charge q_1 on charge q_2, and $\vec{F}_{21}$ for the force *exerted by* charge q_2 on charge q_1. They are written as follows:

$$\vec{F}_{12} = k\frac{q_1 q_2}{r^2}\hat{r}_{12} \tag{461}$$

$$\vec{F}_{21} = k\frac{q_1 q_2}{r^2}\hat{r}_{21} \tag{462}$$

$$\tag{463}$$

Here, $\hat{r}_{12}$ is the unit vector pointing from q_1 to q_2, and similarly, $\hat{r}_{21}$ is the unit vector pointing from q_2 to q_1:

$$\hat{r}_{12} = \frac{\vec{r}_2 - \vec{r}_1}{\|\vec{r}_2 - \vec{r}_1\|} \tag{464}$$

$$\hat{r}_{21} = \frac{\vec{r}_1 - \vec{r}_2}{\|\vec{r}_1 - \vec{r}_2\|} \tag{465}$$

A particularly nice quality about Coulomb's force law - and the whole of electricity and magnetism in general - is that electric forces (and as we'll see later on, magnetic forces) obey the **superposition principle**. This means that the combined electric (or magnetic) force is just a sum of the individual forces pointing between each of the charges. This also means that the differential equations in the theory of electricity and magnetism are **all linear differential equations**, and two solutions can be added together to find a new solution without any extra work. This will be a *very important* and useful fact later on.

Electric field We recall that Coulomb's force law, like any force, is subject to Newton's second law, and thus results in a differential equation that can be solved to find the trajectories of each of the two charges. In the case of two charges q_1, q_2 interacting, the differential equations are:

$$\frac{d^2\vec{r}_2}{dt^2} = k\frac{q_1 q_2}{|\vec{r}_2 - \vec{r}_1|^2}\hat{r}_{12} \tag{466}$$

$$\frac{d^2\vec{r}_1}{dt^2} = k\frac{q_1 q_2}{|\vec{r}_1 - \vec{r}_2|^2}\hat{r}_{21} \tag{467}$$

However, if there are more than two charges interacting, the forces between all the charges must be accounted for, meaning that the differential equations grow extremely long and become solvable only by computer. For this reason, while Coulomb's force law is sometimes useful, a **field formulation** is the far more preferred method of mathematically modelling the interactions of charges.

Recall that fields (in physics) denote quantities that are continuously spread out across all of space, such as the gravitational field and gravitational potential (potential field). The **electric field** is a vector field produced by a charge and extends throughout space. In the field formulation, instead of a charge directly exerting a force on other charges, it is the *electric field* of the charge that exerts the force. Every other charge *also* produces an electric field that exerts a force on all the other charges except themselves. Each of these electric fields, that we've considered separate up to this point, are really just labels for parts of *one* electric field that carries forces between *all* charges. Every charge produces part of the electric field, and the electric field exerts a force on every charge, determining the trajectories of each charge. In words inspired by the physicist John Archibald Wheeler, we may surmize that:

> *"Charges make the electric field change, the electric field makes charges move."*

While technically there is no distinction between the electric fields of different charges - they are all part of the same *singular electric field* - it is mathematically convenient to speak of electric fields specific to a single charged object. For instance the electric field *produced* by a single point charge Q located at the origin is given by:

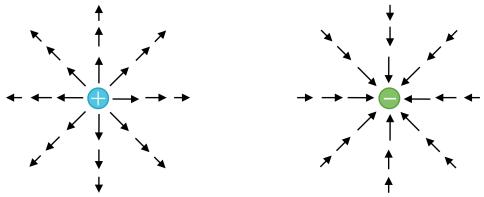
$$\vec{E}(\vec{r}) = \frac{kQ}{r^2}\hat{r} \tag{468}$$

This electric field exerts an electric force $\vec{F}_e$ on any other charge q, given by:

$$\vec{F}_e = q\vec{E} \tag{469}$$

Note
Be sure to remember that the Q here is of the charge *creating* the electric field, and q is of the charge that *feels* the force from the electric field.

Since the electric field is a vector field - more precisely, a force field - we can visualize it with vector plots. Using the example of our point charge, the electric field vectors extend outwards if the point charge is positive, and inwards if the point charge is negative:



Electric fields of a positive point charge (shown left) and a negative point charge (shown right).

When the electric field is created by *two* charges of opposite sign, we call it a dipole. There exist electric fields produced by more than two charges, right up to arrangements of uncountably many charges. In any case with more than one charge, we must superimpose (sum) the individual electric fields from each charge, resulting in an electric field formed by a *superposition* of charges:

$$\vec{E}(\vec{r}) = \sum_i \frac{kQ_i}{|\vec{r} - \vec{r}_i|^2}\hat{r}_i \tag{470}$$

Here, $\vec{r}_i$ is the position of a charge Q_i in the collection of charges, and $\hat{r}_i$ is the unit vector pointing from $\vec{r}_i$ to $\vec{r}$ (this direction can be somewhat confusing: draw these vectors out on paper to see why it's the case).

Note
Unless otherwise specified, we always use $\vec{r} = \langle x, y, z\rangle$ as the **position vector** pointing from the origin to point (x, y, z) in space.

This formula may be slightly clearer if we rewrite it explicitly in terms of coordinate (rather than vector) form:

$$\vec{E}(x, y, z) = \sum_i \frac{kQ_i}{[(x - x_i)^2 + (y - y_i)^2 + (z - z_i)]^2}\hat{r}_i \tag{471}$$

If we want to continue this process by considering a collection of ever-smaller charges, we can find the electric field of a *continuous distribution* of charges by integration. To do so, we shrink the charges Q_i to infinitesimal charges dq, then integrate along every point $\vec{r}' = (x', y', z')$ within the charge-containing region. The charge distribution can be assumed to be continuous as the charges shrink to very very small, so we may define a charge density function $\rho(\vec{r}')$, where $dq' = \rho(\vec{r}')\, dV'$ is the infinitesimal amount of charge contained in a tiny region of space dV' within the charge-containing region. The electric field produced by the entire distribution of charges is then given by:

$$\vec{E}(\vec{r}) = \int \frac{k\, dq'}{|\vec{r} - \vec{r}'|^2}\hat{r}' \tag{472}$$

$$= \int \frac{k\, dq'}{|\vec{r} - \vec{r}'|^2}\frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|} \tag{473}$$

$$= k\int \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|^3}\rho(\vec{r}')\, dV' \tag{474}$$

$$\tag{475}$$

Where $\hat{r}' = \dfrac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|}$ is the unit vector pointing from a point $\vec{r}' = (x', y', z')$ within the charge-containing region to point $\vec{r} = (x, y, z)$. All the primes in the integral (e.g. $\vec{r}', \hat{r}', dV'$) indicate points *within* the charge-containing region, as we integrate over every point within that region, whereas all the unprimed coordinates (e.g. $\vec{r}$) indicate a point in space (which can be outside the charge-containing region). We may *also* write this explicitly in Cartesian coordinates as follows:

$$\vec{E}(x, y, z) = E_x\hat{i} + E_y\hat{j} + E_z\hat{k} \tag{476}$$

$$E_x(x, y, z) = k \int \frac{\rho(x', y', z')(x - x')]}{[(x - x')^2 + (y - y')^2 + (z - z')^2]^{3/2}} dx'dy'dz' \tag{477}$$

$$E_y(x, y, z) = k \int \frac{\rho(x', y', z')(y - y')]}{[(x - x')^2 + (y - y')^2 + (z - z')^2]^{3/2}} dx'dy'dz' \tag{478}$$

$$E_z(x, y, z) = k \int \frac{\rho(x', y', z')(z - z')]}{[(x - x')^2 + (y - y')^2 + (z - z')^2]^{3/2}} dx'dy'dz' \tag{479}$$

This is **the integral form of Coulomb's law** for the electric field, and does give the right expression for the electric field, at least when the electrostatic approximation holds. It is (understandably) rather tedious to solve and often can only be solved by computer.

Note that the integral form of Coulomb's law has three specific cases depending on whether the charge density is a *linear* charge density $\lambda(\vec{r}')$, *surface* charge density $\sigma(\vec{r}')$, or *volume* charge density $\rho(\vec{r}')$.

For linear charge distributions, we have $dq' = \lambda(\vec{r}')dr'$ where dr' is the line element of the linear charge distribution to integrate over (e.g. charged rod, loop of wire, etc.) This means the integral becomes a line integral betweem the endpoints of the linear charge distribution (e.g. between ends of wire, 360 degrees around a circular loop, along the path of a charge helix, etc.):

$$\vec{E}(\vec{r}) = \int_{\text{line}} k\frac{\lambda(\vec{r}')dr'}{|\vec{r} - \vec{r}'|^3}(\vec{r} - \vec{r}') \tag{480}$$

Again, we may choose to work in component form by writing $\vec{E} = E_x\hat{i} + E_y\hat{j} + E_z\hat{k}$, for which we have:

$$E_x(x, y, z) = \int_{\text{line}} k\frac{\lambda(x', y', z')(x - x')}{[(x - x')^2 + (y - y')^2 + (z - z')^2]^{3/2}} dr' \tag{481}$$

$$E_y(x, y, z) = \int_{\text{line}} k\frac{\lambda(x', y', z')(y - y')}{[(x - x')^2 + (y - y')^2 + (z - z')^2]^{3/2}} dr' \tag{482}$$

$$E_x(x, y, z) = \int_{\text{line}} k\frac{\lambda(x', y', z')(z - z')}{[(x - x')^2 + (y - y')^2 + (z - z')^2]^{3/2}} dr' \tag{483}$$

Note that $\lambda(x', y', z')$ describes the charge density *along a line of charge*, **not** at every point in space. It may be more *helpful* to think of x', y', z' as the *parametric curve* given by:

$$\begin{cases} x' = f(s) \\ y' = g(s) \\ z' = h(s) \end{cases} \tag{484}$$

where s is a parameter. Parametrization is *not* needed for actually doing calculations in the integral; this is just a way of building intuition.

Given its level of complexity, the component form of Coulomb's law for the electric field is only good for situations where the (possibly curved) line of charge is align along one axis, or when doing computer-based calculations. It, however, illustrates several easy-to-miss aspects about Coulomb's law.

First, all three components of the electric field are integrated over the *same* region. This means that, for instance, if we consider a line of charge purely along the x-axis, then E_x, E_y, E_z are **all integrated** over $\lambda(x')dx'$, even though they are components of the electric field along different directions. Second, the magnitude of the displacement vector is *the same* regardless of whether we compute E_x, E_y, or E_z. This is why each of the integrals has the same $[(x-x')^2+(y-y')^2+(z-z')^2]^{3/2}$ term in the denomination, even though they are different components of the electric field. While we will now examine charge distributions that are not along a line (or curve), these two properties will still hold.

For surface charge distributions (e.g. plane of charge, disk of charge, etc.) we have $dq' = \sigma(\vec{r}')dA' = \sigma dx'dy'$ where dA' is the surface element of the surface charge distribution to integrate over. Therefore, Coulomb's law becomes a surface integral over every patch of charge dq' across every patch of surface dA':

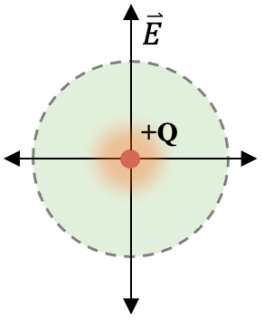
$$\vec{E}(\vec{r}) = \iint\limits_{\text{surface}} k\frac{\sigma(\vec{r}')dA'}{|\vec{r}-\vec{r}'|^3}(\vec{r}-\vec{r}') = \iint\limits_{\text{surface}} k\frac{\sigma(\vec{r}')dx'dy'}{|\vec{r}-\vec{r}'|^3}(\vec{r}-\vec{r}') \tag{485}$$

For volume charge distributions (e.g. solid sphere of charge, shell of charge, block of charge), we have $dq' = \rho dV' = \rho(\vec{r}')dx'dy'dz'$ where dV' is the volume element of the volume charge distribution to integrate over. Therefore, Coulomb's law becomes volume integral over every infinitesimal volume dV' containing charge dq':

$$\vec{E}(\vec{r}) = k\iiint\limits_{\text{volume}} \frac{\rho(\vec{r}')dV'}{|\vec{r}-\vec{r}'|^3}(\vec{r}-\vec{r}') = k\iiint\limits_{\text{volume}} \frac{\rho(\vec{r}')\,dx'dy'dz'}{|\vec{r}-\vec{r}'|^3}(\vec{r}-\vec{r}') \tag{486}$$

Note
It is important to remember that in all cases of applying Coulomb's law, the integrals are *always* done **with respect to the primed coordinates**. That is, we integrate over x', y', z', *not* x, y, z. Each point (x', y', z') represents a point *within the charge distribution*, and we integrate over the contribution to the electric field from all the points within the charge distribution to be able to find the total electric field. In fact, you should consider any x, y, or z appearing in the integral as **constants**, as we integrate over x', y', z' instead of x, y, z.

It is common to say that applying Coulomb's law is finding the electric field by *brute-force*, understandably, given its tediousness. But there is a more *elegant* way of computing the electric field. Suppose we analyze an bounded region of space around an electric field. For instance, this could be the spherical region around a point charge, as shown in the figure below:



A spherical region around point charge of radius r.

We could model this spherical region as a sphere (unsurprisingly). Then the region would be defined by a sphere of radius r, which contains a total volume of:

$$V = \frac{4}{3}\pi r^3 \tag{487}$$

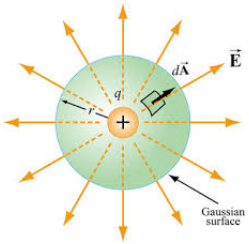
Which is just the volume of a sphere. Now, we can describe the *bounds* of this region as the edge of the sphere. The edge of the sphere is simply its surface, and the formula of the surface area of the sphere is given by:

$$A = 4\pi r^2 \tag{488}$$

If we divide the surface of the sphere into tiny "patches" of surface, each of size dA, then we can *integrate* the electric field over the entire surface of the sphere to get the total electric field "passing" the boundary of the sphere. We call this the **electric flux**, symbol Φ_E, and we can write it as an integral (the circle means "closed boundary", i.e. no holes in the boundary), like this:

$$\Phi_E = \oint_{\text{surface}} \mathbf{E} \cdot d\mathbf{A} \tag{489}$$

We can illustrate the flux across the spherical boundary (the technical term is *Gaussian surface*) of our spherical region as follows:



A Gaussian surface - an imaginary bounding surface surrounding a spherical region.

> **Note**
> The reason this is a dot product rather than normal multiplication is that the amount of electric field "passing" the boundary of our spherical region also depends the angle between the electric field and the surface. In this case, the electric field vectors are perfectly perpendicular to the surface, so it reduces to a normal dot product, but this is not *always* the case.

Through a mathematical law called **Gauss's law**, the total amount of flux is equal to the total charge density *within* the region, multiplied by a constant:

$$\oint \mathbf{E} \cdot d\mathbf{A} = 4\pi k Q_{\text{total}} \tag{490}$$

This leads to a crucial result: the electric field passing through the *bounds* of our spherical region is *determined* by the total amount of charge *within* the spherical region's volume. Thus, if we know the total charge within the spherical region, then we can figure out the electric field within it! In addition, Gauss's law also can be converted (through some vector calculus identities) into a partial differential equation. We call it the *differential form* of Gauss's law, and it takes the following form:

$$\nabla \cdot \mathbf{E} = 4\pi k \rho \tag{491}$$

While not as easy to work with, solutions to the differential form of Gauss's law are perfectly equivalent to the integral version, and also give you the electric field.

Electrical potential energy and electric potential The **electrical potential energy** is the energy stored in a collection of charges, and we denote it by U_E. For two charges q_1, q_2, the electric potential energy stored within the system of charges is given by:

$$U_E = \frac{k q_1 q_2}{r} \tag{492}$$

The electric potential energy arises as a consequence of the electric force. When the electric force is attractive, charges are bound together, and it *takes energy* to "knock" a charge out of position. An example of this is within an atom, where the positively-charged protons and the negatively-charge electrons are held together by an attractive force, preventing them from flying away from each other. In such cases, the electric potential energy of the system is **negative**, meaning that the system is stable (physics-wise) and an amount of energy *equal* in magnitude to the electric potential energy must be *put in* to break apart the system. The converse is true if the electric force is repulsive. In this case, *no energy is required* to "knock" the like charges out of position; in fact, it would *take* energy to keep the like charges in place. So the electric potential energy of the system is **positive**, meaning that energy must be *released* - typically by converting the potential energy of the system to some other form of energy - to keep the system together.

But just as we found that electric fields are mathematically more useful than simply analyzing electric forces, there exists a more useful and elegant way to formulate electric potential energy. Instead of considering the electric potential energy of a collection of charges, we examine just a *single charge* Q, and ask what the electric potential energy *would theoretically be* if we chose to place another charge q at some position $\vec{r}$ to form a two-charge system. We may then write the electric potential energy as $U_E = qV$, where $V(r)$ is the **electric potential**:

$$V(r) = \frac{kQ}{r} \tag{493}$$

The electric potential describes what the electric potential energy *would be* if we place an arbitrary number of charges at any location in space. Unlike the electric potential energy, it is a function of position, which means that it is not a number; it is a **field**. More specifically, it is a scalar field that is closely related to the electric field. But first, it may be helpful to have a more intuitive picture.

The electric potential can be thought of like electrical pressure of a sort. Just as normal water pressure causes fluid motion of water molecules that are within the water, voltage causes the motion of the charges placed in an electric field. One may say that the electric potential "pushes" (positive) charges from regions of *higher energy to lower energy* - outwards if the electric potential is positive, and inwards if the electric potential is negative (the opposite is true for negative charges). The strength of this "push" depends on the difference in electric potential between two points; if two points have nearly the *same* electric potential, the "push" is very weak, but if two points have a *great difference* in electric potential, the "push" can be very strong. But a distribution of charges creates an electric field! Thus the electric potential's variation in space is also the *source* of the electric field, which we may write mathematically as:

$$E = -\nabla V \tag{494}$$

Voltage, also called **potential difference**, is when you evaluate the *difference* in potential between two points, $V(b) - V(a)$, just as we alluded to earlier. These two points could be one point on a wire in the air and the other point on the ground. Another two points could be a charge in empty space and a point infinitely far away. The first point pair is most useful for calculations that require accuracy, while the second is most useful for calculations where we can approximately assume that the potential would become weaker and weaker at longer distances from a charge.

With all that being said, how do we actually go about *calculating* the electric potential? There exist two primary ways of computing the electric potential. The first, the sum/integral method, is very similar to that of the electric field. In the discrete case (i.e. individual charges at different locations) it reads:

$$V(r) = \sum_i \frac{kQ_i}{|\vec{r} - \vec{r}_i|} \tag{495}$$

While in the continuous case, it reads:

$$V(r) = k \int \frac{\rho \, dV}{|\vec{r} - \vec{r}'|} \tag{496}$$

Again, $\vec{r} - \vec{r}_i$ represents the vector pointing from the location $\vec{r}_i$ of a given charge to $\vec{r}$, and $\vec{r} - \vec{r}'$ represents the vector pointing from a given point $\vec{r}'$ in the charge distribution to (the position vector) $\vec{r}$.

The second method of computing the electric potential, however, is the *far* more elegant way of computing the electric potential, and it is **Poisson's equation**. It reads:

$$\nabla^2 V = -4\pi k\rho \tag{497}$$

Poisson's equation is a partial differential equation (PDE) that can be solved for the electric potential using any number of techniques for solving differential equations. Once solved, the electric field is easily found through $\vec{E} = -\nabla V$, and from there, the equations of motion (differential equations describing the trajectories) of any charge in the electric field can be found.

The magnetic field Up to this point, we have considered *electrostatics*, where charges are slow-moving or don't move at all. But what happens if charges *do* move? We observe that a strange *new* force shows up, one that is distinct yet strangely similar to the electric force. We call this force the **magnetic force**.

A magnetic force arises whenever there is a **current**. A current is a *flow* of charge; more precisely, we define it as:

$$I = \frac{dQ}{dt} \tag{498}$$

Where I is the current, and Q is the amount of charge passing through a given cross-section of wire at a given time t. Just as we defined ρ as the charge density, we may also define a **current density**, denoted $\vec{J}$, where $\vec{J}$ is given by:

$$\vec{J} = \frac{\partial I}{\partial A}\hat{I} \tag{499}$$

Here, A is the cross-sectional area through which the current flows, and $\hat{I}$ is the direction of the motion of positive charges. Why a current density would ever prove useful will be revealed in a few sections.

The crucial other component that makes magnetic forces possible is the **magnetic field**. The magnetic field is like the electric field in some ways; it is a vector field, it acts on charged objects, and it spans across all space. But it is also different; it is **velocity-dependent** and vanishes as charges slow to a stop. In addition, there are *no magnetic charges* - in fact, magnetic charges, also called *monopoles*, are forbidden. The closest physically-possible analogue to a "magnetic charge" is a magnetic *dipole*, consisting of two opposite charges.

We write the magnetic field as $\vec{B}$, and the magnetic field strength is given in units of **Tesla**, shorthand T. The magnetic force can then be written in one of two ways:

$$\vec{F}_B = q\vec{v} \times \vec{B} = \int I d\ell \times \vec{B} \tag{500}$$

These two forms describe the magnetic force generated by 1) a moving point charge q and 2) a current-carrying segment of wire of current I. Note that both expressions use a **cross product**: this is because the magnetic force is always *perpendicular* to the direction of the moving charges.

> **Note**
> What about the magnetic fields and forces generated by permanent magnets like bar magnets or fridge magnets? The answer is that while they can be macroscopically-modelled with classical theory, the full explanation for why they remain magnetized even without moving charges requires special relativity and quantum mechanics.

How do we *compute* the magnetic field though? As with before, there are several different options. But first, let's cover an option that you perhaps would *think* could work, but doesn't actually work. Perhaps you would think that since there is a Gauss's law for the *electric field*, there would also be one for the *magnetic field*. Indeed, there is actually one, but it is rather disappointing:

$$\begin{cases} \oint \mathbf{B} \cdot d\mathbf{A} = 0, & \text{(integral form)} \\ \nabla \cdot \mathbf{B} = 0, & \text{(differential form)} \end{cases} \tag{501}$$

Which leaves much to be desired. The formal reason for *why* Gauss's law for magnetic fields is this way, however, is important *conceptually*. Remember how we said that there are **no magnetic charges**. But the right-hand side of Gauss's law for electic fields is the total charge enclosed within a region of space. Since a magnetic charge is *not defined*, the right-hand side of Gauss's law for magnetic fields has to be zero - after all, there are no charges!

In addition, the differential version also tells us that magnetic fields always come in *loops* - the vectors follow looping field lines that ultimately trace back to the point they started from - which means that the magnetic field has *no net divergence*.

So instead of Gauss's law for magnetism, we instead use an analogue of Coulomb's law for the electric field. This is the **Biot-Savart law**. In the discrete case, for charges located at distinct points in space, it takes the form:

$$\mathbf{B}(r) = \frac{\mu_0}{4\pi} \sum_i Q_i \frac{\vec{v}_i \times \vec{r}_i}{|r - r_i|^2} \tag{502}$$

In the continuous case, where we consider many, many moving charges that form a continuous current, Biot-Savart's law becomes an integral:

$$\vec{B}(\vec{r}) = \frac{\mu_0}{4\pi} \int \frac{I\, d\vec{\ell} \times \hat{r}'}{|\vec{r} - \vec{r}'|^2} \tag{503}$$

$$= \frac{\mu_0}{4\pi} \int \frac{I\, d\vec{\ell}}{|\vec{r} - \vec{r}'|^2} \frac{\vec{r} - \vec{r}'}{|\vec{r} - \vec{r}'|} \tag{504}$$

$$= \frac{\mu_0}{4\pi} \int \frac{I d\vec{\ell} \times (\vec{r} - \vec{r}')}{|\vec{r} - \vec{r}'|^3} \tag{505}$$

Where $d\vec{\ell}$ is a current-carrying path segment (such as a segment of wire), $\vec{r}'$ is a point along the current-carrying path, $\vec{r}$ is the position vector, and μ_0 is the magnetic constant. This integral's particular form means that Biot-Savart's law is a (vector) **line integral** over the current-carrying path (which is usually, though not always, a wire), as we integrate over every portion of that path.

As with Coulomb's law for the electric field, Biot-Savart's law is very tedious to work with and can often be only solved by computer. In addition, just like Coulomb's law for the electric field only works when the electrostatic approximation can be made, Biot-Savart's law only works when the **magnetostatic approximation** can be made, meaning that charges move at near-constant velocities and currents change very slowly, if at all.

> **Note**
> The magnetostatic and electrostatic assumptions mean that the electric fields (in the former) and magnetic fields (in the latter) **do not change** - we say they are *static*. Thus, as long as the assumptions hold, neither field has a dependence on time.

The second method that can be used is very similar to Poisson's equation for electric fields. If we define a **magnetic potential A** such that $\mathbf{B} = \nabla \times \mathbf{A}$, the magnetic potential can be solved for by Poisson's equation for the magnetic field, which reads:

$$\nabla^2 \mathbf{A} = -\mu_0 \mathbf{J} \tag{506}$$

In Cartesian coordinates, this expands to:

$$\frac{\partial^2 \mathbf{A}}{\partial x^2} + \frac{\partial^2 \mathbf{A}}{\partial y^2} + \frac{\partial^2 \mathbf{A}}{\partial z^2} = -\mu_0 \, \mathbf{J} \tag{507}$$

This method is used more often in advanced physics such as relativity and quantum mechanics, but it generally follows the same approach as using Poisson's equation for the electric field - solve the (partial) differential equation using any number of methods (and computer if need be), then find the magnetic field by taking its curl, that is, $\mathbf{B} = \nabla \times \mathbf{A}$. Although elegant, Poisson's equation for the magnetic field only works in cases when the magnetostatic approximation holds. In many cases, we *cannot* make this assumption.

The third method for computing the magnetic field, however, works universally even when currents are not constant and charges are changing velocity. It is what we will cover next - **Maxwell's equations**.

Electrodynamics and Maxwell's equations Up to this point, we have discussed electostatics and magnetostatics, but we will now cover **electrodynamics** - the more general study of electricity and magnetism in cases where electric and magnetic fields change through time. The laws of electrodynamics are given by the system of partial differential equations known as **Maxwell's equations**. These equations govern the evolution of the electric and magnetic fields, and they read as follows:

$$\nabla \cdot \mathbf{E} = 4\pi k \rho \tag{508}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{509}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{510}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \tag{511}$$

These equations are also often appear in integral form, in which they are given as:

$$\oint_{\text{surface}} \mathbf{E} \cdot d\mathbf{A} = 4\pi k Q_{\text{total}} \tag{512}$$

$$\oint_{\text{surface}} \mathbf{B} \cdot d\mathbf{A} = 0 \tag{513}$$

$$\oint_{\text{loop}} \mathbf{E} \cdot \vec{d\ell} = -\frac{\partial}{\partial t} \int_{\text{surface}} \mathbf{B} \cdot d\mathbf{A} \tag{514}$$

$$\oint_{\text{loop}} \mathbf{B} \cdot \vec{d\ell} = \mu_0 I + \frac{1}{c^2} \frac{\partial}{\partial t} \int_{\text{surface}} \mathbf{E} \cdot d\mathbf{A} \tag{515}$$

The Maxwell equations show a surprising fact: oscillating electric fields can actually *induce* magnetic fields, and oscillating magnetic fields can actually *induce* electric fields. So rather than two separate phenomena, electricity and magnetism are actually interrelated phenomena, caused by the interplay of electric and magnetic fields. Thus, we often group electricity and magnetism together as **electromagnetism**, and speak of an *electromagnetic field* as the combination of the electric and magnetic components of the field.

The equation that govern the motion of charged objects in an electromagnetic field is the **Lorentz force law**, which reads:

$$m \frac{d^2 \mathbf{r}}{dt^2} = \mathbf{F}_{EM} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \tag{516}$$

Solving the Maxwell equations is a topic so extensive that it is its own field. We'll sketch out one of the most common methods; however, to avoid making this chapter overwhelmingly long, we will have to skip a lot of the details.

The general process of this method is to first set boundary conditions such that the first two equations (Gauss's laws for the electric and magnetic fields) hold true. Then, we are left with the two remaining equations:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{517}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t} \tag{518}$$

The top equation is called **Faraday's law**, and the bottom equation is called the **Maxwell-Ampère law**. Now, it turns out that by reworking the electric and magnetic potentials, we can substitute them in to find an *electromagnetic* potential, assuming that:

$$\mathbf{E} = -\nabla V - \frac{\partial \mathbf{A}}{\partial t} \tag{519}$$

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{520}$$

Then, if we impose something called the *Lorenz gauge condition* (it's something we will cover more in the Expert's Guide), we can substitute the potentials into Maxwell's equations to get:

$$\nabla^2 \mathbf{A} - \frac{1}{c^2}\frac{\partial^2 V}{\partial t^2} = -\mu_0 \mathbf{J} \tag{521}$$

$$\nabla^2 V - \frac{1}{c^2}\frac{\partial^2 V}{\partial t^2} = -4\pi k \rho \tag{522}$$

These equations are much easier (though by no means *easy*) to solve and are often used in advanced electromagnetic theory to solve for complicated field configurations. However, we won't go that far, at least for now. Rather, we'll explore *one* simplified case of Maxwell's equations, and its profound consequences on the physical nature of light.

> **Note**
> For more detailed information about Maxwell's equations, we recommend reading A student's guide to Maxwell's equations, which is slower-paced compared to the guide here in the Elara Handbook.

Electromagnetic waves When only simulating electromagnetic waves radiating within space, and not the source currents or charges, Maxwell's equations can be simplified by setting $\rho = \mathbf{J} = 0$, resulting in two wave equations:

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} = c^2 \nabla^2 \mathbf{E} \tag{523}$$

$$\frac{\partial^2 \mathbf{B}}{\partial t^2} = c^2 \nabla^2 \mathbf{B} \tag{524}$$

Where $c^2 = \frac{1}{\mu_0 \varepsilon_0}$ and c is the speed of light. These are called the *electromagnetic wave equations* because their solutions are very similar to classical wave solutions to the generalized wave equation, such as solutions describing sound waves or the waves formed by a vibrating string. But these waves are special - their speed of propagation is the speed of light. That is to say, the electromagnetic wave equations **describe light**, and their solutions describe all forms of light, from visible light in all its

colors to X-rays to gamma rays to the microwaves and radio waves that carry global communications and internet.

If we take a close look at the electromagnetic wave equations, electromagnetic waves are just a special case of electric and magnetic fields, with the special property that they oscillate throughout space without needing any wires or currents. This is also what makes them ideal for wireless applications, such as WiFi, communications, or in our case, wireless power transfer. They are generated when there are **both** time-varying electric and magnetic fields.

But what physical configurations can *generate* electromagnetic waves? Recall from Maxwell equation #3 (Faraday's law) that a changing magnetic field $\mathbf{B}(t, \mathbf{x})$ causes (induces) an electric field $\mathbf{E}_{\text{ind}}(t, \mathbf{x})$. The induced electric field $\mathbf{E}_{\text{ind}}(t, \mathbf{x})$ can then induce a changing magnetic field $\mathbf{B}_{\text{ind}}(t, \mathbf{x})$ by Maxwell equation #4 (Ampère-Maxwell law). However, for this to be true, the original magnetic field must satisfy the condition $\dfrac{\partial^2 \mathbf{B}}{\partial t^2} \neq 0$. This means it must not be a constant or linear function, because otherwise, the induced electric field from the original magnetic field would produce a constant (or zero) electric field by Maxwell equation #4. In mathematical terms, wave solutions require a current (density) that satisfies $\dfrac{\partial^2 \mathbf{J}}{\partial t^2} \neq 0$. That is to say, the current (density) must also be twice-differentiable in time and not constant or linear, so it cannot be a steady current. And so we have found our answer: in the case where currents are *non-constant*, which means they are *changing* with time, electromagnetic waves are produced.

The classical method of generating electromagnetic waves is to use some type of oscillating current, especially one that follows a sine or cosine curve (AC currents) - this is the basic operating principle of the **antenna**. Another method is to pass current through some form of spinning loop within a magnetic field (such as that generated by a magnet); the changing angle of the loop leads to induced electric fields that produce a changing current, generating (albeit weak) electromagnetic waves (typically radio waves, which can be amplified). A third, and the most common, method is to accelerate charges along a path; this principle is used in magnetrons for microwave ovens, where fast-moving (and accelerating) electrons move through a magnet field, producing (unsurprisingly) microwaves.

But what about the Sun, you may ask, or a light bulb, or a general hot object like a fire or plasma? In these cases, the mechanisms that produce light are *quantum* in nature. We will discuss how that occurs in the next chapter, "The Specifics".

0.1.4 Programming with Python

> "Science is what we understand well enough to explain to a computer; art is everything else."
> **Donald Knuth**

Having gone through an extensive amount of math and physics, we now arrive at the last topic in Chapter 1: scientific programming with the Python programming language. While programming is not a *required* skill to do research for Project Elara, it is a very *useful* skill, and in general, knowledge of programming is tremendously helpful for solving many problems in math and physics.

The theory of programming

In this section, we will be discussing programming using the **Python programming language**. A programming language is like an instructions sheet for a computer; a computer reads code written in a programming language, and runs according to the code's instructions. For instance, the following code in Python results in a message being shown ("printed") by the computer:

```
print("Hi!")
```

By nature, computers are not very smart; they can do arithmetic, read electronic data, and write electronic data, and that's about it. The exact process of "reading" and "writing" data is a bit more sophisticated - for instance, they can read data from a computer mouse that outputs electrical signals when it is moved, or write data to a hard drive (we call this a *filesystem*), or send data into a monitor that translates electrical signals into pixels on a screen. But computers fundamentally do not know what to *do*. The job of programming is for a person to write code that the computer can execute to *do things*, which is what allows computers to perform incredibly complex tasks, incredibly quickly.

Now, code is a broad category for a lot of things that share the common property of storing some form of information. Computer code, also called *source code*, is a specific type of code that can be understood by a computer (there is more nuance, but this is a place to start). Computer code is in many ways similar to mathematical notation; it is made of words and symbols that are arranged in a special way, designed to be human-readable, but following strict rules that encode meaning.

Computer code has different varieties; each of these is called a *programming language*. There are general-purpose programming languages and domain-specific programming languages; there are very old programming languages (some are more than 50 years old!) and very new programming languages; there are programming languages that are simpler to use, and programming languages that are *much* harder to use.

The Python programming language that we'll use is one of the **two main** programming languages used by Project Elara. It is designed for simplicity and user-friendliness, and is very popular and used by scientists worldwide. For this reason, Python is a *very* good language to start programming in. We will now introduce the basics of programming in Python.

> **Note**
> Have experience in Python, but want a refresher? We recommend reading the Learn X in Y minutes guide to Python to review how to write and use Python.

Using Python

There are two ways to be able to use Python: either by installing its specialized software tools from https://www.python.org/, which can take quite a bit of time and troubleshooting, or by using an *online code sandbox* such as Python Sandbox. We *highly* recommend using an online code sandbox to avoid the difficulties of setting up a computer for Python, which is an infamously error-prone process.

If you do choose to install Python on your computer instead of using the online sandbox option, the Python programming language requires that you write code in files with the extension `.py`, which

can then be run using the specialized `python yourcode.py` command (replace `yourcode.py` with the actual filename of your code file) in your operating system's terminal (Powershell for Windows, the Terminal app on Macs, your personal terminal on Linux). There are other ways to work with Python on your computer that we will cover later. Again, running Python on your computer is *not recommended* if you are just starting out and don't have experience with Python.

Variables and data types

The fundamental building blocks of any programming language, including Python, are *variables* and *data types*. In Python, variables are names you assign to a value. For instance, we may write:

```
myvariable = 42
```

We have now created a variable called `myvariable` whose value is the number 42. Variables can be modified (by assigning another value to them), and can be used later in your code wherever you'd like:

```
# note: the hashtag (#) starts a comment, meaning that
# anything after a hashtag on the same line is not
# considered code and not run by the computer

myvariable = 35 # modify variable
newvariable = 3 # create new variable
print(myvariable + newvariable) # use variable and print (display) it
```

Variables can store different types of data. We can give them descriptive names, like `number_one = 1` or `pi = 3.14159`. In Python, the most common basic types of data are strings, integers, floats, and booleans. We describe each of these below:

Data type	What it's for	Example usage
String	Anything involving characters, words, or text	`mystringvariable = "Hiii"`
Integer (int)	Any whole number (positive or negative)	`myintvariable = 20`
Floats	Any decimal number (we call these *floating-point numbers*)	`myfloatvariable = 1.414`
Booleans	A value that is either true or false	`is_sunny = True` or `is_rainy = False`, both `True` and `False` **must be capitalized** in Python

We can also perform *operations* on variables. For instance, we can do arithmetic on variables:

```
my_number = 3.0
my_number = my_number * 5
my_number = my_number - 1
my_number += 30 # this is shorthand for my_number = my_number + 30
my_number -= 12 # this is shorthand for my_number = my_number - 12
divisor = 5
my_number = my_number / divisor
my_number *= 15 # this is shorthand for my_number = my_number * 15
my_number += my_number ** 2 # here ** raises a number to a power
print(my_number)
```

For certain types of variables, we can also *compare* between two variables:

```
a = True
b = False
# Check if a is same as b
print(a == b)

# Check if a is NOT same as b
print(a != b)

# "is" and "not" can also be used for comparisons

# Check if a is true
print(a is True)

# Check if a is not true
print(a is not True)

# Check a is false
print(a is False)
```

We can also use the `>` (greater than), `<` (less than), `>=` (greater or equal to), `<=` (less or equal to) operations:

```
num_one = 5
num_two = 3
# Check if 5 is greater than 3
print(num_one > num_two)

# Check if 5 is less than 3
print(num_one < num_two)

# Check if 5 is less than or equal to 3
print(num_one <= num_two)
```

Functions

We often want to be able to repeat certain lines of code at different places in our code. We can use *functions* to do this. Functions contain a block of code that can be *called* (run) at some later point. In Python, functions are written with the `def` special keyword, the function name followed by `():`, and then an *indented block* containing the code, like this:

```
# create a function named perform_one_plus_one
def perform_one_plus_one():
 print(1 + 1)

# call (run) the function
perform_one_plus_one()
```

> **Note**
> Indentation is very important in Python! Consistency in indentation - that is, indenting by the same amount for code in the same block - makes sure that your code is valid Python code and can be run properly.

Putting a `return` keyword inside the function allows us to be able to save the output of a function to a variable, like this:

```
# create a function called ten_add_one
# that outputs one added to 10
def ten_add_one():
 return 1 + 10

# run the function and save its
# output to a variable
my_function_output = ten_add_one()
print(my_function_output)
```

We can also specify *arguments* to a function, which allow it to take one or more variables as inputs:

```
# create a function called add_numbers
# that adds two numbers and returns
# (outputs) the results
def add_numbers(a, b):
 return a + b

number_1 = 5
number_2 = 8
# save output of function to variable
sum = add_numbers(number_1, number_2)
print(sum)
```

Python (unlike some other programming languages) also allows defining *default arguments*, where one of a function's arguments can be omitted, and the function will substitute in the pre-set default value instead:

```
# create a function called circle_area
# that takes in an argument (the radius
# of the circle) and has a default argument
# for the circle constant pi
def circle_area(radius, pi=3.1415):
 return pi * radius ** 2
```

We can now execute the function using the default argument and omit the value of π:

```
my_circle_area = circle_area(5)
print("Circle area with default argument for pi:")
print(my_circle_area)

# let's now compute the area using a more precise
# value of pi. Although not required, we set the
# variable name in uppercase to show it's a constant
PRECISE_PI = 3.14159265358979

alternative_circle_area = circle_area(5, pi=PRECISE_PI)
print("Circle area with specified value for pi:")
print(alternative_circle_area)
```

Lastly, we should mention that Python has a number of *built-in* functions. Some of these allow inter-conversion of data types - for instance, `str(yourvar)` converts a variable to a string, `int(yourvar)` converts a variable to an integer, `float(yourvar)` converts a variable to a float, and `bool(yourvar)` converts a variable to a boolean (substitute `yourvar` with the variable to convert). Others allow for basic input-output, such as the `print()` function we've already seen. We will introduce a number of other built-in functions later.

Control flow

Now that we know how to create, use, and modify variables, as well as creating functions to run a block of code wherever we want it, we can cover the last important component of a programming language: **control flow**.

Control flow allows us to control the order in which our code is executed. For instance, we can choose to run or skip a block of code based on a condition. This is called a *conditional*, and the first type of conditional we will examine is an *if-statement*.

An **if-statement** in Python is comprised of the `if` keyword, followed by a condition (remember the boolean and comparison operations), followed by an indented block of code to run *if* the condition is true, as shown below:

```
my_number = 5

# Run our block of code ONLY if the condition
# ‘my_number > 3‘ is true

if my_number > 3:
 print("My number is greater than 3!")
```

We can also use the `not` keyword in if-statements, allowing us to express a block of code to run *if* the condition is false, as shown:

```
sunny_today = False

# Run our block of code ONLY if the boolean
# variable ‘sunny_today‘ is NOT true

if not sunny_today:
 print("Today isn’t a sunny day.")
```

We can add an *else* block after an if-statement to immediately run a block of code if the condition in the if-statement is not satisfied, like this:

```
sky = "Blue"

if sky == "Purple":
 print("Sky is purple!")
else:
 print("The color of the sky is:")
 print(sky)
```

We often to run another if-statement following an else-statement. We *could* write this by nesting (placing) our second if-statement in our first if-statement, like this:

```
name = "John"

if name == "David":
 print("Name is David")
else:
 if name == "John":
  print("Name is John")
```

But if we have multiple if-statements to run after, this becomes very tedious to write and hard to read. Instead, Python offers an `elif` key word (short for “else if”), that simplifies the above code to:

```
name = "John"

if name == "David":
 print("Name is David")
elif name == "John":
 # runs code if the name is not David
 # and the name is also John
 print("Name is John")
```

Python also allows us to re-execute a line of code multiple times - these are called *loops*. The first type of loop is called a **while loop**. While loops are formed with a `while` keyword, followed by some condition, and then an indented block of code, like this:

```
x = 1
while x < 3:
 # code in this block is re -run
 # as long as x is less than 3
 print("X is", x)
 x += 1
 print("Added one to x, continuing to loop...")

print("Loop finished")
```

The block of code within the while loop repeats as long as the condition is satisfied, but once the condition is *no longer satisfied*, the while loop exits. In our previous example, our condition was that $x < 3$, and so long as this was true, the code block within the while loop repeats. But we add 1 to x on every loop - that's what `x += 1` does - and so after running twice, the value of x is 3. This means that $x < 3$ is *no longer true*, and so the loop is ended, and we say we have *exited* the loop.

While loops are good for certain tasks, but they have one major issue - you must carefully keep track of the condition to make sure that the loop *actually* exits. It is easy to accidentally write a loop that never exits (that is, until either your operating system force-shutdowns your running Python or you get a computer crash). Here is an example:

```
while True:
 print("Looping forever...")
```

Do *not* actually try to run this code for the reasons just mentioned! But since `True` is always true, this while loop never exits, and runs forever. This makes while-loops a potential source of issues, so there is another type of loop that is often used instead. This is the **for-loop**.

Instead of exiting only on a certain condition, a for-loop *iterates* (loops) through a finite number of items, which means that unless if you do something really really hacky, for-loops always exit and won't run forever. A for-loop can take several different forms, and we will show the most common of them here. The first is iterating across a range of numbers. The for loop starts with the `for` keyword, then defines a variable to hold the current number in the range, then defines the range, and finally, there is an indented code block that runs on every iteration. We show this below:

```
# We run a block of code once
# for every number less than 10,
# starting from 0 (i.e. all numbers
# between 0 and 9)

for number in range(10):
 # the ‘number‘ variable holds the
 # current number
 print("Current number is", number)
```

```
 # the for loop runs for every number
 # until 9
```

There are other types of for-loops that we will cover shortly, once we reach the next section.

Objects, lists, and dicts

By this point, we have covered the most basic Python functionalities. These functionalities - variables, arithmetical and logical operations, functions, and control flow - are not unique to Python; almost every programming language has them, and a programming language is basically *required* to have them in order to be considered a programming language (this is called *Turing completeness*, although that is a far more vast topic). But Python extends these functionalities with an extensive *standard library* that provides more features.

To understand the Python standard library, let's first discuss what a *software library* is. A software library is a collection of software code that can be used by other code to provide additional functionality. Python's standard library is an example of a software library, with the special property that it is *distributed with* the Python language tools when you install it - that's why we call it a *standard* library. Other libraries are typically downloaded, and we call these libraries *extenal libraries*.

Python's standard library includes more complex data-types called *objects* (this is technically an oversimplification but will do for now). Objects are data-types that have properties and have their own specialized functions, which we call *methods*. Among the two most common objects in Python are lists and dictionaries.

Lists are Python's way to, unsurprisingly, make sense of a group of things. For instance, we can make a list of numbers, or a list of integers:

```
ls1 = [1, 2, 3, 4, 5] # our first list
ls2 = ["a", "b", "c", "d", "e"] # our second list
```

You can put any data type you want in a list:

```
ls3 = [1, "a", 2, "b", 3, 3.78592, True, False]
```

You can put lists inside of lists:

```
ls4 = [[5, 7, 9], ["a", "b", "c"], [3, "vedant", True]]
```

We can put variables inside lists as well:

```
ls5 = [ls1, ls2, ls3, ls4]
```

To find the size of a list, we use `len()`:

```
len(ls1)
```

To access the elements of the list, we use bracket syntax. However, note that in Python, we start counting from zero, so the first element is index 0:

```
print(ls1)
print("First element:", ls1[0])
```

Similarly, the second element is index 1, the fifth element is index 4, and the 100th element is index 99:

```
print(ls2)
print("Second element:", ls2[1])
```

We can also count backwards, in which case the last element is index -1:

```
print(ls1)
print("Last element:", ls1[ -1])

print(ls3)
print(ls3[ -1])
```

The usefulness of a list is that we can store a collection of different pieces of information together, and operate on them all simultaneously. This is where **iterators** come in - they allow us to apply an operation on every element in a list. One type of iterator is our familiar **for-loop**, and now we will see how it is perfectly suited to working with lists.

For instance, suppose we wanted to increase every element in a list by one. We could always do this:

```
numbers_ls = [1, 25, 389, 1578, 12903]

numbers_ls[0] += 1
numbers_ls[1] += 1
numbers_ls[2] += 1
numbers_ls[3] += 1
numbers_ls[4] += 1

print(numbers_ls)
```

But for any longer example, it would be exhausting to write every single case. Instead, we could write a **for loop**:

```
numbers_ls = [1, 25, 389, 1578, 12903]

for i in range(len(numbers_ls)):
    numbers_ls[i] += 1

print(numbers_ls)
```

Let's take a closer look at this for-loop. In that loop, `range(len(numbers_ls))` returns 5, which tells us we're going to repeat the for loop's operation 5 times. Each time, we have a variable `i` - the first time, `i` will be equal to 0 (remember Python counts from 0), the second time, `i` will be equal to 1, the third, `i` will be equal to 2, and so on, until `i` is equal to 4.

Our operation then takes the ith element of the list and adds one to itself. For example, if `i = 0`, then we're really doing `numbers_ls[0] += 1`; if `i = 1`, then we're doing `numbers_ls[1] += 1`; and so on and so forth.

What if we just wanted to print out all the elements of our list? We could write that with a for-loop as well:

```
# here we don't have to use range()
# just the list itself
for number in numbers_ls:
 print(number)
```

Aside from iterating over lists with for loops, Python has some standard defined functions for lists. For example, we can add to a list:

```
numbers_ls2 = [4, 5, 6, 7]
numbers_ls2.append(8)
print(numbers_ls2)
```

We can remove elements from a list:

```
numbers_ls2_clone = numbers_ls2.copy() # make a copy of a list
numbers_ls2_clone.pop() # remove last element
numbers_ls2_clone.pop(1) # remove element at index 1
print(numbers_ls2_clone)
```

And most usefully, we can slice a list. A slice of a list `[a:b]` returns a list **starting** from index a and **ending before** index b:

```
numbers_ls2 = [4, 5, 6, 7, 8]

# Slice with n: returns list STARTING from index n
numbers_ls2[2:]
# [6, 7, 8]
numbers_ls2[3:]
# [7]

# Slice with :n returns list ENDING before index n
numbers_ls2[:2]
# [4, 5]

# Double slice returns list STARTING from
# index n and ENDING before index n
numbers_ls2[1:3]
# original: [4, | 5, 6, | 7,]
# sliced: [5, 6]
```

Finally, we can use `index()` to find the index of an element within a list:

```
fruits = ["apples", "oranges", "pears"]
fruits.index("apples") # returns index 0
```

After lists, we have tuples, which are like lists, but unmodifiable, meaning that you **cannot** change elements within a tuple. We write tuples using rounded brackets:

```
tup1 = (1, 2, 3, 4)
# So this doesn't work:
# tup1.pop()
```

We can find the number of elements in a tuple in the same way as a list:

```
len(tup1)
```

We can't change tuples but we can slice them and save the result to a **new** tuple:

```
tup2 = tup1[1: -1]
print("original:", tup1)
print("sliced:", tup2)
```

We can unpack tuples (this works for lists as well):

```
coords = (3.5, -8.0, 10.3)
x, y, z = coords
print(x)
```

Finally, the last of the commonly-used collection types is the dictionary. Dictionaries are used to store **structured** data. In more technical terms, they map keys (entries) to values (definition of each entry).

We make a dictionary by entering keys with their corresponding values. Note keys for dictionaries have to be fixed-memory types (for technical reasons). Fixed-memory types include ints, floats, strings, and tuples. In the below example, we create a dictionary with two keys, `name` and `age`:

```
our_ages_dict = { "name": "John", "age": 30 }
# we can add a key -value pair like this:
our_ages_dict["nationality"] = "United States"

print(our_ages_dict)
```

We access dictionary values with `[]` where the key is inside the brackets:

```
our_ages_dict["name"]
```

The other way to get a key is to use the `get()` function:

```
our_ages_dict.get("name")
```

If you try to find an invalid key, it will return `None`:

```
# the only keys are "name", "age", and "nationality",
# and "home" is not one of them
our_ages_dict.get("home")
```

To get all the keys, use the `keys()` function, which will return the keys as a list:

```
list(our_ages_dict.keys())
```

Use the `values()` function to do the same thing with the values:

```
list(our_ages_dict.values())
```

We can use "in" to check if a key is in a dict:

```
"name" in our_ages_dict
```

Next steps

While the abilities of Python with just its standard library are already extensive, the real power of Python is its ease of using *external* libraries. There exists a Python library for almost anything, and you can browse through them on `https://pypi.org/`, the official site for Python libraries. Many Python libraries are so extensively used that they have become industry-standard and you will rarely find a research publication that involve some sort of computational science that *don't* use them.

At Elara, we make extensive use of Python and its libraries. We will cover the scientific computing libraries used in Project Elara in the next chapter, as well as more advanced programming techniques such as object-oriented programming, list comprehensions, lambda functions, decorators, and vectorized operations. But a working knowledge of Python programming is already enough to get you far.

0.2 The specifics

0.2.1 Writing in Markdown and LaTeX

Please see https://codeberg.org/elaraproject/elara-labs/src/branch/main/tutorial-for-latex.pdf

0.2.2 Comprehensive guide to programming

In an earlier chapter of the Handbook, we've already discussed the fundamentals of programming using Python. But there is a lot more to programming than Python! In this section, we'll discuss programming in a variety of languages used in Project Elara.

Programming in JavaScript

Standard Library and Imports JavaScript has a built-in standard library available in all environments. In Node.js, modules can be imported using `require()` or `import`. Some examples:

- `import fs from 'fs';` for reading and writing files
- `import path from 'path';` for working with file paths
- `import readline from 'readline';` for reading user input in the terminal

Comments

```
// single line comment
/* multi
line
comment */
```

Data Types and Variables Each variable is a container for a value. JavaScript is dynamically typed, so the type is inferred from the value. Variables are declared using `let`, `const`, or `var`.

- `let variableName = value;` — declares a block-scoped variable that can be reassigned
- `const variableName = value;` — declares a block-scoped variable that cannot be reassigned
- `number`: stores integers and decimals
- `string`: stores text
- `boolean`: stores either `true` or `false`
- `null`: represents an intentional absence of value
- `undefined`: a variable that has been declared but not assigned a value
- `object`: stores collections of key-value pairs
- `array`: stores ordered lists of values (a special kind of object)

User Input and Output In a browser, `console.log()` prints to the developer console. In Node.js, the `readline` module is needed for user input.

```
console.log("Hello, World!");            // printing to the console

// Reading user input in Node.js
import readline from 'readline';

const rl = readline.createInterface({ input: process.stdin });

rl.question("What is your name? ", (name) => {
    console.log("Hello " + name);
    rl.close();
});
```

Operators Operators are used to manipulate variables in different ways.
Arithmetic (mainly used for numbers):

```
+  // Addition
 -  // Subtraction
*  // Multiplication
/  // Division
%  // Modulus (gives the remainder)
** // Exponentiation
++ // Increment (adds 1 to variable)
 - - // Decrement (subtracts 1 from variable)
```

Assignment Operators (Shorthand version of doing an operation and then assigning it to the variable):

```
+=
 -=
*=
/=
%=
**=
```

Comparison Operators (Compares two statements and returns a boolean):

```
===  // strict equality (checks value AND type)
!==  // strict inequality
>
<
>=
<=
```

Logical Operators (Can also compare two statements and returns a boolean):

```
&&  // and
||  // or
!   // not
```

Programming in C and C++

Standard Template Library (STL) C++ uses the standard libarary for different variables and functions.
`std::` must be added to the beginning of variable/function.
By writing `using namespace std;`, `std::` does not have to be written.
Some examples:

- `#include <iostream>;` for user input and output
- `#include <string>;` to use std::string data type

Comments

```
// single line comment
/* multi
line
comment */
```

Data Types and Variables Each variable is a container for a value with a data type. `type variableName = value;`

- `int`: stores integers
- `float` & `double`: stores decimal values, double can store more decimal places
- `char`: stores a single letter
- `std::string`: stores text
- `bool`: stores either `true` or `false`

User Input and Output The `<iostream>` library is needed for user input and output.

```
std::string name;                          // declaring a variable
std::cin >> name;                          // taking in user input and storing it in the varia
std::cout << "Hello " << num << std::endl; // printing to the console (std::endl creates a new
```

Operators Operators are used to manipulate variables in different ways.
Arithmetic (mainly used for integers and decimals):

```
+ // Addition
 - // Subtraction
* // Multiplication
/ // Division
% // Modulus (gives the remainder)
++ // Increment (adds 1 to variable)
 - - // Decrement (subtracts 1 from variable)
```

Assignment Operators (Shorthand version of doing an operation and then assigning it to the variable):

```
+= // addition assignment
 -= // subtraction assignment
*= // multiplication assignment
/= // division assignment
%= // modulus assignment
```

Comparison Operators (Compares two statements and returns a boolean):

```
== // equal to
!= // not equal to
> // greater than
< // less than
>= // greater than or equal to
<= // less than or equal to
```

Logical Operators (Can also compare two statements and returns a boolean):

```
&&  // and
||  // or
!   // not
```

C++ Strings To use strings and string functions, `<string>` is needed.

```
#include <string>;

std::string a = "Hello";
std::string b = " World";

// String concatenation (adding two strings together will create a new string that contains bo
std::string c = a + b;

// Returns the length of the string, .size() does the same
int length = c.length();

// Acessing and changing the first character of the string
b[0] = 'Z';

// C style strings (array of char data types)
char d[] = "apple";

// Escape characters (Special characters that can be added to a string)
\' // Single quote
\" // Double quote
\\ // Backslash
\n // New line
\t // Tab
```

Statements and Loops

```
// If statement
int a;
int b;
if (a > b) {
  std::cout << "a is greater";
} else if (b > a) {
  std::cout << "b is greater";
} else {
    std::cout << "a is equal to b";
}

// While loop
int i = 0;
```

```
while (i < 5) {
  std::cout << i;
  i++;
}

// For loop
for (int i = 0; i < 5; i++) {
  std::cout << i;
}

// Break & continue statements
break; // when placed inside a loop and the break; is reached, the loop will stop and move to
continue; // when placed inside a loop and continue; is reached, the loop will skip the remain
```

Functions

```
// Creating a function
// Functions can return any data type and is declared before the function name (void means ret
// Parameters can be included which pass variable into the functon, or can have no parameters
// By default a variable is passed by value, meaning the variable is copied and any changes to
// & symbol before the parameter is passing a variable by reference. This means the variable i
void functionName( // dataType param1, dataType& param2, ...) {
  // code here
}

// Declaring the function
int main() {
  functioName( // param1, param2, ... );
  return 0;
}
```

Arrays

```
// To declare an array, define the data type stored in the array and the number of arguments
// Arrays can only store one type of data type and elements cannot be added or removed
int numbers[4];
std::string fruits[3] = {"apple", "banana", "orange"};

// Accessing and changing array elements
std::string fruit = fruits[2];
fruits[0] = "pineapple";

// Traversing an array
for (int i = 0; i < fruits.size(); i++) {
  std::cout << fruits[i];
}

// Traversing using for each loop
for (int fruit : fruits) {
  std::cout << fruits[i];
}

// Multidimensional arrays (array with arrays inside)
int matrix[5][5];
```

Pointers

```
// Pointers are variables that store the memory adress of another variable
// The memory address is where a variable is stored which can be referenced

int x = 5;
int* ptr = &x; // & means get the adress of the variable
std::cout << ptr << std::endl; // prints something like 0x23f4e6

// Dereferencing pointers (this gets the value of the variable referenced by the pointer)
std::cout << *ptr << std::endl; // prints 5

// Changing pointer values
*ptr = 7; // this also changes x

int y = 3;
int* ptr2 = &y;
ptr = ptr2; // ptr now points to ptr2's variable which is y
```

File I/O

```
#include <iostream>
#include <fstream> // needed to use files

int main() {
  // Writing to a file
  std::ofstream myInputFile("filename1.txt"); // create and open the file
  myInputFile << "Hello World" << std::endl; // writing to the file
  myInputFile.close(); // close the file

  // Reading to a file
  std::ifstream myOutputFile("filename2.txt"); // open input file
  std::string line;  // used to read line by line of the input file
  while (getline(myInputFile, line)) {
    std::cout << line << std::endl;
  }
  myInputFile.close();
}
```

Command line arguments

```
// C++ programs can take arguments from the command line when executing code from the terminal
// argc is the number of command line arguments, (the program name itself is one of the argume
// argv is a vector(a type of array) that stores those arguments as c style strings
int main(int argc, char* argv[]) {
  // checking number of arguments
  if (argc != 2) {
    std::cerr << "Wrong number of arguments"; // prints an error message to the console
    exit(1); // exits the program
  }

  // accessing the arguments
  std::string programName = argv[0];
  std::string argument1 = argv[1];
}
```

Structs

```
// Structs are containers that can group variables into one data type
struct Coordinate {
  int x;
  int y;
};

// Acessing and setting variables of the struct
int main() {
  Coordinate p1;
  p1.x = 0;
  p1.y = 1;
}
```

Classes & Objects

```
// Classes are templates for objects. Classes by default have private member variables and pub
class Person {
// Member variables
private:
  std::string name;
  int age;
  std::string job;

// Member functions
public:
  Person(std::string& personName, int& personAge, std::string& personJob) : name(personName),
  std::string getName() const { return name;}
  void setJob(const std::string& jobTitle) { job = jobTitle; }
};

int main() {
  Person john("John", 30, "Scientist");
  Person jane("Jane", 25, "Engineer");

  john.setJob("Architect");
  std::string name2 = jane.getName();
}
```

Vectors

```
// Vectors are dynamic or resizeable arrays
#include <vector>

std::vector<int> nums = {1, 2, 3, 4, 5};

// Acessing and changing elements
int x = nums[2];
nums[0] = x;

// Adding an element
nums.push_back(6);

// Removing the last element
```

```
nums.pop_back();

// Get first and last elements
int y = nums.front();
int z = nums.back();

// Check if vector is empty & remove all elements from the vector
bool isEmpty = nums.empty();
nums.clear();

// Traversing vector
for (int num : nums) {
  std::cout << num;
}
```

Iterators

```
// Iterators are general form of pointers that can be used to traverse containers
std::vector<int> nums = {1, 2, 3, 4, 5};

std::vector<int>::iterator it;  // Used to traverse containers, access & remove elements
std::vector<int>::const_iterator itr;  // Iterator values cannot be changed, used to traverse

for (itr = nums.begin(); itr != nums.end(); ++itr) {
  std::cout << *it; // Dereference iterator to access value
}
it = nums.erase(it);
```

Numerical computing

Computers evolved from calculators to solve complicated math problems. Unsurprisingly, the relationship between computers and math has stayed to this day. Computers are simply *incredibly good* at solving problems that would take ages to do by hand. This is the field of **numerical computing**: a field of programming centered around doing math on a computer as *efficiently* as possible.

In this chapter, we will primarily be using the Python language, for its simplicity and its ubiquity in numerical computing. In fact, Python is so frequently used in numerical computing that the "Scientific Python stack" is well-understood to be composed of a set of Python libraries specialized for numerical computing. In this chapter, we will explain how to use these libraries to implement mathematical calculations, solve scientific problems, and make visualizations and graphs.

Using the NumPy library First, import NumPy, which is often abbreviated as `np`:

```
import numpy as np
```

NumPy is used when working with arrays or lists. These lists can contain any type of object and can be any size. Using NumPy makes creating and manipulating arrays much easier because it provides plenty of prebuilt functionality.

```
arr_1 = np.array([1, 2, 3])
arr_2 = np.array(['a', 'b', 'c'])
```

Numpy arrays can also nest arrays within arrays. These shapes can become very complicated so using array.shape can give you the ovaral dimentions of a numpy array.

```
arr_1 = np.array([1, 2, 3])
arr_2 = np.array([[1, 2, 3]])
arr_3 = np.array([[[1, 2, 3],[4, 5, 6]],[[7, 8, 9], [10, 11, 12]]])
arr_4 = np.array([[1], [2], [3]])
print(f"arr_1 shape: {arr_1.shape}")
print(f"arr_2 shape: {arr_2.shape}")
print(f"arr_3 shape: {arr_3.shape}")
print(f"arr_4 shape: {arr_4.shape}")
```

NumPy arrays have all the same attributes as normal arrays. Like normal arrays they can be indexed starting at 0.

```
arr_1 = np.array([1, 2, 3])
arr_2 = np.array([[1, 2, 3]])
arr_3 = np.array([[[1, 2, 3],[4, 5, 6]],[[7, 8, 9], [10, 11, 12]]])
print(f"arr_1 1st index: {arr_1[0]}")
print(f"arr_1 2th index: {arr_1[2]}")
print(f"arr_3 multiIndexed: {arr_3[1][0][1]}")
print(f"arr_3 listIndex: {arr_3[0][1]}")
```

You can use NumPy to create default arrays, such as an array of zeros or ones.

```
arr_zeros = np.zeros(4)
arr_ones = np.ones(1)
arr_random_trinary = np.random.randint(0, 3, size=10)
print(f"arr_zeros: {arr_zeros}")
print(f"arr_ones: {arr_ones}")
print(f"arr_random_trinary: {arr_random_trinary}")
```

NumPy has built-in functions that save the user from having to implement them. Some of the more useful functions include np.max(), np.min(), np.mean(), np.median(), and np.std().

```
arr_1 = np.array([1, -2, 3, 7, 8, 0, 20, 5, 3])
print(f"arr_1 maxValue: {arr_1.max()}")
print(f"arr_1 minValue: {arr_1.min()}")
print(f"Mean of array: {arr_1.mean()}")
print(f"Median of array: {np.median(arr_1)}")
print(f"Standard Deviation of array: {arr_1.std()}")
```

You can apply basic operations to arrays of similar sizes, such as addition, subtraction, and other arithmetic operations.

```
arr_1 = np.array([1, 2, 4])
arr_2 = np.array([2, 2, 3])
arr_3 = np.array([[1, 2],[3, 4]])
print(f"(arr_1 + arr_2): {np.add(arr_1, arr_2)}")
print(f"(arr_1 - arr_2): {np.subtract(arr_1, arr_2)}")
print(f"(arr_1 * arr_2): {np.multiply(arr_1, arr_2)}")
print(f"(arr_1 / arr_2): {np.divide(arr_1, arr_2)}")
print(f"(arr_1) + (arr_2): {np.concatenate((arr_1, arr_2), axis=0)}")
print(f"sqrt(arr_1): {np.sqrt(arr_1)}\n")
print(f"Transposed arr_3: \n{arr_3.T}")
```

Embedded development

Background

What is a Microcontroller? A microcontroller can be thought of as a compact and low-powered computer. They run on a single chip and are used for specific tasks, such as reading sensors and controlling motors. A microcontroller utilizes pins to carry out operations. These are called General Purpose Input Output pins, otherwise known as GPIO pins. When using a microcontroller, it is important to know its GPIO voltage and current restraints. By not adhering to these constraints, we risk damaging the microcontroller (aka "frying it"). This means possibly melting internal circuitry and shorting input/output pins.

Microcontrollers vs Microprocessors One mistake many make is confusing microprocessors for microcontrollers and vice versa. While they may sound similar in name, they are different in nature. A microprocessor is a general purpose CPU that requires external memory and peripherals(these are other hardware components). On the other hand, microcontrollers have processors, memory, and peripherals. They both have their own place where one outperforms another. The table below from Dr. Urvashi Singh's Introduction to Microprocessors and Microcontrollers does a great job showing the difference.

DIFFERENTIATING FEATURES

FEATURES	MICROPROCESSOR	MICROCONTROLLER
Function	**Process** the **general** task, only	Both **Process** and **Control** the **specific** task
Memory	**No** in-built memory	In-built ROM and RAM memories
Application	**General** purpose (eg. PC, modems, printers)	**Specific** purpose (eg. A.C. and washing machine, home automation)
Complexity	**More** complex, **large** no. of instructions	**Less** complex, **less** no. of instructions
Cost	**High** (design time is more)	**Low** (design time is low)
Efficiency	**Less**	**More**
Architecture	**Von Neumann** (program and data stored in same memory)	**Harvard** (program and data stored in different memory)
Example	8085, 8086	8051

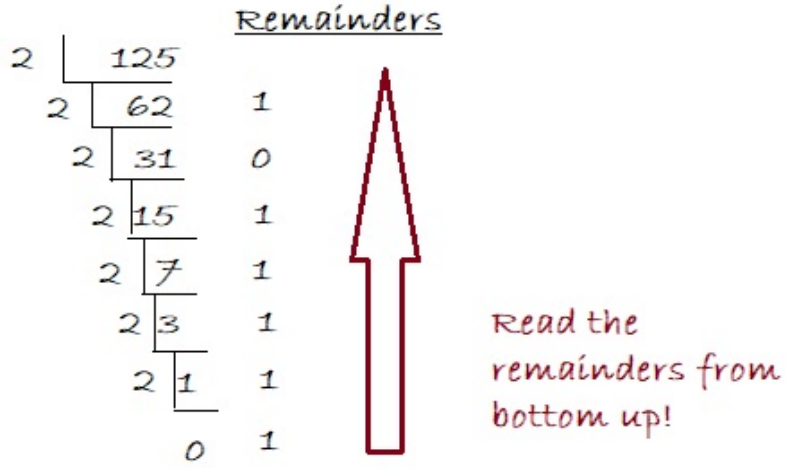
Skills and Protocols Before you start using microcontrollers, there are a couple things that are beneficial to know beforehand. If you have prior C/C++ experience, you're in great shape to start, but don't worry if you don't, we have a C/C++ guide you can learn from. It would most definitely be helpful to familiarize yourself with how to create a circuit and know what to look out for and know how to they work. Most microcontrollers are coded using C/C++ but some are also coded using Java, Python, Rust, and other languages.

Overtime, you'll learn about protocols such as I2C, SPI, and UART. Protocols like these determine how the microcontroller shares and utilizes data from peripherals(such as sensors as you may recall). However, we'll get to that later.

Number Systems You may be wondering, why do I have to learn about number systems to code a microcontroller? Microcontrollers, FPGAs, and circuits all require a basic knowledge of them. Numbers are the basis to how a computer works. Technically speaking, microcontrollers use base 2, base 10, and base 16, known as binary, decimal, and hexadecimal respectively. Microcontrollers store data in binary, where the 0s and 1s are bits and can be used to interpret voltages. Hexadecimal is used for memory maps, registers, and memory addresses. Both are essential for digital logic and circuits in general.

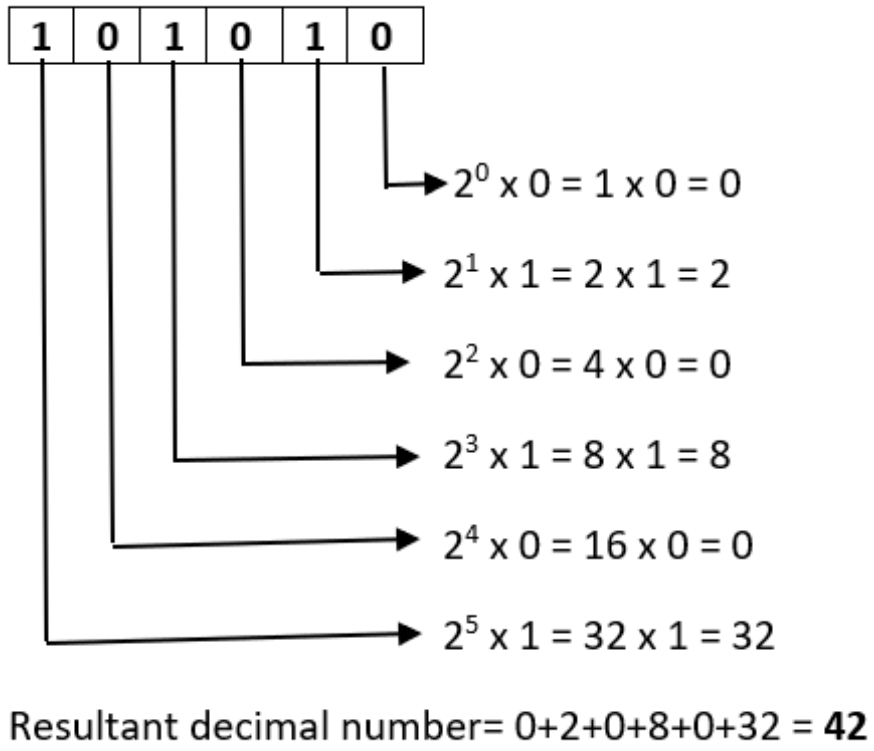
Binary We'll start off learning about binary. The decimal system is probably the one you think of using when you count, and we can use this for binary. Base 10 means that there are 10 digits being used to represent the number, which culminates in making groupings of tens. Binary will use just 2 digits, 0 and 1. How do we convert from these two systems then?

Decimal to Binary To convert from base 10 to base 2, the simplest method is to use divison. This means dividing the base 10 number by 2 repeatedly while recording the remainder, using the quotient as the new dividend while 2 remains the constant divisor until we get a quotient of 0 and some remainder. The binary number will be the remainders from the last to the first. Here is an example from Wendy Qui's AP Computer Science lesson:



Therefore, 125 in decimal is 1111101 in binary.

Binary to Decimal To convert from base 2 to base 2, the simplest method is to use the positional value method. This means going from the least significant bit(LSB), which will be the number in the right-most place, to the most significant it(MSB) and assigning it a power of 2, always starting from 0 being the LSB. We multiply the bit by this power of two, and sum all of the bits till we get our decimal number. Here is an example from Geeks for Geeks:



Therefore, 101010 in binary is 42 in decimal.

0.2.3 Guides to essential software

In Project Elara's research, there are many tools and software that we use frequently, to the point that they are essential to our work. The following chapters will cover some of these software.

0.2.4 Machine learning

To conclude our foray into software and software-related topics, we will go over **machine learning**. Machine learning is essential to the modern world and finds many applications in the sciences (we will go over examples in later chapters), but it can be quite difficult to understand. Its formidability often comes from its heavily developer-oriented nature, the difficulties of preparing and working with datasets (which often are incomplete and in a format unsuitable for machine learning), and the complexity of tuning hyperparameters — that is, configuration variables — to get a neural network to fit a dataset. Machine learning is both an art and a science, and a finicky one at that. But it is also, for better or worse, a powerful technology. And for any data-heavy field, it is an invaluable tool.

Fundamentals of machine learning

Machine learning is the process of using a computer to automatically create a mathematical model. The main difference between machine learning and other modeling techniques is that machine learning doesn't give a computer an algorithm for a *specific task*, but instead helps the computer *develop* an algorithm on its own. Machine learning models have proven to be massively successful for a variety of tasks, including classification, object detection, and machine translation, among many others.

The simplest type of machine learning uses a model called a **multi-layer perceptron** (MLP). This type of model is based on two mathematical operations, matrix-vector multiplication and vector addition. It takes in an input variable (for instance, size, position, color, etc.) encoded as a variable $\mathbf{x}$, then performs the operation $f(\mathbf{x}) = \sigma(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$. The matrix $\mathbf{w}$ is known as the **weights**, and the vector $\mathbf{b}$ is known as the **biases**.

In machine learning, we divide a dataset into two parts: the inputs $\mathbf{x}$ (also called the *features*), and the outputs $\mathbf{y}$ (also called the *labels*). For instance, if we were to build a model to predict whether an image is a cat or a dog, then the features would be a list of images and the labels would be a list of `["cat", "cat", "dog", "dog", ...]` that corresponds to the correct animal for each image. When the model outputs a prediction $\mathbf{y} = f(\mathbf{x})$ for some given input vector $\mathbf{x}$ (for instance, a vector of image pixels), we compare it to the real value in the dataset $\mathbf{y}_{\text{true}}$.

The mathematical way to represent this comparison is the **loss function**, which measures the error in the model's predictions. The loss function $\mathbf{L}(\mathbf{x}, \mathbf{y}_{\text{true}})$ compares each element of the predicted outputs with their real values in the dataset and squares the difference, by the following equation:

$$L(\mathbf{x}, f(\mathbf{x}), \mathbf{y}_{\text{true}}) = \sum_i (\mathbf{y}_i - \mathbf{y}_{\text{true},i})^2, \quad \mathbf{y}_i = f(\mathbf{x}) \tag{525}$$

The model "learns" by minimizing the loss function, until the model reaches a minimum in the error of its predictions. This is accomplished by the process of **gradient descent**, which comes from multivariable calculus. Recall that the gradient points in the direction of steepest ascent, meaning that the *negative* of the gradient points in the direction of steepest *descent*. Therefore, we calculate the gradient ∇L for the loss function, and adjust the weights and biases based on the gradient, as given by the following equation:

$$\begin{aligned} \mathbf{w}_{n+1} &= \mathbf{w}_n - \nabla L \\ \mathbf{b}_{n+1} &= \mathbf{b}_n - \nabla L \end{aligned}$$

If we think of the loss function as a surface $z = L(\mathbf{x}, f(\mathbf{x}), \mathbf{y}_{\text{true}})$, then the gradient descent procedure is equivalent to finding a path to the minimum of the surface, as we show in Figure 1.

We may repeat as many iterations of gradient descent until we are satisfied with the model's performance (that is, the error of the model is low enough for our purposes). The model has now "learned" from the dataset, and it can make accurate predictions even on data that is not in its dataset.

However, while the MLP model is relatively simple to understand (compared to other model types), it is harder to train and use compared to other, more mathematically-sophisticated machine-learning

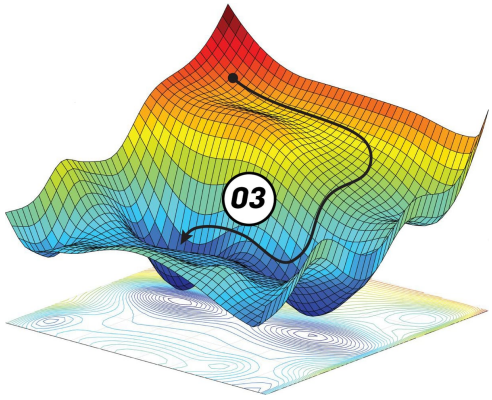


Figure 1: Gradient descent uses the gradient of a function to figure out which direction to go to iteratively approach its minima. Image credit: Paperspace/DigitalOcean.

models, such as Kernel Ridge Regression (KRR), Extra Trees Regression (ETR), Random Forest Regression (RFR), and Support Vector Regression (SVR). There are also other types of neural networks, like Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), Transformers, and even others. Here, we will not strive to explain every type of machine-learning model, but simply explain the broad concepts and show how you can use machine learning in practice.

Coding the model

As the details of machine learning are quite complicated, it is convenient to use libraries that already implement many of the functions and classes for making machine learning models.

0.2.5 Advanced classical physics

Having spent the last few chapters on software and programming, it's time to return to physics! So far, we've tackled calculus, vectors, matrices, and Newtonian mechanics; now we're ready to move on to more advanced topics. In the following chapters, we will cover (among other things) more advanced electrodynamics and quantum mechanics, as well as delving deep into the theory behind Project Elara's research and understanding how to perform calculations, run simulations, and perform experimental testing. By the end of this chapter, you will have gained a solid understanding of Project Elara and will be ready to perform research that might one day change the world - we hope that the reading will be very much worth it.

Lagrangian and Hamiltonian mechanics

Newton's laws and conservation of energy are two approaches to solving for the equations of motion of an object. We can make Newtonian mechanics more elegant by extending them to fields and potentials. But ultimately, Newtonian mechanics is still cumbersome to use. Here is an alternate, more beautiful approach - Lagrangian mechanics.

Lagrangian Mechanics The **Lagrangian** is the difference of an object's kinetic and potential energies, and is denoted by:

$$\mathcal{L}(\S, \dot{\S}) = K(\dot{x}) - U(x) \tag{526}$$

Note that the dots are used for the time derivatives - that is, $\dot{x} = \frac{dx}{dt}$. The **action** is a fundamental quantity of all physical systems and is given by the time integral of the Lagrangian:

$$S = \int_{t_1}^{t_2} \mathcal{L}(x, \dot{x})\ dt \tag{527}$$

The **principle of stationary action** states that for any given system, the action is stationary. What does stationary mean? Recall the idea of stationary *points* in calculus - which include minima and maxima. For the action to be stationary, that means the Lagrangian must be a stationary *function*, which are analogous to stationary points, just for the action, which is a function of functions (what we call a *functional*, which we'll go more in-depth with later).

But what form does that Lagrangian have to take to obey the principle of stationary action? The short answer is that it must obey the following equation, known as the **Euler-Lagrange equation**:

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} = 0 \tag{528}$$

where again, $\dot{x} = \dfrac{dx}{dt}$ is the velocity. This is one of the most fundamental and profound equations of physics, and works for any particle's Lagrangian (particle, remember, can be a big object like a planet or star, it is a generic term in physics). Once you write down the Euler-Lagrange equation, you just need to take the derivatives of the Lagrangian and substitute to get the equations of motion (the differential equations you use to solve for the trajectory of the particle). Applying it, at least conceptually, is fairly simple.

But to gain a deeper understanding of *why* this equation works, we must first dive into the theory of **functionals** and **variational calculus**. If this section is too math-heavy, feel free to skip this section - it's not required for applying the Euler-Lagrange equation. But for those that want the step-by-step derivation - let's dive in!

Functionals A **functional** is a function that takes in *other* functions as input. In contrast to a function f which takes in a real number x and outputs $f(x)$, a functional $\mathcal{L}$ takes in a function $y(x)$ and outputs $\mathcal{L}(y(x), y'(x), x)$.

The derivative appearing in $\mathcal{L}(y, y', x)$ and the mention of the word "calculus" suggests that functionals are based in **differential operators**, such as derivatives and integrals. Indeed, this is the case - a great number of functionals are in fact integrals.

Consider, for instance, a functional that appears - under a different name - in a first introduction to calculus. This is the functional expression for the **arc length**:

$$S = \int_a^b \sqrt{1 + y'^2}\, dx \tag{529}$$

While an introductory treatment of calculus may simply give this formula with the provided function y and its derivative y', the calculus of variations would consider this formula a **functional** of an *arbitrary* function f in the form:

$$S(y, y', x) = \int_a^b \sqrt{1 + y'^2}\, dx \tag{530}$$

The calculus of variations is concerned with **optimizing** functionals to find their stationary points. In many cases, we want to obtain the *minimum* or *maximum* of a funtional, but remember that stationary points are more general and can include things like saddle points and other points of inflection (i.e. points around which the second derivative changes sign).

In our case, we want to figure out *which path* $y(x)$ is the *shortest distance* between points $x = a$ and $x = b$. Translated to mathematical terms, we can say that we want to *optimize* $S(y, y', x)$ for the function $y(x)$ that minimizes S. But how do we do so? The answer requires a fair bit of explaining, so this is a section to be read through slowly.

The general functional optimization problem Consider a general functional $S(f, f', q)$ where the functional S is a function of $f(q)$, $f'(q)$, and q. Here, $f(q)$ is a parametric function of one parameter q - we will explore specific cases of $f(q)$ later (hint: one of these will be the *position* function $x(t)$ which is a parametric function where the parameter is t). Our functional $S(f, f', q)$ is given by:

$$S(f, f', q) = \int_{q_1}^{q_2} \mathcal{L}(f, f', q)\, dq \tag{531}$$

All of this is certainly very abstract, so let us examine what it all means. S is a **functional**, meaning that it takes some function $f(q)$ and outputs a number. The precise thing it *does*, in this case, is to integrate any *composite function* of f, its derivative $f'(q)$, and its input q, between two points in the domain of f. For notational clarity, we call this composite function of f, f', and q as $\mathcal{L}(f, f', q)$. As we are taking the integral of the composite function, this results in a number, since definite integrals return a number. So to sum it all up, S is a functional, that, given any function $f(q)$ - whatever the function may be - returns the definite integral of any possible composite function of f, its derivative f', and its input q.

We want to find the function f that minimizes or maximizes S. This means we want to find a function for which S does not change with respect to f (similar to how the derivative is zero at a critical point in normal calculus). To find this optimal function, let us vary S by adding a function $\eta(q)$ multiplied by a tiny number ε to f between q_1 and q_2 - this represents adding a tiny shift, also called a *variation*, to S. Our particular shift is such that $\eta(q_1) = \eta(q_2) = 0$, meaning that $\eta(q)$ vanishes at the endpoints, since we want this variation to only be between q_1 and q_2 (and nowhere outside of that range). We then have:

$$S(f + \varepsilon\eta, f' + \varepsilon\eta', q) = \int_{q_1}^{q_2} \mathcal{L}(f + \varepsilon\eta, f' + \varepsilon\eta', q)\, dq \tag{532}$$

Our next step is to find the *amount* of change δS between $S(f, f', q)$ and $S(f+\varepsilon\eta, f'+\varepsilon\eta', q)$. As a first step, we want to compute $\mathcal{L}(f+\varepsilon\eta, f'+\varepsilon\eta', q)$, as that will allow us to compute $S(f+\varepsilon\eta, f'+\varepsilon\eta', q)$, which we need in order to calculate δS.

> **Note**
> We use δS instead of dS or ∂S as (1) δS is specifically for functionals and (2) the latter two symbols already have (multiple) reserved uses and we don't want to muddle up the notation and make the meanings unclear.

Recall how, in single-variable calculus, we can express a small shift $y(x + h)$ in a function $y(x)$ for some tiny number h by:

$$y(x + h) = y(x) + y'(x)h \tag{533}$$

This idea can be generalized to multivariable calculus, where we can express a small shift in a function $f(x + \sigma, y + \lambda)$ in a function $f(x, y)$ by:

$$f(x + \sigma, y + \lambda) = f(x, y) + \frac{\partial f}{\partial x}\sigma + \frac{\partial f}{\partial y}\lambda \tag{534}$$

In a similar way, in the calculus of variations, we can write:

$$\mathcal{L}(f+\varepsilon\eta, f'+\varepsilon\eta', q) = \mathcal{L}(f, f', q) + \frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon + \frac{\partial \mathcal{L}}{\partial f'}\eta'\varepsilon \tag{535}$$

$$= \mathcal{L}(f, f', q) + \left(\frac{\partial \mathcal{L}}{\partial f}\eta + \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\right)\varepsilon \tag{536}$$

Now, by substitution, we can find $S(f+\varepsilon\eta, f'+\varepsilon\eta', q)$:

$$S(f+\varepsilon\eta, f'+\varepsilon\eta', q) = \int_{q_1}^{q_2} \mathcal{L}(f+\varepsilon\eta, f'+\varepsilon\eta', q)\,dq \tag{537}$$

$$= \int_{q_1}^{q_2} \left[\mathcal{L}(f, f', q) + \left(\frac{\partial \mathcal{L}}{\partial f}\eta + \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\right)\varepsilon\right] dq \tag{538}$$

We may find the *change in* S, which we will call δS, by subtracting $S(f, f', q)$ from the left-hand side:

$$\delta S = S(f+\varepsilon\eta, f'+\varepsilon\eta', q) - S(f, f', q) \tag{539}$$

$$= \int_{q_1}^{q_2} \left[\mathcal{L}(f, f', q) + \left(\frac{\partial \mathcal{L}}{\partial f}\eta + \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\right)\varepsilon\right] dq - \int_{q_1}^{q_2} \mathcal{L}(f, f', q)\,dq \tag{540}$$

$$= \int_{q_1}^{q_2} \left(\frac{\partial \mathcal{L}}{\partial f}\eta + \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\right)\varepsilon\,dt \tag{541}$$

In the limit as $\varepsilon \to 0$, we would expect that $\delta S = 0$, as the function that maximizes (or minimizes) S, again, is the function for which S **does not change with respect to** f. In formal language, this is called the process of *varying* S by a *variation* ε, and then demanding that $\lim_{\varepsilon\to 0} \delta S = 0$. This is why this form of calculus is called *the calculus of variations* or *variational calculus*. By setting $\delta S = 0$ we have:

$$\int_{q_1}^{q_2} \left(\frac{\partial \mathcal{L}}{\partial f}\eta + \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\right)\varepsilon\,dt = 0 \tag{542}$$

We would, however, prefer some way to get rid of the added function $\eta(q)$ to obtain an equation that doesn't depend on η. We can do this by explicitly performing the above integral. First, we split the sum into two parts for mathematical convenience for the following steps:

$$\int_{q_1}^{q_2} \left(\frac{\partial \mathcal{L}}{\partial f}\eta + \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\right)\varepsilon\,dt = \int_{q_1}^{q_2} \frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon\,dt + \int_{q_1}^{q_2} \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\varepsilon\;dt \tag{543}$$

We now simplify the second term in the integral by performing **integration by parts** to evaluate the integral. Recall that the integration by parts formula is as follows:

$$\int_a^b u dv = uv\Big|_a^b - \int_a^b v du \tag{544}$$

If we let $u = \dfrac{\partial \mathcal{L}}{\partial f'}\varepsilon$ and $dv = \dfrac{d\eta}{dq}$, then $v = \displaystyle\int \frac{d\eta}{dq}dq = \eta(q)$ and $du = \dfrac{d}{dq}\left(\dfrac{\partial \mathcal{L}}{\partial f'}\right)\varepsilon$. By substituting these in (we keep the first term there and don't evaluate, we only perform integration by parts on the second term) we have:

$$\underbrace{\int_{q_1}^{q_2} \frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon\,dt}_{\text{no need to evaluate}} + \underbrace{\int_{q_1}^{q_2} \frac{\partial \mathcal{L}}{\partial f'}\frac{d\eta}{dq}\varepsilon\;dt}_{\text{integrate by parts}} \tag{545}$$

$$= \int_{q_1}^{q_2} \frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon\,dt + \underbrace{\left[\frac{\partial \mathcal{L}}{\partial f'}\eta\,\varepsilon\Big|_{q_1}^{q_2} - \int_{q_1}^{q_2} \eta\frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right)\varepsilon\,dt\right]}_{\text{result of integration by parts}} \tag{546}$$

But recall from earlier that we defined $\eta(q)$ such that $\eta(q_2) = \eta(q_1) = 0$, meaning that the $\frac{\partial \mathcal{L}}{\partial f'}\eta\,\varepsilon\Big|_{q_1}^{q_2}$ term goes to zero. Therfore, we are only left with:

$$\int_{q_1}^{q_2} \frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon\,dt + \frac{\partial \mathcal{L}}{\partial f'}\eta\,\varepsilon\Big|_{q_1}^{q_2} - \int_{q_1}^{q_2} \eta\frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right)\varepsilon\,dt \tag{547}$$

$$= \int_{q_1}^{q_2} \frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon\,dt - \int_{q_1}^{q_2} \eta\frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right)\varepsilon\,dt \tag{548}$$

$$= \int_{q_1}^{q_2} \left[\frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon - \frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right)\eta\,\varepsilon\right] dt \tag{549}$$

$$\tag{550}$$

Where in the last term we re-joined the sum of the integrals (which will make the next steps much easier). We know that the integral quantity we derived in the last step must be equal to zero, given that $\delta S = 0$ is our fundamental requirement for finding the stationary points (minima, maxima, etc.) of functionals. Therefore we have:

$$\int_{q_1}^{q_2} \left[\frac{\partial \mathcal{L}}{\partial f}\eta\,\varepsilon - \frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right)\eta\,\varepsilon\right] dt \tag{551}$$

$$= \int_{q_1}^{q_2} \left[\frac{\partial \mathcal{L}}{\partial f} - \frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right)\right]\eta\,\varepsilon\;dt \tag{552}$$

$$= 0 \tag{553}$$

Where we factored the common terms out of the integral in the last step. But since our integral is zero, by the **fundamental lemma of the calculus of variations**, our integrand *must* be zero as well, and resultingly our quantity in the squared brackets must *also* be zero. That is:

$$\int_{q_1}^{q_2} \left[\frac{\partial \mathcal{L}}{\partial f} - \frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right)\right]\eta\,\varepsilon\;dt = 0 \quad \Rightarrow \quad \frac{\partial \mathcal{L}}{\partial f} - \frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right) = 0 \tag{554}$$

The last result is the general form of the **Euler-Lagrange equation** for our functional S. Since our functional is a very general functional, the Euler-Lagrange equation applies to a huge set of functionals - indeed, *all* functionals in the form $S[f(q), f'(q), q]$ (it is customary to use squared brackets when writing out the functional in its full form, but for our short form $S(f, f', q)$ it is permissible to simply use parentheses). Thus, it is an extremely crucial and useful equation, so let us write it down one more time:

$$\frac{\partial \mathcal{L}}{\partial f} - \frac{d}{dq}\left(\frac{\partial \mathcal{L}}{\partial f'}\right) = 0 \tag{555}$$

Application of the Euler-Lagrange equation to arc length Now, let us return to our example of the arc length functional:

$$S(y, y', x) = \int_a^b \sqrt{1 + y'^2}\,dx \tag{556}$$

We want to find $y(x)$ that minimizes this functional, and for this we can use the Euler-Lagrange equation. In this case, $f = y(x)$, $f' = y'$, and $\mathcal{L} = \sqrt{1 + y'^2}$, so the Euler-Lagrange equation for this particular functional reads:

$$\frac{\partial \mathcal{L}}{\partial y} - \frac{d}{dx}\left(\frac{\partial \mathcal{L}}{\partial y'}\right) = 0 \tag{557}$$

We may now compute the derivatives (which is much-simplified by the fact that $\mathcal{L} = \sqrt{1 + y'^2}$ *does not depend on* y, but we must be careful to remember that $\frac{d}{dx}f(y') = f'(y')y''$ due to the chain rule):

$$\frac{\partial \mathcal{L}}{\partial y} = 0 \tag{558}$$

$$\frac{\partial \mathcal{L}}{\partial y'} = \frac{1}{2}\frac{2y'}{\sqrt{1+y'^2}} = \frac{y'}{\sqrt{1+y'^2}} = y'(1+y'^2)^{-1/2} \tag{559}$$

$$\frac{d}{dx}\frac{\partial \mathcal{L}}{\partial y'^2} = y''(1+y'^2)^{-1/2} + \left(-\frac{1}{2}\right)(1+y'^2)^{-3/2}(2y')y'' \tag{560}$$

$$= y''(1+y'^2)^{-1/2} - y'(1+y'^2)^{-3/2}y'' \tag{561}$$

$$= y''[(1+y'^2)^{-1/2} - (1+y'^2)^{-3/2}] \tag{562}$$

Thus, substituting into the Euler-Lagrange equation, we have:

$$\frac{\partial \mathcal{L}}{\partial y} - \frac{d}{dx}\left(\frac{\partial \mathcal{L}}{\partial y'}\right) = -y''[(1+y'^2)^{-1/2} - (1+y'^2)^{-3/2}] \tag{563}$$

$$= y''[(1+y'^2)^{-3/2} - (1+y'^2)^{-1/2}] \tag{564}$$

$$= 0 \tag{565}$$

Note
The differentiation is indeed quite a bit tedious, and we could've used computer algebra systems to take the derivatives for us to speed up this process. We will cover computer algebra systems in depth in Chapter 2.

While this may look very complicated, remember that the quantity in the square brackets is zero:

$$y''[(1+y'^2)^{-3/2} - (1+y'^2)^{-1/2}] = 0 \tag{566}$$

$$\Rightarrow y''[(1+y'^2)^{-3/2} - (1+y'^2)^{-1/2}] = 0 \tag{567}$$

$$\Rightarrow y'' = 0 \tag{568}$$

This becomes a differential equation that is straightforward to integrate:

$$y'' = 0 \tag{569}$$

$$y' = \int y''dx = \int 0\,dx = b = \text{const.} \tag{570}$$

$$y = \int y'dx = \int b\,dx = mx + b \tag{571}$$

Where m, b are constants. This is simply an equation of a straight line! By applying the calculus of variations, we have therefore shown that the *shortest path between two points* a, b - in functional terms, the path that minimizes the arc length - is a **straight line**. It may seem to be an obvious result, but *proving it* required quite a bit of calculus!

Note that the one restriction we must place on this result is that we assume $S = \int \sqrt{1+y'^2}\,dx$ is the *right equation* for the arc length. For regular Euclidean space, this is always the right equation, and Euclidean space is what we'll work with 99% of the time. But in higher dimensions, and especially in non-Euclidean geometries, the arc length equation is no longer the correct equation for the arc length. We must then use differential geometry to construct the right equation for the arc length. But that is a topic we will cover in Chapter 3.

The Euler-Lagrange equation in physics In physics, we consider a specific case of the Euler-Lagrange equation, where (as mentioned at the beginning) $q = t$ is the time, $f = x(t)$ is the position, and $f' = \dot{x} = \dfrac{dx}{dt}$ is the velocity. Therefore, the Euler-Lagrange equation, in its common form used in physics (specifically, Lagrangian mechanics), becomes:

$$\frac{\partial \mathcal{L}}{\partial x} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} = 0 \tag{572}$$

Again, the Euler-Lagrange equation can be used to solve for the equations of motion as long as the Lagrangian is known. Note that for a more general set of coordinate systems, where the system is not one-dimensional motion along the x axis, there is an Euler-Lagrange equation that applies to each coordinate, each of which takes the following form:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = 0 \tag{573}$$

where q_i stands in for the particular coordinate, so q_i can be any one of x, y, z when working in Cartesian coordinates, or any one of r, θ, ϕ when working in spherical coordinates. And in the specific case when we are interested in solving for the motion of a system of objects, and not just of one individual object, it should be noted that the kinetic and potential energies are those of the **system** - that is, the sum of the kinetic and potential energies of every object in the system:

$$K = \sum_i K_i = K_1 + K_2 + K_3 + \cdots + K_n \tag{574}$$

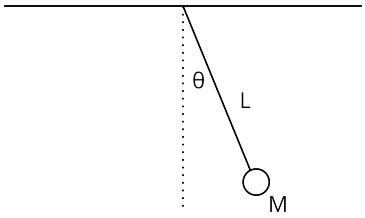
$$U = \sum_i U_i = U_1 + U_2 + U_3 + \cdots + U_n \tag{575}$$

Note that the Euler-Lagrange equations apply primarily to closed systems, i.e. systems with no external force acting on them. If there is an external applied force on the system that does work W, then the Euler-Lagrange equations become:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} - \frac{\partial \mathcal{L}}{\partial q_i} = \frac{\partial W}{\partial q_i} \tag{576}$$

Applying Lagrangian mechanics Having examined the fundamental theory behind Lagrangian mechanics, we will now look at a few examples of increasing difficulty, to illustrate its usefulness and mathematical elegance.

Using Lagrangian mechanics to solve the harmonic oscillator



For the single pendulum problem, we first find the equations $x(t)$ and $y(t)$ given our coordinate system. Our coordinate system is based on the point $(0, 0)$ located at the point where the pendulum is attached to the ceiling. Using basic trigonometry, we find that:

$$x(t) = \ell \sin\theta(t) \tag{577}$$

$$y(t) = -\ell \cos\theta(t) \tag{578}$$

Where $y(t)$ is negative because the pendulum is at a negative height relative to our origin. Using our expressions for $x(t)$ and $y(t)$, we want to find the expression for the kinetic energy K. We know that:

$$K = \frac{1}{2}mv^2 \tag{579}$$

And that:

$$v = \sqrt{v_x^2 + v_y^2} = \sqrt{\left(\frac{dx}{dt}\right)^2 + \left(\frac{dy}{dt}\right)^2} \tag{580}$$

To do this, we solve for $\frac{dx}{dt}$ and $\frac{dy}{dt}$. This takes a bit of care, because we need to implicitly differentiate $x(t)$ and $y(t)$ with respect to t, where:

$$\frac{dx}{dt} = \frac{dx}{d\theta}\frac{d\theta}{dt} \tag{581}$$

$$\frac{dy}{dt} = \frac{dy}{d\theta}\frac{d\theta}{dt} \tag{582}$$

Implicitly differentiating, we have:

$$v_x = \frac{dx}{dt} = \ell\cos\theta\frac{d\theta}{dt} \tag{583}$$

$$v_y = \frac{dy}{dt} = \ell\sin\theta\frac{d\theta}{dt} \tag{584}$$

Now, we plug our values for v_x and v_y into the kinetic energy equation, which gives us:

$$K = \frac{1}{2}m\left(\ell^2\cos^2\theta\frac{d\theta}{dt} + \ell^2\sin^2\theta\frac{d\theta}{dt}\right) \tag{585}$$

If we do a little simplification by factoring and remembering that $\cos^2\theta + \sin^2\theta = 1$, we get:

$$K = \frac{1}{2}m\ell^2\left(\frac{d\theta}{dt}\right)^2 \tag{586}$$

Now we find the potential energy. Remember that close to Earth, the potential energy is determined by and only by the vertical distance between the origin (which is the reference height of zero) and the measured point. This means that:

$$U = mgh \tag{587}$$

The height in this case is negative (because it's below the origin) and we're only taking the vertical component of the height (hence $\cos\theta$) so:

$$U = -mg\cos\theta\ell \tag{588}$$

Putting it all together, our Lagrangian is:

$$\mathcal{L} = \frac{1}{2}m\ell^2\left(\frac{d\theta}{dt}\right)^2 + mg\cos\theta\ell \tag{589}$$

Here, our coordinates are determined in terms of the angle θ only, so the Euler-Lagrange equations take the form:

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}} - \frac{\partial\mathcal{L}}{\partial\theta} = 0 \tag{590}$$

If we plug this into the Euler-Lagrange equation, we get:

$$m\frac{d^2\theta}{dt^2}\ell^2 + mg\sin\theta\ell = 0 \tag{591}$$

We can rewrite this as:

$$m\frac{d^2\theta}{dt^2}\ell^2 = -mg\sin\theta\ell \tag{592}$$

And cancel out the common factors to yield:

$$\frac{d^2\theta}{dt^2} = -\frac{g}{\ell}\sin\theta \tag{593}$$

We have arrived at our answer. This is the differential equation of the simple pendulum. Note that while this equation is impossible to solve analytically directly, we can use the small-angle approximation of $\sin\theta \approx \theta$ to get:

$$\frac{d^2\theta}{dt^2} = -\frac{g}{\ell}\theta \tag{594}$$

Of which the solution is:

$$\theta(t) = \theta_{max}\sin\left(\sqrt{\frac{g}{\ell}}t + \phi_0\right) \tag{595}$$

Using Lagrangian mechanics to solve the orbit equation We want to derive the orbit of Earth around the Sun. To do so, we again first derive the expressions for $x(t)$ and $y(t)$ in terms of the solar-earth system:

$$x(t) = r(t)\cos\theta(t) \tag{596}$$
$$y(t) = r(t)\sin\theta(t) \tag{597}$$

Differentiating both (and remembering to use the product rule), we find that:

$$v_x(t) = \cos\theta(t)\frac{dr}{dt} - r(t)\sin\theta(t)\frac{d\theta}{dt} \tag{598}$$
$$v_y(t) = \sin\theta(t)\frac{dr}{dt} + r(t)\cos\theta\frac{d\theta}{dt} \tag{599}$$

Which we can use to find the kinetic energy (after lots of algebra and several passes at using the identity $\sin^2\theta + \cos^2\theta = 1$):

$$K = \frac{1}{2}m\left[\left(\frac{dr}{dt}\right)^2 + r^2\left(\frac{d\theta}{dt}\right)^2\right] \tag{600}$$

The potential energy is given by $U = -\frac{GMm}{r}$, so the Lagrangian is:

$$\mathcal{L} = \frac{1}{2}m(\dot{r}^2 + r^2\dot{\theta}^2) + \frac{GMm}{r} \tag{601}$$

Applying the Euler-Lagrange equations to each coordinate, r and θ, present in the Lagrangian, we have:

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{r}} - \frac{\partial\mathcal{L}}{\partial r} = 0 \tag{602}$$
$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}} - \frac{\partial\mathcal{L}}{\partial\theta} = 0 \tag{603}$$

Solving both equations yields the equations of motion for the Earth, we find that:

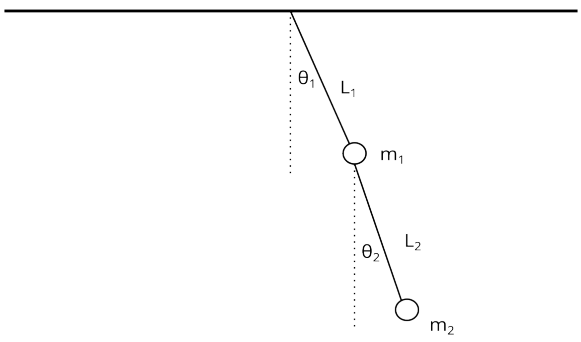
$$\frac{d^2r}{dt^2} = r\left(\frac{d\theta}{dt}\right)^2 - \frac{GM}{r^2} \tag{604}$$
$$= -\frac{2}{r}\frac{dr}{dt}\frac{d\theta}{dt} \tag{605}$$

These can be solved analytically, but for the sake of simplicity here they will be solved using a numerical differential equation solver:

```
import matplotlib.pyplot as plt
import numpy as np
from scipy.integrate import solve_ivp

def newtonian_d_dt(t, X, G=6.67e -11, M=2e30):
    r, theta, u, v = X
    dr_dt = u
    dtheta_dt = v
    du_dt = r * v ** 2 -(G * M) / (r ** 2)
    dv_dt = -(2 * u * v) / r
    return dr_dt, dtheta_dt, du_dt, dv_dt

r0 = 1.5e11
theta0 = np.pi/2
u0 = 0
v0 = 1.99e -7 # 2pi / 365 days
newtonian_initial = [r0, theta0, u0, v0]

tmax = 365 * 24 * 60 * 60 # 1 year
samples = 5000
t = np.linspace(0, tmax, samples)
newtonian = solve_ivp(newtonian_d_dt, (0, tmax), y0=newtonian_initial, dense_output=True)
sol = newtonian.sol(t)

fig = plt.figure()
ax = plt.axes()

r = sol[0]
theta = sol[1]

# Convert from polar to cartesian
x1 = r * np.cos(theta)
y1 = r * np.sin(theta)

ax.plot(x1, y1)
ax.set_title('Plot of Newtonian orbit')
plt.show()
```

As it can be seen, the orbit is a ellipse, and we have arrived at this result using Lagrangian mechanics!

Using Lagrangians to solve the double pendulum problem We will now tackle a problem that would be very difficult to solve using Newton's laws, but much easier with Lagrangian mechanics. Here we have a system as follows:



Here, the notable difference is that we have a **system** as opposed to a single object, and we need to find the kinetic and potential energies of the entire system. To do this, we divide the kinetic and potential energies into two parts:

$$K = K_1 + K_2 \tag{606}$$
$$U = U_1 + U_2 \tag{607}$$

Where K_1 and K_2 are respectively the kinetic energies of the first pendulum mass and second pendulum mass, and likewise with U_1 and U_2 and their potential energies.

We will first derive the kinetic energies, because they are harder :(As we know, we first setup a coordinate system where the point $(0, 0)$ is centered on the point the double pendulum is attached to the ceiling. Then, we write the position functions of the first pendulum:

$$x_1(t) = \ell_1 \sin\theta_1(t) \tag{608}$$
$$y_1(t) = -\ell_1 \cos\theta_1(t) \tag{609}$$

We figure these out from basic trigonometry and the fact that $y_1(t)$ is negative, as it is below the origin. We then take the derivatives to find the x and y components of the velocity:

$$\frac{dx_1}{dt} = \ell_1 \cos\theta_1(t)\frac{d\theta_1}{dt} \tag{610}$$
$$\frac{dy_1}{dt} = \ell_1 \sin\theta_1(t)\frac{d\theta_1}{dt} \tag{611}$$

Using this, we can find K_1:

$$K_1 = \frac{1}{2}m_1\left(\dot{x}_1^2 + \dot{y}_1^2\right) \tag{612}$$
$$= \frac{1}{2}m_1\left[\left(\ell_1 \cos\theta_1(t)\frac{d\theta_1}{dt}\right)^2 + \left(\ell_1 \sin\theta_1(t)\frac{d\theta_1}{dt}\right)^2\right] \tag{613}$$

Using the trig identity $\sin^2\theta + \cos^2\theta = 1$, this is trivial to simplify into:

$$K_1 = \frac{1}{2}m_1\ell_1^2\left(\frac{d\theta}{dt}\right)^2 \tag{614}$$

Then, we write the position functions of the second pendulum:

$$x_2(t) = x_1(t) + \ell_2 \sin\theta_2(t) \tag{615}$$
$$y_2(t) = y_1(t) + (-\ell_2 \cos\theta_2(t)) \tag{616}$$

Here, we add the x and y displacement of the second pendulum with the x and y displacement of the first to find the total displacement from the origin, because remember, we're using the *same* coordinate system for both pendulums. If we sub in the values of $x_1(t)$ and $y_1(t)$, we have:

$$x_2(t) = \ell_1 \sin\theta_1(t) + \ell_2 \sin\theta_2(t) \tag{617}$$
$$y_2(t) = -\ell_1 \cos\theta_1(t) - \ell_2 \cos\theta_2(t)) \tag{618}$$

We compute their derivatives:

$$\frac{dx_2}{dt} = \ell_1 \cos\theta_1(t)\frac{d\theta_1}{dt} + \ell_2 \cos\theta_2(t)\frac{d\theta_2}{dt} \tag{619}$$

$$\frac{dy_2}{dt} = \ell_1 \sin\theta_1(t)\frac{d\theta_1}{dt} + \ell_2 \sin\theta_2(t)\frac{d\theta_2}{dt} \tag{620}$$

And then plug them into the kinetic energy formula to find:

$$K_2 = \frac{1}{2}m_2\left[\left(\ell_1 \cos\theta_1(t)\frac{d\theta_1}{dt} + \ell_2 \cos\theta_2(t)\frac{d\theta_2}{dt}\right)^2 + \left(\ell_1 \sin\theta_1(t)\frac{d\theta_1}{dt} + \ell_2 \sin\theta_2(t)\frac{d\theta_2}{dt}\right)^2\right] \tag{621}$$

Expanding that out, and using $\sin^2\theta + \cos^2\theta = 1$ to simplify, we have:

$$K_2 = \frac{1}{2}m_2\left[\ell_1^2\left(\frac{d\theta_1}{dt}\right)^2 + 2\ell_1\ell_2\frac{d\theta_1}{dt}\frac{d\theta_2}{dt}(\cos\theta_1\cos\theta_2 + \sin\theta_2\sin\theta_2) + \ell_2^2\left(\frac{d\theta_2}{dt}\right)^2\right] \tag{622}$$

Here, we can use the identity $\cos x \cos y + \sin x \sin y = \cos(x - y)$ to simplify to:

$$K_2 = \frac{1}{2}m_2\left[\ell_1^2\left(\frac{d\theta_1}{dt}\right)^2 + 2\ell_1\ell_2\frac{d\theta_1}{dt}\frac{d\theta_2}{dt}\cos(\theta_1 - \theta_2) + \ell_2^2\left(\frac{d\theta_2}{dt}\right)^2\right] \tag{623}$$

Now, combining the two kinetic energies together, we end up with:

$$K = \frac{1}{2}m_1\ell_1^2\left(\frac{d\theta}{dt}\right)^2 + \frac{1}{2}m_2\left[\ell_1^2\left(\frac{d\theta_1}{dt}\right)^2 + 2\ell_1\ell_2\frac{d\theta_1}{dt}\frac{d\theta_2}{dt}\cos(\theta_1 - \theta_2) + \ell_2^2\left(\frac{d\theta_2}{dt}\right)^2\right] \tag{624}$$

We use a similar approach for the potential energies - we add the potential energy of the first pendulum and the second to find the total system's potential energy:

$$U = U_1 + U_2 \tag{625}$$

$$= m_1 g(-\ell_1 \cos\theta_1(t)) + m_2 g(-\ell_1 \cos\theta_1(t) - \ell_2 \cos\theta_2(t)) \tag{626}$$

After factoring, we have:

$$U = -(m_1 + m_2)g\ell_1 \cos\theta_1(t) - m_2 g\ell_2 \cos\theta_2(t) \tag{627}$$

Using $\mathcal{L} = K - U$, we substitute into the two Euler-Lagrange equations (one for θ_1 and one for θ_2):

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1} - \frac{\partial\mathcal{L}}{\partial\theta_2} = 0 \tag{628}$$

$$\frac{d}{dt}\frac{\partial\mathcal{L}}{\partial\dot{\theta}_1} - \frac{\partial\mathcal{L}}{\partial\theta_2} = 0 \tag{629}$$

With the Lagrangian:

$$\mathcal{L} = \frac{1}{2}m_1\ell_1^2\left(\frac{d\theta}{dt}\right)^2 + \frac{1}{2}m_2\left[\ell_1^2\left(\frac{d\theta_1}{dt}\right)^2\right. \tag{630}$$

$$\left. + 2\ell_1\ell_2\frac{d\theta_1}{dt}\frac{d\theta_2}{dt}\cos(\theta_1 - \theta_2) + \ell_2^2\left(\frac{d\theta_2}{dt}\right)^2\right] \tag{631}$$

$$+ (m_1 + m_2)g\ell_1 \cos\theta_1(t) + m_2 g\ell_2 \cos\theta_2(t) \tag{632}$$

To find the equations of motion, which are given by (source):

$$(m_1+m_2)l_1\ddot{\theta}_1 + m_2 l_2 \ddot{\theta}_2 \cos(\theta_1-\theta_2) + m_2 l_2 \dot{\theta}_2^2 \sin(\theta_1-\theta_2) + (m_1+m_2)g\sin\theta_1 = 0 \tag{633}$$

$$l_2\ddot{\theta}_2 + l_1\ddot{\theta}_1\cos(\theta_1-\theta_2) - l_1\dot{\theta}_1^2\sin(\theta_1-\theta_2) + g\sin\theta_2 = 0 \tag{634}$$

These equations are completely unsolvable analytically, but they can be solved numerically to yield the position of a double pendulum with time.

Langrangian to Newtonian mechanics Let's see how we can recover Newton's 2nd law from the Euler-Lagrange equation. Remember that the equation (in the case of one-dimenional motion along the x axis) is given by:

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{x}} = \frac{\partial \mathcal{L}}{\partial x} \tag{635}$$

And remember that the Lagrangian is given by:

$$\mathcal{L} = \frac{1}{2}m\dot{x}^2 - U(x) \tag{636}$$

Applying the Euler-Lagrange equations to the general Lagrangian, we get:

$$m\ddot{x} = -\frac{dU}{dx} \tag{637}$$

Recalling that $F = -\frac{dU}{dx}$ and $\ddot{x} = a$, we have recovered Newton's 2nd law!

$$F = ma \tag{638}$$

We can even use Lagrangian mechanics on simple problems and check that it matches with Newtonian mechanics. Let's do our freefall example from earlier. With $K = \frac{1}{2}m\dot{y}^2$ and $U = mgy$, we use the Euler-Lagrange equations to find:

$$m\frac{d^2y}{dt^2} = -mg \tag{639}$$

Which we can simplify to:

$$a_y = -g \tag{640}$$

Which reproduces the Newtonian result!

Hamiltonian mechanics

Noether's theorem

Field Lagrangians Besides working with particles and their trajectories, we are also often interested in *fields* in physics, such as the electromagnetic or gravitational fields. To find the differential equations that describe these fields, we need an Euler-Lagrange equation for *fields* rather than particles.

Let us consider a generic field $\varphi(\mathbf{r}, t)$. For reasons that will be elaborated in more detailed in the special and general relativity sections (we will give a rough outline for why in a note further down this section), it is conventional to group the space and time components in one vector $\mathbf{X}$ which has four components, one of time and three of space, and where c is the speed of light:

$$\mathbf{X} = (ct, \mathbf{r}) = \begin{bmatrix} ct \\ x \\ y \\ z \end{bmatrix} \tag{641}$$

It is also convention to group the time derivative and gradient operators together, which we call the **four-gradient** and denote $\partial_{\mathbf{X}}$:

$$\partial_{\mathbf{X}} = \left(-\frac{1}{c}, \nabla\right) = \left\langle -\frac{1}{c}, \frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right\rangle \tag{642}$$

Note for the advanced reader
In special and general relativity, we call $\mathbf{X}$ the **four-position** and we write it in tensor notation as x^μ, and we call $\partial_{\mathbf{X}}$ the **four-gradient** and we write it similarly in tensor notation as ∂_μ. The reason we need these four-component vectors (and vector derivatives) is that in the theories of relativity, four-dimensional *spacetime* becomes of fundamental importance whereas space and time are reduced to perspective-dependent aspects of spacetime.

We will cover more on tensors in this chapter, but knowing tensors is *not* required for our (rough) treatment of Lagrangians in field theory. We will proceed with simply the four-position and four-gradient written in vector form.

With these new vectors we may write the field as $\varphi(\mathbf{X})$. The action for the field is given by:

$$S[\varphi, \partial_{\mathbf{X}}\varphi, \mathbf{X}] = \int_\Omega \mathcal{L}(\varphi, \partial_{\mathbf{X}}\varphi, \mathbf{X}) d^4x \tag{643}$$

Where $\mathcal{L}$ is our *field Lagrangian*, $d^4x = dV\,dt$ is an infinitesimal portion of space and time, and Ω is the domain of all space and all times. We will not repeat the *full* derivation of the Euler-Lagrange equation, but the steps are very similar to what we have already seen with the single-object Lagrangian case. The result is the **Euler-Lagrange equation for fields**, which takes the form:

$$\frac{\partial \mathcal{L}}{\partial \varphi} - \frac{\partial}{\partial \mathbf{X}}\left(\frac{\partial \mathcal{L}}{\partial(\partial_{\mathbf{X}}\varphi)}\right) = 0 \tag{644}$$

Note
This section is *optional* and discusses advanced physics, so by all means read on if interested, but otherwise feel free to skip this part!

Lagrangians to the stars The Lagrangian formulation of classical mechanics is so powerful, precisely because it relies on a differential equation that can be generalized. Beyond classical mechanics, the Lagrangian isn't always necessarily $\mathcal{L} = K - U$, but the Euler-Lagrange equations still hold true, and so does the principle of stationary action. Thus, a theory - including those that involve fields - can be written *as a Lagrangian*, as the Euler-Lagrange equations yield the equations of motion for each theory, on which the rest of the theory is built on! *This* is the reason behind learning Lagrangian mechanics.

We will end with one final thought - one of the most successful theories in all of physics, the **Standard model** of particle physics (which is a *quantum field theory*), is encapsulated in one compact Lagrangian:

$$\mathcal{L}_{\text{SM}} = -\frac{1}{4}F_{\mu\nu}F^{\mu\nu} + i\overline{\psi}\gamma_\beta D^\beta + h.c. + \psi_i y_{ij}\psi_j\phi + h.c. + |D_\mu\phi|^2 - V(\phi) \tag{645}$$

And one of the most mathematically beautiful theories, in fact one we will see very soon, **General Relativity** (which is a *classical field theory* for gravity), is described in another compact Lagrangian:

$$\mathcal{L}_{\text{GR}} = \frac{1}{2\kappa}R\sqrt{-g} + \mathcal{L}_{\text{matter}} \tag{646}$$

Who knew that Lagrangians could take us deep into the hearts of atoms, and to the furthest stars...? :)

Electrodynamics

> "We can scarcely avoid the inference that light consists in the transverse undulations of the same medium which is the cause of electric and magnetic phenomena."
> **James Clerk Maxwell**

In the first chapter, we introduced the classical theory of electricity and magnetism. We discussed electric and magnetic fields, and we ended with a discussion on Maxwell's equations and electrodynamics. Now, we will take a closer look at electrodynamics - and gain critical insights into the nature of light and the electromagnetic field.

Formulations of the Maxwell equations We will start by reviewing the Maxwell's equations. Written in the language of vector calculus, they take the form of a system of four partial differential equations, which we show below:

$$\nabla \cdot \mathbf{E} = \frac{1}{\epsilon_0}\rho \tag{647}$$

$$\nabla \cdot \mathbf{B} = 0 \tag{648}$$

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{649}$$

$$\nabla \times \mathbf{B} = \mu_0 \mathbf{J} + \frac{1}{c^2}\frac{\partial \mathbf{E}}{\partial t} \tag{650}$$

This form is excellent for conceptual understanding and for illustrating the relationships between the electric and magnetic fields. However, it is rather poorly-suited for actually finding *solutions*. First, there are *four* PDEs to solve, and second, the last two PDEs are *coupled*, meaning that they cannot be solved independently of each other.

But we can simplify the Maxwell equations into a more tractable form. First, we can reduce the number of equations to solve, because, technically, the first two equations are just constraint equations. That is to say, these two equations don't need to be solved because they are *implicitly satisfied* for any solution of Maxwell equation #3 and Maxwell equation #4, a result first derived by J. A. Stratton in 1941. To show this, let's first take the divergence of the time derivative of $\mathbf{B}$. We get:

$$\nabla \cdot \left(\frac{\partial \mathbf{B}}{\partial t}\right) = \nabla \cdot (-\nabla \times \mathbf{E}) = 0 \tag{651}$$

Where we substitute in the left-hand side of Maxwell equation #3 and use the vector calculus identity that the divergence of the curl is always zero (the vector calculus identities Wikipedia page is a highly-useful resource when doing a lot of vector calculus). But remember, taking the divergence only affects the spatial components of fields (because $\nabla \cdot \mathbf{F} = \partial_x \mathbf{F}_x + \partial_y \mathbf{F}_y + \partial_z \mathbf{F}_z$, there is no time-dependent part in it) so we can swap the time derivative operator and the divergence operator:

$$\nabla \cdot \left(\frac{\partial \mathbf{B}}{\partial t}\right) = \frac{\partial}{\partial t}(\nabla \cdot \mathbf{B}) \tag{652}$$

However, we already found that $\nabla \cdot (\partial_t \mathbf{B}) = 0$, so:

$$\frac{\partial}{\partial t}(\nabla \cdot \mathbf{B}) = 0 \tag{653}$$

This means if the initial condition for the magnetic field obeys $\nabla \cdot \mathbf{B} = 0$, then $\nabla \cdot \mathbf{B} = 0$ holds for all times t, and therefore, Gauss's law for magnetism is **automatically satisfied** for any solution of just the last two of Maxwell's equations.

Meanwhile, we can rearrange Maxwell equation #4 to have the time-derivative on the left via:

$$\frac{\partial \mathbf{E}}{\partial t} = c^2(\nabla \times \mathbf{B} - \mu_0 \mathbf{J}) \tag{654}$$

Which, knowing that $c^2 = \frac{1}{\mu_0 \varepsilon_0}$, we can simplify to:

$$\frac{\partial \mathbf{E}}{\partial t} = \left(\frac{1}{\mu_0 \varepsilon_0} \nabla \times \mathbf{B} - \frac{1}{\varepsilon_0} \mathbf{J} \right) \tag{655}$$

Taking the divergence of both sides, we get:

$$\nabla \cdot \left(\frac{\partial \mathbf{E}}{\partial t} \right) = \nabla \cdot \left(\frac{1}{\mu_0 \varepsilon_0} \nabla \times \mathbf{B} - \frac{1}{\varepsilon_0} \mathbf{J} \right) = -\frac{1}{\varepsilon_0} \nabla \cdot \mathbf{J} \tag{656}$$

Again, due to the fact that taking the divergence doesn't include taking any time derivatives, we can swap out the time derivative and divergence operators, that is:

$$\nabla \cdot \left(\frac{\partial \mathbf{E}}{\partial t} \right) = \frac{\partial}{\partial t} (\nabla \cdot \mathbf{E}) \tag{657}$$

So substituting that in, we have:

$$\frac{\partial}{\partial t} (\nabla \cdot \mathbf{E}) = -\frac{1}{\varepsilon_0} \nabla \cdot \mathbf{J} \tag{658}$$

But due to charge conservation, the current density $\mathbf{J}$ and charge density ρ are related by the partial differential equation:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0 \tag{659}$$

This is the **continuity equation** for charge conservation and it means that charge can never be created or destroyed, just moved around. If we rearrange it, we get:

$$\nabla \cdot \mathbf{J} = -\frac{\partial \rho}{\partial t} \tag{660}$$

If we substitute this into $\partial_t (\nabla \cdot \mathbf{E}) = -1/\varepsilon_0 \nabla \cdot \mathbf{J}$ which we derived two steps before, we get:

$$\frac{\partial}{\partial t} (\nabla \cdot \mathbf{E}) = -\frac{1}{\varepsilon_0} \nabla \cdot \mathbf{J} = \frac{1}{\varepsilon_0} \frac{\partial \rho}{\partial t} \tag{661}$$

So that we have the relation:

$$\frac{\partial}{\partial t} (\nabla \cdot \mathbf{E}) = \frac{1}{\varepsilon_0} \frac{\partial \rho}{\partial t} \tag{662}$$

Rearranging the terms and doing some straightforward rewriting, we get:

$$\frac{\partial}{\partial t} (\nabla \cdot \mathbf{E}) - \frac{1}{\varepsilon_0} \frac{\partial \rho}{\partial t} = 0 \Rightarrow \frac{\partial}{\partial t} \left(\nabla \cdot \mathbf{E} - \frac{\rho}{\varepsilon_0} \right) = 0 \tag{663}$$

But we know that $\nabla \cdot \mathbf{E} - \rho/\varepsilon_0$ is just a different way of writing Maxwell equation #1, which is Gauss's law for electricity! So if the initial condition for the electric field obeys $\nabla \cdot \mathbf{E} - \rho/\varepsilon_0 = 0$, then $\nabla \cdot \mathbf{E} - \rho/\varepsilon_0 = 0$ holds for all times t, and therefore, Gauss's law for electricity, just like Gauss's law for magnetism that we looked at earlier, is also **automatically satisfied** for any solution of just the last two of Maxwell's equations. To our relief, this means that so long as we check our initial conditions to ensure that the divergence of both the electric and magnetic fields is zero, Maxwell's equations reduce to just two coupled PDEs!

$$\frac{\partial \mathbf{E}}{\partial t} = c^2 (\nabla \times \mathbf{B} - \mu_0 \mathbf{J}) \tag{664}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{665}$$

In free space, that is, when we're not considering any current sources, we can set $\mathbf{J} = \rho = 0$, meaning that $\nabla \cdot \mathbf{E} = 0$ as well as $\nabla \cdot \mathbf{B} = 0$. Maxwell's equations then become:

$$\frac{\partial \mathbf{E}}{\partial t} = c^2 \nabla \times \mathbf{B} \tag{666}$$
$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \tag{667}$$

If we take the curl of both sides of each, we find that:

$$\nabla \times \left(\frac{\partial \mathbf{E}}{\partial t} \right) = \nabla \times c^2 \nabla \times \mathbf{B} \tag{668}$$
$$\nabla \times \left(\frac{\partial \mathbf{B}}{\partial t} \right) = -\nabla \times \mathbf{E} \tag{669}$$

Using the double-curl vector calculus identity $\nabla \times (\nabla \times \mathbf{A}) = \nabla(\nabla \cdot \mathbf{A}) - \nabla^2 \mathbf{A}$, and swapping the time derivative and curl operators (remember that like the divergence, the curl is spatial-only, doesn't have any effect on time), we have:

$$\frac{\partial}{\partial t}(\nabla \times \mathbf{E}) = -c^2 \nabla^2 \mathbf{B} \tag{670}$$
$$\frac{\partial}{\partial t}(\nabla \times \mathbf{B}) = \nabla^2 \mathbf{E} \tag{671}$$

But we know what $\nabla \times \mathbf{E}$ and $\nabla \times \mathbf{B}$ are from Maxwell equations #3 and #4, and we can substitute those in:

$$\frac{\partial}{\partial t}(\nabla \times \mathbf{E}) = \frac{\partial}{\partial t}\left(-\frac{\partial \mathbf{B}}{\partial t} \right) = -\frac{\partial^2 \mathbf{B}}{\partial t^2} \tag{672}$$
$$\frac{\partial}{\partial t}(\nabla \times \mathbf{B}) = \frac{\partial}{\partial t}\left(\frac{1}{c^2} \frac{\partial \mathbf{E}}{\partial t} \right) = \frac{1}{c^2} \frac{\partial^2 \mathbf{E}}{\partial t^2} \tag{673}$$

Equating these expressions with the expressions we got by applying the double-curl identity, we get the wave equations that we analyzed earlier:

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} = c^2 \nabla^2 \mathbf{E} \tag{674}$$
$$\frac{\partial^2 \mathbf{B}}{\partial t^2} = c^2 \nabla^2 \mathbf{B} \tag{675}$$
$$\tag{676}$$

So we can summarize the results as follows:

Configuration	System of equations	Interpretation
Free space (no current/charge sources)	$\frac{\partial^2 \mathbf{E}}{\partial t^2} = c^2 \nabla^2 \mathbf{E}$, $\frac{\partial^2 \mathbf{B}}{\partial t^2} = c^2 \nabla^2 \mathbf{B}$	EM waves
General case (can have current/charge sources)	$\frac{\partial \mathbf{E}}{\partial t} = c^2(\nabla \times \mathbf{B} - \mu_0 \mathbf{J})$, $\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$	Any EM field configuration

But again, the PDEs themselves are not enough. For a complete boundary-value problem, we also need the boundary conditions, and those come from the physics we want to model. We will now look at different boundary-value problems in which we can solve the Maxwell equations analytically.

Solutions to the electromagnetic wave equation The solutions to the first case - the Maxwell equations in free space - are of particular interest, as their solutions describe *electromagnetic waves*. The first, and simplest, solution that we will examine is that of the plane wave, given by:

$$\mathbf{E}(t, x, y, z) = \mathbf{E}_0 \cos(k_x x + k_y y + k_z z - kct + \phi) \tag{677}$$

Which is sometimes written in more compact notation as:

$$\mathbf{E}(t, \mathbf{r}) = \mathbf{E}_0 \cos k(\hat{\mathbf{n}} \cdot \mathbf{r} - ct + \phi) \tag{678}$$

Or written in equivalent complex exponential form (recall: $e^{i\phi} = \cos\phi + i\sin\phi$), omitting the phase shift for now since we can always include it back in later:

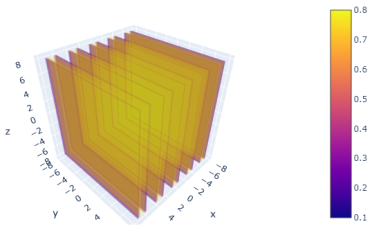
$$\mathbf{E}(t, \mathbf{r}) = \mathbf{E}_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r} - ct)} \tag{679}$$

The magnitude (electric field strength) is then given by:

$$E(t, \mathbf{r}) = \|\mathbf{E}\| = E_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r} - ct)} \tag{680}$$

Where $\hat{\mathbf{n}}$ is the normal vector that points in the direction of propagation of the wave. Of course, complex-valued waves are just for mathematical convenience; to get any physically meaningful results, we only consider the real part of the complex exponential, and ignore the imaginary part.

Recall that the output of the function is the *magnitude* of the wave, and in electrodynamics it is associated with the magnitude of the electromagnetic field at that point. In 1 dimension, this is the familiar sinusoidal wave; in 2 dimensions, this becomes a sinusoidally-rippling surface, see https://www.math3d.org/JdgRCZD3l. In 3 dimensions, it is a bit more difficult to visualize, but here is a volume (density) plot to serve as a visual:



The magnitude of the electric field for a plane wave, which oscillates sinusoidally along the direction of propagation.

The intensity still varies sinusoidally, but the wavefronts are planar, as is expected. Note that plane waves are a mathematical idealization; true waves are only approximately planar and would spread out over long enough distances.

We now want to find the associated $\mathbf{B}$ field from the electric field. To do so recall the third of Maxwell's equations:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{681}$$

Taking the curl of $\mathbf{E}$ is highly nontrivial; however, luckily, we have a very nice identity that $\nabla \times \mathbf{A} = \hat{\mathbf{A}} \times \nabla A$ where $A = \|\mathbf{A}\|$, the proof is in the appendix and is very involved and was partially the result of countless hours on board an airplane spent doing math because I am a weird person. Be aware that $\hat{\mathbf{E}} \neq \hat{\mathbf{n}}$, the unit vector of the electric field is **not** the same thing as the normal vector in the argument of the complex exponential! The electric field has $\hat{\mathbf{E}} = \mathbf{E}/E = \mathbf{E}_0/E_0$, remember the unit vector doesn't care about magnitude, only direction, so these two are equivalent. The $\hat{\mathbf{n}}$ in the argument of the complex exponential is the direction of *propagation* of the wave, but $\hat{\mathbf{E}}$ is the

unit vector, i.e. the direction that the *electric field is align.* These are not the same thing because electromagnetic fields are **transverse**; the electric field oscillates up and down, but the wave moves forward.

With that distinction out of the way, we can resume the calculations. The magnitude of $\mathbf{E}$ is:

$$E = \|\mathbf{E}\| = E_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} \tag{682}$$

The gradient, ∇E, becomes:

$$\nabla E = ik\hat{\mathbf{n}}E_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} \tag{683}$$

Which can be found by simply taking the gradient component-by-component, and then rearranging them into a vector. Applying the previously-mentioned identity $\nabla \times \mathbf{A} = \hat{\mathbf{A}} \times \nabla A$ results in:

$$\begin{aligned}
\nabla \times \mathbf{E} &= \frac{\mathbf{E}_0}{E_0} \times ik\hat{\mathbf{n}}E_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} && (684)\\
&= \mathbf{E}_0 \times ik\hat{\mathbf{n}}E_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} && (685)\\
&= -ik\hat{\mathbf{n}} \times E_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} && (686)\\
&= -ik\hat{\mathbf{n}} \times \mathbf{E} && (687)
\end{aligned}$$

Where we used the property of cross products that $\mathbf{A} \times \mathbf{B} = -\mathbf{B} \times \mathbf{A}$, and the fact that the cross product is linear so we can simply pull out the constant factors.

Solving for Maxwell #3 and plugging in the previously-obtained expression, we have:

$$\nabla \times \mathbf{E} = -ik\hat{\mathbf{n}} \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t} \tag{688}$$

Which we simply integrate with respect to time to obtain the magnetic field:

$$\begin{aligned}
\mathbf{B} &= -\int -ik\hat{\mathbf{n}} \times \mathbf{E}\, dt && (689)\\
&= ik \int \hat{\mathbf{n}} \times \mathbf{E}_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} && (690)\\
&= \frac{ik}{ikc}\hat{\mathbf{n}} \times \mathbf{E}_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} && (691)\\
&= \frac{1}{c}\hat{\mathbf{n}} \times \mathbf{E} && (692)
\end{aligned}$$

Or written explicitly, the magnetic field is given by:

$$\mathbf{B} = \mathbf{B}_0 e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} = \hat{\mathbf{n}} \times \frac{\mathbf{E}_0}{c} e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} \tag{693}$$

The fact that:

$$\mathbf{B} = \frac{1}{c}\hat{\mathbf{n}} \times \mathbf{E} \tag{694}$$

means that the magnetic fields and electric fields are perpendicular to each other. Here's a visual to make things more clear:

We should mention that this diagram is not entirely physically accurate; in reality, the electric and magnetic fields span every point in space, so imagine every point filled with electric and magnetic vectors. In addition, the plane waves span the entire domain too, here only slices (isosurfaces is the technical term) are shown. Also, plane waves can propagate in any direction, not just forwards along the x axis, this is a simplified example. However, it is a diagram that is a good approximation for the physical reality.

Electromagnetic waves, like all waves, deliver power and energy from one location to another. Electromagnetic power is given by the Poynting vector:

Electric and magnetic fields of a plane wave. Notice how the wavefronts are flat, and $\mathbf{E}$ and $\mathbf{B}$ are perpendicular to each other.

$$\mathbf{S} = \frac{1}{\mu_0}\mathbf{E} \times \mathbf{B} \tag{695}$$

The magnitude of the Poynting vector is the **power flux density**: the power passing through each unit area in space. In our case, since $\mathbf{B} = \frac{1}{c}\hat{\mathbf{n}} \times \mathbf{E}$, if we take E to be the magnitude of the electric field, then $S = \frac{1}{\mu_0 c}E^2$ (why? because the electric and magnetic fields are mutually orthogonal and orthogonal to the normal vector so the cross product is one). To get the actual power, we have to integrate over all considered surface areas:

$$P = \iint \mathbf{S} \cdot d\mathbf{A} = \frac{1}{\mu_0 c} \iint E^2 \, dA \tag{696}$$

In the case that the surface areas are uniform, we have:

$$P = E_0^2 A e^{2ik(\hat{\mathbf{n}}\cdot\mathbf{r} - ct)} \tag{697}$$

Taking only the real part and using the trigonometric identity $\cos^2\theta = \frac{1}{2}(1 + \cos 2\theta)$, we have:

$$P = E_0^2 A \cos^2 k(\hat{\mathbf{n}} \cdot \mathbf{r} - ct) \tag{698}$$

As a side note, if considering surfaces of constant area, then the intensity would correspondingly be:

$$I = \frac{P}{A} = E_0^2 \cos^2 k(\hat{\mathbf{n}} \cdot \mathbf{r} - ct) \tag{699}$$

Further wave solutions Plane waves are not physically possible to realize. They are only a good approximation for electromagnetic waves that have dispersed far enough that they *appear* to be parallel over small enough cross-sections, and thus the electric and magnetic fields are well-approximated with flat, perfectly constant wavefronts. For example, sunlight can be modeled using plane waves, because the waves of light have spread out sufficiently far by the time they reach earth and because we view a tiny cross-section of those waves (they are a solar system radius spread out!) that they look like plane waves. However, real waves falloff by the inverse of distance, where $r = \|\mathbf{r}\|$:

$$\mathbf{E}(t, \mathbf{r}) = \frac{E_0}{r} e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r} - ct)} \tag{700}$$

This ensures that assuming surfaces of uniform area, the measured power through each area is given by:

$$P = \frac{E_0^2 A}{r^2} \cos^2 k(\hat{\mathbf{n}} \cdot \mathbf{r} - ct) \tag{701}$$

Which is an inverse-square law, required to guarantee energy conservation. Spherical waves are often *also* called plane waves, due to the fact that other than their spherical propagation, they are essentially *the same* as plane waves. This is because they are monochromatic (have a fixed wavelength, and thus fixed wavenumber) and their wavefronts are flat and uniform.

But we will now examine a solution that is neither monochromatic nor has flat wavefronts. To start, recall that the linearly of the electromagnetic wave equation means that *any superposition* of

plane waves will *also* be a solution. This means that we can sum any number of plane waves together to form a new wave. In the limit of infinitely-many plane waves added together, we obtain an integral:

$$\mathbf{E}(\mathbf{r},t) = \int_{-\infty}^{\infty} \frac{dk}{2\pi} A(k) e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} \tag{702}$$

where $A(k)$ is the *amplitude* of each component plane wave, and dividing by 2π is just a way of normalizing the plane wave (and we technically don't need it). But this is just in the 1-dimensional case, when we consider only one component of the wavevector. For the full 3-dimensional case, the integral becomes:

$$\mathbf{E}(\mathbf{r},t) = \iiint_{-\infty}^{\infty} \frac{d^3k}{(2\pi)^3} A(k) e^{ik(\hat{\mathbf{n}}\cdot\mathbf{r}-ct)} \tag{703}$$

The integral over an infinite number of waves of differing wavelengths means that the resulting wave can take very different shapes as compared to the component plane waves. In addition, by integrating over individual plane waves, destructive and constructive interference effects (that is, where waves added together amplify each other in some regions and cancel out in others) means that the wave no longer needs infinite energy to be created and obeys energy conservation.

The Helmholtz equation Up to this point, we have worked with solutions to the electromagnetic wave equation that were variations of plane waves. But for analyzing more complicated problems in electromagnetics, we need to look beyond plane waves.

For this, we need to directly work with the electromagnetic wave equation:

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} = c^2 \nabla^2 \mathbf{E} \tag{704}$$

To solve it, we assume that the electric field $\mathbf{E}(\mathbf{r},t)$ can be written as a product of two functions. We will call the first function $\mathbf{E}_s(\mathbf{r})$, where s denotes "spatial" (for reasons that will become apparent later), and the second function $\mathcal{E}(t)$. Then we have:

$$\mathbf{E}(\mathbf{r},t) = \mathbf{E}_s(\mathbf{r})\mathcal{E}(t) \tag{705}$$

We may now find the Laplacian and second time derivative of the electric field explicitly:

$$\nabla^2 \mathbf{E} = \nabla^2(\mathbf{E}_s(\mathbf{r})\mathcal{E}(t)) = \mathcal{E}\nabla^2 \mathbf{E}_s \tag{706}$$

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} = \frac{\partial^2}{\partial t^2}(\mathbf{E}_s(\mathbf{r})\mathcal{E}(t)) = \mathbf{E}_s \frac{\partial^2 \mathcal{E}}{\partial t^2} \tag{707}$$

> **Note**
> The reason why these derivatives turned out to have such simple forms, despite the fact that we normally need to use the verbose product rule when differentiating products of functions, is because $\mathbf{E}_s(\mathbf{r})$ is dependent only on space and $\mathcal{E}(t)$ is dependent only on time. Since the Laplacian is *also* dependent only on space and the 2nd time (partial) derivative is *also* dependent only on time, the components independent of space (for the first) and independent of time (for the second) can be treated as constants on differentiation, leading to these simple expressions.

By substituting the computed Laplacian and second time derivative into the electromagnetic wave equation, we have:

$$\mathbf{E}_s \frac{\partial^2 \mathcal{E}}{\partial t^2} = c^2 \mathcal{E} \nabla^2 \mathbf{E}_s \tag{708}$$

If we then divide both sides by $\mathbf{E}_s(\mathbf{r})\mathcal{E}(t)$ (which is algebraically sound), we have:

$$\frac{1}{\mathbf{E}_s(\mathbf{r})\mathcal{E}(t)}\mathbf{E}_s\frac{\partial^2\mathcal{E}}{\partial t^2} = \frac{1}{\mathbf{E}_s(\mathbf{r})\mathcal{E}(t)}c^2\mathcal{E}\nabla^2\mathbf{E}_s \tag{709}$$

$$\frac{1}{\mathcal{E}(t)}\frac{\partial^2\mathcal{E}}{\partial t^2} = \frac{1}{\mathbf{E}_s(\mathbf{r})}c^2\nabla^2\mathbf{E}_s \tag{710}$$

It is also common (as a matter of convention) to divide both sides by c^2, such that we have:

$$\frac{1}{c^2\mathcal{E}(t)}\frac{\partial^2\mathcal{E}}{\partial t^2} = \frac{1}{\mathbf{E}_s(\mathbf{r})}\nabla^2\mathbf{E}_s \tag{711}$$

Since each side is now composed of functions and derivatives of only *one* variable (t and $\mathbf{r}$ respectively), we say we have *separated* the variables. And because we have two combinations of derivatives with respect to different variables, they must be equal to a constant (or some multiple or square or cube of a constant). As a matter of fact, we can choose *any* constant formed by any operation on another constant, it just matters that it preserves the fact that it is *constant*. We will call this constant $-k^2$ (this is, in fact, the square of the *wavenumber*, the scalar magnitude of the wavevector, which can be shown via dimensional analysis):

$$\frac{1}{c^2\mathcal{E}(t)}\frac{\partial^2\mathcal{E}}{\partial t^2} = \frac{1}{\mathbf{E}_s(\mathbf{r})}\nabla^2\mathbf{E}_s = -k^2 \tag{712}$$

By just looking at the second equation, we have:

$$\frac{1}{\mathbf{E}_s(\mathbf{r})}\nabla^2\mathbf{E}_s = -k^2 \tag{713}$$

Which we can rearrange as:

$$\nabla^2\mathbf{E}_s + k^2\mathbf{E}_s = 0 \tag{714}$$

This is the **Helmholtz equation**, the time-independent form of the electromagnetic wave equation. The subscripts can often be dropped to write it in the more simple form $(\nabla^2 + k^2)\mathbf{E} = 0$. By separating the time and space components, we can much more readily solve complicated boundary-value problems and only need to solve for spatial boundary conditions

Gaussian beams A Gaussian beam is a particular solution to the Helmholtz equation, given by:

$$\mathbf{E}(r,z) = E_0\,\hat{\mathbf{x}}\,\frac{w_0}{w(z)}\exp\left(\frac{-r^2}{w(z)^2}\right)\exp\left(-i\left(kz + k\frac{r^2}{2R(z)} - \psi(z)\right)\right) \tag{715}$$

Where r is the radial coordinate along the cross-section of the beam, z is the coordinate along the axis of propagation, w_0 is the beam diameter at $z = 0$, k is the magnitude of the wavevector, and $w(z)$, $R(z)$ and $\psi(z)$ are functions we will elaborate on shortly. The Gaussian beam models highly-focused and highly-directional electromagnetic waves, such as laser beams or the waves produced by an idealized parabolic antenna - understandably, it is highly-relevant to antenna and laser physics as well as optics.

One of the most important characteristics of a Gaussian beam is its **beam diameter**. The function $w(z)$ describes how the beam loses focus and spreads out with distance. $w(z)$ gives the general beam diameter for any given value of z, given by:

$$w(z) = w_0\sqrt{1 + \left(\frac{z}{z_R}\right)^2}, \quad z_R = \frac{\pi {w_0}^2 n}{\lambda} = \frac{1}{2}k{w_0}^2 \tag{716}$$

Where n is the refractive index of the medium, and $n \geq 1$, with $n = 1$ in vacuum. The function $\psi(z)$ is called the **Gouy phase**, and is given by:

$$\psi(z) = \arctan\left(\frac{z}{z_R}\right) \tag{717}$$

Meanwhile, the function $R(z)$ is the **radius of curvature** of the beam, which describes the curvature of the wavefront, and is given by:

$$R(z) = z\left[1 + \left(\frac{z_R}{z}\right)^2\right] \tag{718}$$

The magnitude of the Poynting vector (power flux density) of the Gaussian beam is:

$$S = \frac{{E_0}^2}{2\eta}\left(\frac{w_0}{w(z)}\right)^2 \exp\left(\frac{-2r^2}{w(z)^2}\right) \tag{719}$$

Where η is the impedance (a form of electric resistance), and can be approximated via $\eta = 377\Omega$ which is the impedance of free space. That is to say, the power falls off with an exponential decay, but the decay is also proportional to the electric field strength and inversely proportional to $w(z)$. Allowing $w(z)$ to increase slowly and using a powerful electric field would be able to minimize the decay.

Conservation laws In physics, every theory has its conservation laws, and electrodynamics is no different. There are two important conservation laws within electrodynamics. The first is the **global conservation of charge**:

$$\frac{d}{dt}Q_{\text{total}} = \frac{d}{dt}\iiint \rho\, dV = 0 \tag{720}$$

The conservation of charge requires that the total charge across all of space stay constant. Charges can change between different regions of space, but the *total amount of charge* across all space is conserved. Thus, we call it the *global* conservation of charge.

The second is the **continuity equation**:

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot \mathbf{J} \tag{721}$$

The continuity equation imposes the additional constraint that any change in the charge density ρ must be coupled with a change in the current density $\mathbf{J}$. In physical terms, it means that the charge within a region can *only* decrease by charges moving in (or out) of the region.

Let's examine what this means. The global conservation of charge would allow charges to vanish from one local region and appear in another region (so long as the total charge does not change). But the continuity equation would say that this is *impossible*. The amount of charge within a region can **only** change by charges *entering or leaving the region*. Furthermore, any amount of charge *entering* a region must mean that the same amount of charge *left* another region. Thus the net charge of the two regions does *not* change, and the total amount of charge across all space does not change either.

To the best of our knowledge, charge conservation is universal. Requiring that the continuity equation be satisfied is a fundamental requirement to ensuring that solutions to Maxwell's equations are physically possible.

0.2.6 Mathematical methods of modern physics

Having covered the Lagrangian and Hamiltonian formulations of classical mechanics, we are now ready to move to relativistic and quantum physics. These are the two theories that form the core of **modern physics**, and they require different mathematical tools compared to classical mechanics. Among just a few topics we will explore are tensors, tensor calculus, complex numbers, operators, and eigenvalue problems - all of which play a key role in building our modern understanding in physics.

Tensor Calculus

As we move into more advanced physics, vector calculus quickly becomes insufficient, and vector notation becomes incredibly inelegant and hard to work with. We need a new type of mathematical object to describe vector-like quantities in these theories - **tensors**. Fluid mechanics, advanced electrodynamics, and many numerical methods for partial differential equations *all* use tensors. General relativity is famously written in the language of differential geometry, known for its difficulty, and uses tensors near-exclusively. This is hopefully a gentler guide to tensors, one that preserves the essense of their mathematical beauty without baffling the mind.

Invariants The whole idea behind tensors is that we want to create mathematical objects that are the same in every coordinate system - we call that **invariant**. Why is this? The reason is because in the universe, objects *are* invariant. Intuitively, if you ask a friend to push you, whether you measure the force of your friend's push in cylindrical or spherical or cartesian coordinates, you're going to feel the same force. If you have, say, a magnetic force, that force is going to still exist and be the same physical quantity whichever coordinate system we use. The same goes for any other physical quantity, such as momentum, velocity, energy, etc. It makes sense that just using a different coordinate system to measure, say, a racecar, wouldn't cause that racecar to be 20% faster or 15% slower - instead, the speed of the racecar would be the same. Mathematical objects that have the same *physical* quantity between differing coordinate systems are called **tensors**. Some tensors are scalars such as temperature; others are vectors such as position; still others are matrices; the commonality is that they correspond to one measurable physical object.

However, because we use coordinate systems to measure physical quantities, the *components* of a physical quantity may be dependent on the coordinate system we use, even though the physical quantity remains the same. For example, consider a vector $\vec{a}$. This vector would have the following components in Cartesian coordinates:

$$\begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix} \tag{722}$$

And the following components in polar coordinates:

$$\begin{bmatrix} 2 \\ \frac{\pi}{3} \end{bmatrix} \tag{723}$$

These represent the exact same vector, but the *components* are different. This is the motivation to find common transformation laws that help transform the the components of the same physical quantity - the same tensor - between differing coordinate systems. Thus, we can express the same physical object, say force, in one coordinate system or another, but the laws of physics that describe that physical object would be the **same** in any coordinate system.

Tensors and tensor notation A 3D vector in Cartesian coordinates can be written in terms of the basis vectors (e_x, e_y, e_z):

$$\vec{V} = a_x e_x + a_y e_y + a_z e_z \tag{724}$$

Thus, we can write the sum more compactly as follows:

$$\vec{V_i} = \sum_{i=0}^{3} a_i e_i \tag{725}$$

In Einstein summation notation, the summation is implied, so we can more simply write our vector as:

$$\vec{V_i} = a_i e_i \tag{726}$$

When writing vectors as tensors, it's customary to put the indices on top (as superscripts) instead of on the bottom (subscripts). Note that these are **not** raising to a power. So we now have:

$$\vec{V} \Rightarrow V^i \tag{727}$$

Given two vectors $\vec{A}$ and $\vec{J}$, we can now write the inner product more succintly as:

$$\vec{A} \cdot \vec{J} = A^i J^i \tag{728}$$

And we can define other vector field operations, such as the gradient, using the same notation:

$$\nabla f = \frac{\partial f}{\partial x^i} \tag{729}$$

Note
Remember that when writing tensors, the letters we use for the summed-over index are arbitary. We use i because that's how we write summation most often, but we could use any letter we want. Later one, we'll use greek letters when denoting tensors in spacetime, but that's just a convention, not an absolute rule.

Contravariant tensors, usually just called vectors, are most generally written with Einstein summation convention. We denote their components with V_i and their basis with e_i, so:

$$\vec{V} = V^i e_i \tag{730}$$

Covariant tensors, usually just called covectors, are also written using the convention. They are marked by a tilde over the letter, and are denoted (in terms of their basis e^i) as:

$$\tilde{V} = V_i e^i \tag{731}$$

Covectors in the 2D and 3D case can be thought of as row vectors that are multiplied by column vectors to give the dot product:

$$\begin{bmatrix} a & b & c \end{bmatrix} \cdot \begin{bmatrix} d \\ e \\ f \end{bmatrix} \tag{732}$$

Note on generalization
In the more general N dimensional case, covectors are what you multiply by a vector to obtain a scalar using the dot product. They don't have to be row vectors.

Tensors that are not scalars, covectors, or vectors are formed by multiplying one or more vectors with covectors using the rules of the tensor product (which we'll see next) and tensor algebra.

Rules of Tensor Algebra Scalar multiplication rule:

$$T_{\alpha\beta} = kX_{\alpha\beta} \tag{733}$$

Addition/subtraction rule:

$$T^{\alpha}_{\ \beta} = A^{\alpha}_{\ \beta} \pm B^{\alpha}_{\ \beta} \tag{734}$$

Multiplication rule (which raises the rank of a tensor) - also called **tensor product**:

$$T^{\alpha\beta} = A^{\alpha} B^{\beta} \tag{735}$$

$$T_{\lambda}^{\alpha\beta\gamma} = (A^{\alpha}B^{\beta})\, C_{\lambda}^{\gamma} \tag{736}$$

Contraction rule (which lowers the rank of a tensor) by multiplying with another tensor with an identical index in the opposite position:

$$T_{\lambda} = A_{\lambda}^{\alpha}B_{\alpha} \tag{737}$$

$$T_{\lambda} = A_{\alpha\lambda}B^{\alpha} \tag{738}$$

The other tensor to be multiplied by is often going to be the *metric tensor* - more on that later.
The contraction rule also applies when a tensor has identical upper and lower indices:

$$T^{\gamma} = T_{\alpha}^{\alpha\gamma} \tag{739}$$

Tensors can also obey several crucial symmetries. A symmetric tensor is one in the form:

$$T_{\alpha\beta} = T_{\beta\alpha} \tag{740}$$

Whereas an antisymmetric tensor is in the form:

$$T_{\alpha\beta} = -T_{\beta\alpha} \tag{741}$$

The double contraction of a symmetric tensor $S^{\alpha\beta}$ and antisymmetric tensor $A_{\alpha\beta}$ is zero:

$$A_{\alpha\beta}S^{\alpha\beta} = 0 \tag{742}$$

Basis vectors and basis covectors obey the relation:

$$e^{i}e_{j} = \delta^{i}{}_{j} \tag{743}$$

Where $\delta^{i}{}_{j}$ (called the Kronecker delta) is defined as:

$$\delta_{ij} = \begin{cases} 1 \text{ if } i = j \\ 0 \text{ if } i \neq j \end{cases} \tag{744}$$

For example, the 3D version of the Kronecker delta is given by:

$$\delta_{ij} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{745}$$

You might recognize this as the **identity matrix**, and the special property of the identity matrix is that it returns the original vector when multiplied by any vector.

If we take two basis vectors in Cartesian space and take their dot product, the result is Kronecker delta:

$$e_{i} \cdot e_{j} = \delta_{ij} \tag{746}$$

The Kronecker delta is the Cartesian form of the more general **metric tensor**, which is when we take the dot product of two non-Cartesian basis vectors:

$$e_{\mu} \cdot e_{\nu} = g_{\mu\nu} \tag{747}$$

Given a covector $\tilde{A}$ and a vector $\vec{B}$, we find that we end up with a scalar using tensor notation :

$$\tilde{A}\vec{B} = A_{i}B^{j}e^{i}e_{j} = A_{i}B^{j}\delta^{i}{}_{j} = S \tag{748}$$

Let's examine this equation in detail to familiarize ourselves with tensor algebra. First, we wrote out our vector and covector in terms of their components and basis:

$$\tilde{A} \Rightarrow A_i e^i \tag{749}$$

$$\vec{B} \Rightarrow B^j e_j \tag{750}$$

We then used the Kronecker delta to rewrite the dot product of the basis vectors as the Kronecker delta:

$$e^i e_j = \delta^i{}_j \tag{751}$$

Now, we used the Kronecker delta to relabel the indices of the components of our vector and covector:

$$A_i B^j \delta^i{}_j = A^i{}_i B^j{}_j \tag{752}$$

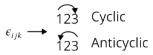
Finally, we used tensor contraction due to the same index on the upper and lower indices to contract the components to scalars:

$$A^i{}_i B^j{}_j = AB \Rightarrow S \tag{753}$$

We can also write the cross product using tensors. To do this, we introduce the **Levi-Civita symbol**:

$$\epsilon_{ijk} = \begin{cases} 1 \text{ if cyclic permutation} \\ -1 \text{ if anticyclic permutation} \\ 0 \text{ if indices repeat} \end{cases} \tag{754}$$

A cyclic permutation is if the indices go in order from left to right, such as 123, 231, or 312. An anticyclic permutation is if the indices go in order from right to left, such as 321, 213, or 132. Indices repeat in any case like 122 or 233:



With this defined, we can express the cross product as:

$$\vec{A} \times \vec{B} = \epsilon_{ijk} A^j B^k \tag{755}$$

Or if we insist on producing a vector rather than a covector, though the result is identical:

$$\vec{A} \times \vec{B} = \epsilon^i{}_{jk} A^j B^k \tag{756}$$

Coordinate transformations To understand tensor transformations, let's consider the case of transforming a vector $\vec{V}$ between (x, y) and (r, θ) coordinates.

The crux of the problem is that we have two definitions of $\vec{V}$:

$$\vec{V} = v^x e_x + v^y e_y \tag{757}$$

$$\vec{V} = v^r e_r + v^\theta e_\theta \tag{758}$$

What we need is a reliable way to express e_x in terms of r, and e_y in terms of θ.

At first glance, there is not much we can do. However, we know that the components of a vector are **linearly independent** - that is, do not affect each other. This means we can say that:

$$v^x e_x = v^r e_r \tag{759}$$

$$v^y e_y = v^\theta e_\theta \tag{760}$$

Let's focus on the first equation first. We can replace v^x with just a function x, as v^x is really just a displacement in the x direction. We can similarly replace v^r with a function r, as v^r is just a displacement in the r direction:

$$xe_x = re_r \tag{761}$$

Given that this first equation holds true, then the same should be true in terms of the differentials of the components:

$$dxe_x = dre_r \tag{762}$$

Now, if we divide by dx on both sides of the equation, we have:

$$e_x = \frac{dr}{dx}e_r \tag{763}$$

Using the same process for the second equation, we get a similar result for e_y:

$$e_y = \frac{d\theta}{dy}e_\theta \tag{764}$$

Now, plugging in this to our original expression for our vector:

$$\vec{V} = v^x e_x + v^y e_y \tag{765}$$

We have:

$$\vec{V} = v^x \frac{dr}{dx}e_r + v^y \frac{d\theta}{dy}e_\theta \tag{766}$$

But we also know that:

$$\vec{V} = v^r e_r + v^\theta e_\theta \tag{767}$$

Equating this equation and the previous, we have:

$$v^r e_r + v^\theta e_\theta = v^x \frac{dr}{dx}e_r + v^y \frac{d\theta}{dy}e_\theta \tag{768}$$

Using index notation, we can write this (with $i' = r, \theta$ and $i = x, y$):

$$v^{i'} e_{i'} = v^i \frac{dx^{i'}}{dx^i}e_{i'} \tag{769}$$

> **Note**
> Remember the summation is implied, which is how we were able to shrink the two terms on the left hand side and the right hand side to just one term on either side.

We notice that the $e_{i'}$ term appears on both sides of the equation, so we can cancel them out to have:

$$v^{i'} = v^i \frac{dx^{i'}}{dx^i} \tag{770}$$

Finally, note that technically, $x^{i'}$ is a multivariable function, so:

$$v^{i'} = v^i \frac{\partial x^{i'}}{\partial x^i} \tag{771}$$

We have derived the vector transformation law. A similar derivation, just with oppositely-placed indices, is true for covectors.

Tensor transformation law What's most interesting about tensors is how they are defined, because they are defined in terms of *how they transform*. For instance, let's go through the formal definitions of contravariant and covariant tensors. They are defined like so:

Vectors (contravariant tensors) transform by the following transformation law from coordinate system x^i to $x^{i'}$:

$$V^{i'} = \frac{\partial x^{i'}}{\partial x^i} V^i \tag{772}$$

While this definition is already workable, mathematicians more abstractly define contravariant tensors in a slightly different fashion: as the tangent vector to a parametrized curve in spacetime. The parameter used is most commonly τ, the proper time. So:

$$V^i = \frac{dx^i}{d\tau} = \left(\frac{dx^0}{d\tau} + \frac{dx^1}{d\tau} + \frac{dx^2}{d\tau} + \frac{dx^3}{d\tau} \right) \tag{773}$$

Note
Remember: We are using the convention $(x^0, x^1, x^2, x^3) = (t, x, y, z)$

Covectors (covariant tensors) transform instead by the following transformation law:

$$V_{i'} = \frac{\partial x^i}{\partial x^{i'}} V_i \tag{774}$$

Again, we can also define covariant tensors with an alternate method. Consider a function $f(x^i)$, where again $x^i = (x^0, x^1, x^2, x^3) = (t, x, y, z)$. Its gradient would be given by:

$$V_i = \frac{\partial f}{\partial x^i} = \left(\frac{\partial f}{\partial x^0}, \frac{\partial f}{\partial x^1}, \frac{\partial f}{\partial x^2}, \frac{\partial f}{\partial x^3} \right) \tag{775}$$

The components of the gradient, given by $\frac{\partial f}{\partial x^i}$, are the components of a covariant tensor.

So now, having seen how covariant tensors and contravariant tensors transform, we can more generally describe what a tensor is:

Definition of a tensor
A mathematical object represented by a collection of components that transform according to certain transformation laws.

The **type** (m, n) of a tensor is given by how many upper indices (m) it has, and how many lower indices (n) it has. The total number of upper and lower indices is its **rank**.

Now, let's look through several typical tensors:

First, we have scalars, which are $(0, 0)$ or rank-0 tensors.

$$S \tag{776}$$

Then, we have vectors and covectors, both rank-1, which are respectively $(1, 0)$ and $(0, 1)$ tensors. We usually write vectors like this:

$$V^i = \begin{bmatrix} a \\ b \\ c \\ d \end{bmatrix} \tag{777}$$

And we usually write covectors like this:

$$V_i = \begin{bmatrix} a & b & c & d \end{bmatrix} \tag{778}$$

Multiplying a vector and a covector is taking their **tensor product**, which gives a rank-2 tensor, also called a matrix:

$$V^\alpha V_\beta = T^\alpha{}_\beta \tag{779}$$

$$T^\alpha{}_\beta = \begin{pmatrix} T_{00} & T_{01} & T_{02} & T_{03} \\ T_{10} & T_{11} & T_{12} & T_{13} \\ T_{20} & T_{21} & T_{22} & T_{23} \\ T_{30} & T_{31} & T_{32} & T_{33} \end{pmatrix} \tag{780}$$

The tensor product is the primary way we construct new tensors in tensor calculus (which we'll use in differential geometry). Any two tensors can be multiplied through the tensor product, so we can use the tensor product to construct any tensor. Most importantly, because the tensor transformation rule is preserved every time we take the tensor product, we can now use it to define *any* tensor. This is the most general **definition of a tensor**:

$$T^{\mu'_1\mu'_2\ldots\mu'_m}_{\nu'_1\nu'_2\ldots\nu'_n} = \frac{\partial x^{\mu'_1}}{\partial x^{\mu_1}}\frac{\partial x^{\mu'_2}}{\partial x^{\mu_2}}\cdots\frac{\partial x^{\mu'_m}}{\partial x^{\mu'_m}}\frac{\partial x^{\nu_1}}{\partial x^{\nu'_1}}\frac{\partial x^{\nu_2}}{\partial x^{\nu'_2}}\cdots\frac{\partial x^{\nu_n}}{\partial x^{\nu'_n}}T^{\mu_1\mu_2\ldots\mu_m}_{\nu_1\nu_2\ldots\nu_n} \tag{781}$$

This horribly long and scary-looking equation is what gives tensors their reputation for frying minds, but, reassuringly, you almost never need to use this definition in practice. However, the definition is really just a formalized version of a simple idea: a tensor is a combination of vector and covector transformations that preserve the tensor transformation law. We therefore arrive at the common but unhelpful observation that:

> "A tensor is anything that transforms like a tensor."

Note
This section along with its following two sections deal with increasingly-complex mathematics and physics. They are only meant to serve as illustrative examples for those curious. You are **not expected** to immediately get what they mean, and you don't need to understand them to be able to use tensors!

Practical application: Newton's laws in Tensor Calculus To rewrite Newton's laws of motion in tensor calculus is not difficult. For example, we can take $\vec{F} = m\vec{a}$ and rewrite with tensors as:

$$F^i = ma^i \tag{782}$$

We can rewrite a^i as:

$$a^i = \frac{d^2x^i}{dt^2} \tag{783}$$

Allowing us to write Newton's second law as:

$$F^i = m\frac{d^2x^i}{dt^2} \tag{784}$$

In practice, however, this is only true in Cartesian coordinates. To generalize to any set of curvilinear coordinates, the actual correct formula is:

$$F^i = m\left(\frac{d^2x^i}{dt^2} + \Gamma^i_{\mu\nu}\frac{dx^\mu}{dt}\frac{dx^\nu}{dt}\right) \tag{785}$$

Where $\Gamma^i_{\mu\nu}$ is a quantity that depends on the metric $g_{\mu\nu}$ of the space (more about this next chapter).

Our new form of Newton's (2nd) law is now valid in all coordinate systems. Obviously, though, no one writes Newton's laws in this way; this is just a demonstration. Instead, let's look at a more practical example.

Practical application 2: Maxwell's equations in tensor calculus When using tensor calculus to express Maxwell's equations, the advantages of tensors becomes much more clear. But first, let's take a look at Maxwell's typical laws (these are expressed in SI units):

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} \tag{786}$$

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{787}$$

$$\nabla \times \vec{B} = 0 \tag{788}$$

$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \tag{789}$$

We can define a quantity $\vec{A}$ called the vector potential, where:

$$\vec{B} = \nabla \times \vec{A} \tag{790}$$

And a quantity called ϕ called the scalar potential, where:

$$\vec{E} = -\nabla\phi - \frac{\partial \vec{A}}{\partial t} \tag{791}$$

This allows us to simplify Maxwell's 4 equations to just 2:

$$A^\mu = \left(\frac{1}{c}\phi, \vec{A}\right) \tag{792}$$

$$J^\mu = (c\rho, \vec{J}) \tag{793}$$

We can then define an electromagnetic tensor F_{ij} like this:

$$F_{ij} = \frac{\partial A_j}{\partial x^i} - \frac{\partial A_i}{\partial x^j} \tag{794}$$

Now, using the electromagnetic field tensor, we can rewrite Maxwell's equation using just 2 tensor equations, which, like any tensor equation, is coordinate-independent[1]:

$$\partial_m F_{ik} + \partial_k F_{mi} + \partial_i F_{km} = 0 \tag{795}$$

$$\partial_i F^{ik} = \mu_0 J^k \tag{796}$$

Practical application 3: Differential geometry in General Relativity Studying differential geometry as it applies to General Relativity starts with the basic Pythagorean theorem. The distance Δs between any 2 points (x_1, y_1) and (x_2, y_2) in a 2D space can be found with the theorem:

$$\Delta s^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2 \tag{797}$$

Defining $\Delta x = x_2 - x_1, \Delta y = y_2 - y_1$, we can rewrite this as:

$$\Delta s^2 = \Delta x^2 + \Delta y^2 \tag{798}$$

[1] A useful reference, from which this derivation was based, can be found at https://profoundphysics.com/are-maxwell-equations-relativistic/

Note
Here, we use the notation that $\Delta x^2 = (\Delta x)^2$ and $\Delta y^2 = (\Delta y)^2$.

Taking the limit as $\Delta x, \Delta y$ grows infinitesimally tiny, we can replace Δs with ds, and $\Delta x, \Delta y$ become dx, dy. So the calculus version of Pythagoras's theorem is given by:

$$ds^2 = dx^2 + dy^2 \tag{799}$$

We call this the **line element** of the space, which tells us how distances between two points are measured.

The line element of 3D space is very similar to the 2D case - just add an additional coordinate:

$$ds^2 = dx^2 + dy^2 + dz^2 \tag{800}$$

However, in 4D spacetime, things are a bit different. In flat spacetime, also called Minkowski spacetime, the line element is given by:

$$ds^2 = -c^2dt^2 + dx^2 + dy^2 + dz^2 \tag{801}$$

This is equal to the inner product of the infinitesimal displacement vectors of Euclidean space multiplied by a matrix:

$$ds^2 = \begin{bmatrix} cdt & dx & dy & dz \end{bmatrix} \begin{bmatrix} cdt \\ dx \\ dy \\ dz \end{bmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{802}$$

This matrix is called the **metric tensor** - we've seen this before, it generalizes the dot product to spaces where the basis vectors aren't constant. Why the need for a metric tensor? Imagine we had two vectors, x^μ and x^ν. Their product would be given by:

$$x^\mu \cdot x^\nu = x^\mu e_\mu x^\nu e_\nu = g_{\mu\nu} x^\mu x^\nu \tag{803}$$

This is why we need a metric tensor. Note that, as we saw before, the metric tensor is a rank-2 tensor, as it is the tensor product of two rank-1 vectors (which are contravariant tensors).

Also note that we often use the metric tensor for raising or lowering indices via tensor contraction. Thus:

$$T_\beta = g_{\alpha\beta} T^\alpha \tag{804}$$

$$T^\beta = g^{\alpha\beta} T_\alpha \tag{805}$$

And if we multiply the metric tensor by its inverse, we get the identity matrix, equal to the Kronecker delta, which yields a scalar:

$$g^{\alpha\beta} g_{\beta\mu} = \delta^\alpha_\mu \tag{806}$$

Finally, we can convert line elements between different coordinate systems. A more tensor-based system for doing so will be given the next chapter, but as just an example, let's attempt to convert the 3D line element to spherical coordinates.

In spherical coordinates, (r, θ, ϕ), we recall that spherical coordinates are defined in terms of cartesian coordinates as follows:

$$x = r \sin\theta \cos\phi \tag{807}$$

$$y = r \sin\theta \sin\phi \tag{808}$$

$$z = r \cos\theta \tag{809}$$

Thus, after evaluating the differentials using the total differential rule, we find that:

$$dx = \frac{\partial x}{\partial r}dr + \frac{\partial x}{\partial \theta}d\theta + \frac{\partial x}{\partial \phi}d\phi \tag{810}$$

$$= \sin\theta\cos\phi dr + r\cos\theta\cos\phi d\theta - r\sin\phi\sin\theta d\phi \tag{811}$$

$$dy = \frac{\partial y}{\partial r}dr + \frac{\partial y}{\partial \theta}d\theta + \frac{\partial y}{\partial \phi}d\phi \tag{812}$$

$$= \sin\phi\sin\theta dr + r\sin\theta\cos\theta d\theta + r\sin\theta\cos\phi d\phi \tag{813}$$

$$dz = \frac{\partial z}{\partial r}dr + \frac{\partial z}{\partial \theta}d\theta \tag{814}$$

$$= \cos\theta dr - r\sin\theta d\theta \tag{815}$$

After an ungodly long process of adding in our evaluated values for the squares of the three differentials dx, dy, and dz, we finally get the line element in spherical coordinates:

$$dl^2 = dr^2 + r^2 d\theta^2 + r^2\sin^2\theta d\phi^2 \tag{816}$$

Therefore, our metric tensor becomes:

$$g_{\alpha\beta} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & r^2\sin^2(\theta) \end{pmatrix} \tag{817}$$

Which still satisfies:

$$dl^2 = \begin{bmatrix} dr & d\theta & d\phi \end{bmatrix} \begin{bmatrix} dr \\ d\theta \\ d\phi \end{bmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & r^2 & 0 \\ 0 & 0 & r^2\sin^2(\theta) \end{pmatrix} \tag{818}$$

After doing all of these calculations with tensors, why do we need to use them at all? It's because **tensor equations take the same form in any coordinate system**. This important fact will be crucial in working on a theory of general relativity.

Mathematics for Quantum Mechanics

> "Life is complex sometimes"
> **Unknown author**

Typically, the spirit of the Elara handbook is to present all the ideas of a topic at once, without a lot of prior background knowledge. However, for quantum mechanics, it is *especially* good to get a grasp of complex numbers, because the use of complex numbers often makes the mathematical formalism of quantum mechanics foreign and indecipherable. So dive in, enjoy, and let's see how an mathematical idea that was once said to exist "only in our minds" would turn out to be one of the most invaluable tools to solve real-world problems...

Imaginary and complex numbers Say we were struggling with a problem. In fact, a very peculiar problem. Suppose that problem was:

$$x^2 + 1 = 0 \tag{819}$$

How could we solve this for x? Using the usual algebraic methods, we immediately hit a roadblock:

$$x = \sqrt{-1} \tag{820}$$

But surely that does not make sense, you say! You cannot take the square root of a negative number, certainly! Or can you...?

Indeed, the above equation does not make sense in terms of the numbers we are familiar with, called the real numbers, and denoted with $\mathbb{R}$. However, we can define - or perhaps, to use a more artistical term, imagine - a set of numbers for which this equation *does* make sense. We therefore define:

$$i = \sqrt{-1} \tag{821}$$

We can then create a system of numbers, which we term **imaginary numbers**, based on i: $2i$, $3i$, πi, $1.52434i$, $\frac{3}{4}i$. And we can go further! We can even create a system of numbers that contains both real numbers and imaginary numbers, such as $3 + 4i$, $5 + 6i$, or $9 - 2i$. We call these **complex numbers**. A complex number z is defined by:

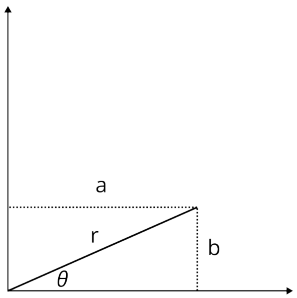
$$z = a + bi \tag{822}$$

The real and imaginary parts of the complex number z are:

$$\mathrm{Re}(z) = a, \quad \mathrm{Im}(z) = bi \tag{823}$$

Complex numbers can be added, subtracted, multiplied and divided just like real numbers. Other than containing imaginary parts as well as real parts, they may well just be considered ordinary numbers.

The complex plane But what makes complex numbers so...different? Perhaps one way of thinking about them is that they are two-dimensional numbers. Yes, you heard that right, not two dimensional *vectors* - two dimensional *numbers*. We can draw out complex numbers by plotting their real part on the x-axis and imaginary part on the y-axis, to get the **complex plane**:



Here, notice that $r = \sqrt{a^2 + b^2}$, and that thanks to trigonometry, $a = r\cos\theta$ and $b = r\sin\theta$. If we take the complex number formula, and substitute the identities, then we have:

$$z = r\cos\theta + i(r\sin\theta) \tag{824}$$

If $r = 1$, then:

$$z = \cos\theta + i\sin\theta \tag{825}$$

This is called the **polar** representation of complex numbers.

Euler's formula We already know from the polar representation of complex numbers that any complex number $z = a + bi$ can be written as:

$$z = \cos(\theta) + i\sin(\theta) \tag{826}$$

If we were to take the derivative of z with respect to θ, we'd get:

$$\frac{dz}{d\theta} = i\cos\theta - \sin\theta \tag{827}$$

Now, this is interesting, because unlike many derivatives, we can actually write the right-hand side of this equation *in terms of* z:

$$i\cos\theta - \sin\theta = i \cdot (\cos\theta + i\sin\theta) \Rightarrow \frac{dz}{d\theta} = iz \tag{828}$$

We can solve this differential equation, like we did for other differential equations earlier in this book. First, separate the variables:

$$dz = izd\theta \tag{829}$$

$$\int \frac{dz}{z} = \int id\theta \tag{830}$$

Then, integrate to get:

$$\ln(z) = i\theta + C \tag{831}$$

We can take the exponential of both sides to get:

$$z = e^{i\theta + C} \tag{832}$$

Now we just need an initial condition, $z(0)$. But we know that:

$$z = \cos(\theta) + i\sin(\theta) \tag{833}$$

So, $z(0)$ would be:

$$z(0) = \cos(0) + i\sin(0) = 1 \tag{834}$$

Using our initial condition $z(0) = 1$, we can now solve the differential equation exactly:

$$z = e^{i\theta} \tag{835}$$

And remember, since $z = \cos(\theta) + i\sin(\theta)$, we can substitute that in to get:

$$e^{i\theta} = \cos(\theta) + i\sin(\theta) \tag{836}$$

This is **Euler's formula**. Note that when $\theta = \pi$, we have:

$$e^{i\pi} = -1 \tag{837}$$

Which we can simplify to:

$$e^{i\pi} + 1 = 0 \tag{838}$$

This is **Euler's identity**, one of the most famous equations ever, showing that two previously completely unrelated ideas, those being trigonometry and exponentiation, are actually linked. You've got to appreciate Euler's skill as a mathematician after looking at this equation, and wonder if the true genius was Euler, not Einstein.

Complex conjugation If we have an imaginary number $a+bi$, it turns out that if we multiply that by its **complex conjugate** $a - bi$, we get a real number:

$$(a + bi)(a - bi) = a^2 + b^2 \tag{839}$$

This is because by definition, $i^2 = 1$, so $-b^2i^2 = -b^2(-1) = b^2$.

Complex waveforms Using Euler's formula, we can write waves in terms of a function of e rather than using sines and cosines:

$$\psi(x, t) = Ae^{i\theta(x,t)} \tag{840}$$

Why do this? Exponentials are far, far easier to differentiate and integrate than sines and cosines, and so writing out waves in this form allows them to be more easily expressed in quantum mechanics, which involves hoards of derivatives and integrals.

Before we finish this short chapter and jump into quantum mechanics, there is one last fact about complex numbers that is trivial, but confuses a lot of people. That fact is that:

$$-i = \frac{1}{i} \tag{841}$$

Remember this fact, and we will meet you again in quantum mechanics!

0.2.7 Quantum mechanics and modern physics

Up to this point, we have only discussed **classical mechanics**. Classical mechanics comprises physics as it was before the turn of the 20th century. By that point, physicists have already figured out (or at least had a good understanding of) many important areas of physics, such as mechanics, electrodynamics, and gravitation. But when they tried to apply their existing theories to the study of the atom (and matter on a microscopic scale in general), their predictions were wildly off.

So a new theory of physics was developed - one radically different from classical mechanics. This is the famous theory of **quantum mechanics**, and while it explained - brilliantly, in fact - the inner workings of atoms and the fundamental building blocks of matter, it leaves many questions unanswered, much to the disdain of physicists. The hopes of the physicists who wanted to create a theory of physics to explain *all* observered phenomena were dashed.

For our purposes, however, we will discuss quantum mechanics because of its importance in describing a variety of complex phenommena that are relevant to the Project's technologies. We hope that the following sections can break down the famously-intimidating theory of quantum mechanics into something much more comprehensible, and by the end, you will be left with a solid understanding of quantum physics.

Quantum mechanics, Part 1

> "When you change the way you look at things, the things you look at change."
> **Max Plank**

Up to this point, we've explored physics that obeyed the principles of predictability and certainty. We took it for granted that particles moved on definite paths, that if we knew everything about the system we could predict everything about how it would evolve. All of those assumptions completely break down in quantum mechanics, when we take a look at the world near the scale of atoms. We'll explore this fascinating new world - and in time, hopefully realize that, as Sean Carroll said, *"the quantum world is not spooky or incomprehensible - it's just way different"*.

```
from matplotlib import cm
import matplotlib.pyplot as plt
import numpy as np

%matplotlib inline
plt.rcParams["font.family"] = "serif"
plt.rcParams['mathtext.fontset'] = 'stix'
```

Quantization and wavefunctions When we speak of observable properties of a macroscopic object, such as its speed, momentum, and energy, we expect that these properties can take on *any* value. For instance, we can have a speed of 5.34 m/s, or 100,000 m/s, or 0.0015 m/s. These are called **continuous** properties.

But in the quantum world, certain properties of microscopic particles can *only take* certain values - specifically, values that are a multiple of some indivisible unit value. One way to think about it is to imagine you have several basketballs - it would make sense to have 0 basketballs, or 50 basketballs, or 12 basketballs, but it wouldn't make sense to have π basketballs or 1.513 basketballs. The indivisible unit, in our case, is one basketball, and in quantum mechanics we call the unit a **quantum** (plural *quanta*). For example, the quantum of charge is the elementary charge constant $e = 1.602 \times 10^{-19}$ Coulombs, which is *exactly* the charge of a single electron (ignoring the sign). This means that you can only have a charge of $2e$ or $5e$ or $17e$, never $1.352e$. We call properties that are composed of quanta **quantized**.

Energy is important, because it was Einstein who first discovered that the energy of light is quantized, and in particular that the quantum of light energy is given by:

$$E_p = hf \tag{842}$$

where h is Plank's constant, approximately 6.626×10^{-34}, and f is the frequency of the wave - this is a result we will manually derive later. According to Einstein's formula, this means that if you measure the energy of any light, it must be $0E_p$ or $1E_p$ or $5E_p$ or $1000E_p$, not $1.5E_p$. Light can't just take any energy it wants; the energy must strictly be $N \cdot E_p$, where N is the number of photons in the measured light.

Einstein recognized that the quantization of light isn't just a mathematical rule of integer multiples, but that quanta of light are actually the energy carried by physical particles that we call **photons**. But isn't light a wave? Wasn't that what Maxwell's equations said? Physicists later recognized that Maxwell's equations were *incomplete* - at a certain limit for smaller and smaller particles, they no longer offer a full description of electromagnetism.

More strange observations followed the discovery of light quanta (photons) and the recognition that electrons were the quantum of electric charge. Classically, electrons were modelled as (point) particles, with exact physical properties. But if an electron is a particle, why can't we just measure its observable properties (such as position) and know how it behaves? Through experimentation, scientists tried, and failed, to do exactly this. They concluded that quantum particles cannot be found exactly. We can only measure probabilities by using a **wavefunction** that describes the state of a quantum system.

The wavefunction is oftentimes a complex function, and has no direct physical meaning (which makes sense; all physical quantities are real numbers, not complex numbers!) However, it turns out that if you integrate the square of the wavefunction, you find the **probability** of a particle being at a particular position, which we write:

$$P(x) = \int |\Psi(x,t)^2| dx \tag{843}$$

> **Note**
> We will show later why $P(x)$ does not have a dependence on time, even though $\Psi(x,t)$ is a function of both space and time.

Note that we can generalize this to multiple dimensions:

$$P(\mathbf{r}) = \int |\Psi(\mathbf{r},t)^2| \, dx \, dy \, dz \tag{844}$$

where $\mathbf{r} = (x,y,z)$ is a shorthand for the three spatial coordinates. As an electron (or any other quantum particle) has to be *somewhere*, integrating over all space gives a probability of 1, that is, 100% probability of being at *some point in space*:

$$\int_{-\infty}^{\infty} |\Psi(\mathbf{r},t)^2| \; dx \; dy \; dz = 1 \tag{845}$$

> **Note**
> We typically use an uppercase Ψ for time-dependent wavefunctions and a lowercase ψ for time-independent wavefunctions.

The Schrödinger equation If quantum systems are fundamentally probabilistic in nature, one may think that predicting any physical properties of them would be near-impossible. Fortunately, this is not the case. In fact, there exists a partial differential equation that can be solved to find the wavefunction $\Psi(\mathbf{r},t)$ for a single quantum particle: the **Schrödinger equation**. It is given by:

$$i\hbar \frac{\partial \Psi}{\partial t} = \left(-\frac{\hbar^2}{2m} \nabla^2 + V(\mathbf{r},t) \right) \Psi \tag{846}$$

Where $\hbar = h/2\pi$ is **Planck's reduced constant**, m is the mass of the particle, V is the potential energy of the particle, and i is the imaginary unit. Note that this expression for the Schrödinger equation applies to a single particle, *not* a system of particles; we will cover the multi-particle Schrödinger equation later.

Solving the Schrödinger equation yields the wavefunction, which can then be used to find the probability of a particle being at a certain position. As another stroke of luck, we find that when the potential V only depends on position, that is, $V = V(\mathbf{r})$, then the Schrödinger equation becomes a *separable* linear differential equation.

This means that we have two (main) techniques for solving the Schrödinger equation by hand, based on the techniques we saw previously when we looked at how to solve PDEs in the differential equations section in the first chapter. We can either use the "method of inspired guessing" or the proper separation of variables technique. And we will cover both here.

Solving by inspired guessing To solve the Schrödinger equation by inspired guessing, we first note that this approach only works when $V(\mathbf{r}) = 0$. In addition, while we can use this technique in three dimensions, we will focus on the easier 1-dimensional case to start. The Schrödinger equation for $V(\mathbf{r}) = 0$ (which is the case for a **free particle** unconstrained by any potential) becomes:

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} \tag{847}$$

Note that the Schrödinger equation looks quite similar to the heat equation that we studied earlier in the section on differential equations. Remember that the heat equation had solutions of sinusoids in space multipled with exponential solutions in time, so we would expect that the solutions to the Schrödinger equation would be at least somewhat similar. However, since the Schrödinger equation describes a complex-valued wavefunction instead of a real-valued heat distribution, we would expect *complex-valued* sinusoids. Recall that by Euler's formula, we have $e^{i\phi} = \cos\phi + i\sin\phi$ - therefore, we could make an educated guess that the spatial portion of the wavefunction would be a complex exponential with the form:

$$\psi_x = e^{ikx} \tag{848}$$

where k is some constant factor to get the units right (remember that transcendental functions like e^x can only take dimensionless values in their argument). Now, recalling that the heat equation had exponential solutions in time, we would expect a *complex exponential* in time (since again, the Schrödinger describes a complex-valued wavefunction). In addition, recalling that the heat equation's solutions in time were *decaying* exponentials, we would presume that the argument to the complex exponential would have a negative sign, with some constant factor to get the units right (which we'll call ω). So we could make the following guess:

$$\psi_t = e^{-i\omega t} \tag{849}$$

Finally, recalling that a solution to the heat equation was formed by multiplying its spatial and temporal solution components, we can do the same to have:

$$\Psi(x,t) = \psi_x\psi_t = e^{ikx}e^{-i\omega t} = e^{i(kx-\omega t)} \tag{850}$$

This is a plane wave that is a *particular solution* to the Schrödinger equation, and we can indeed verify it is correct by substituting it back into the Schrödinger equation. First, we compute the derivatives:

$$\frac{\partial\Psi}{\partial t} = \frac{\partial}{\partial t}e^{i(kx-\omega t)} = -i\omega e^{i(kx-\omega t)} \tag{851}$$

$$\frac{\partial\Psi}{\partial x} = \frac{\partial}{\partial x}e^{i(kx-\omega t)} = ike^{i(kx-\omega t)} \tag{852}$$

$$\frac{\partial^2\Psi}{\partial x^2} = \frac{\partial}{\partial x}\left(\frac{\partial\Psi}{\partial x}\right) = \frac{\partial}{\partial x}ike^{i(kx-\omega t)} = -k^2e^{i(kx-\omega t)} \tag{853}$$

$$\tag{854}$$

Then, we plug these derivatives back into the Schrodinger equation:

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\Psi}{\partial x^2} \Rightarrow \tag{855}$$

$$i\hbar(-i\omega e^{i(kx-\omega t)}) = -\frac{\hbar^2}{2m}(-k^2)e^{i(kx-\omega t)} \tag{856}$$

$$\hbar\omega e^{i(kx-\omega t)} = \frac{\hbar^2k^2}{2m}e^{i(kx-\omega t)} \tag{857}$$

$$\hbar\omega = \frac{\hbar^2k^2}{2m} \tag{858}$$

In which we find that the Schrödinger equation is indeed satisfied, as long as $\hbar\omega = \frac{\hbar^2 k^2}{2m}$. A Python-generated plot of a plane wave is shown below (the code is unfortunately rather lengthy):

```
# Data for plotting
tmin, tmax = 0, 2
xmin, xmax = 0, 2
samples = 2000

t = np.arange(tmin, tmax, (xmax - xmin)/samples)
x = np.arange(xmin, xmax, (tmax - tmin)/samples)

# wavelengths in spatial domain
space_cyles = 3
# periods in temporal domain
time_cycles = 3

wavelength = (xmax - xmin) / space_cyles
period = (tmax - tmin) / time_cycles
freq = 1/period
k = 2 * np.pi / wavelength
omega = 2 * np.pi * freq

T, X = np.meshgrid(t, x)
phase = k*X - omega*T
psi = np.exp(1j*phase) # j is imaginary unit in Python
magn = np.abs(psi)

def wavefunction_component_ax3d(ax, x=x, t=t, X=X, T=T, phase=phase, ptype="real"):
    # create colormap based on phase
    phase_0_to_2pi = phase % (2*np.pi) # to rescale phase to [0, 2pi]
    cmap = cm.cool(phase_0_to_2pi/2)
    # distinguish between real/imaginary components via ptype arg
    if ptype == "imaginary":
        psi_label = r"$\text{Re}(\Psi)$"
        ax.plot_surface(X, T, np.imag(psi), linewidth=0,
                        facecolors=cmap)
    elif ptype == "real":
        psi_label = r"$\text{Im}(\Psi)$"
        ax.plot_surface(X, T, np.real(psi), linewidth=0,
                        facecolors=cmap)
    else:
        print("You didn't specify a ptype (real or imaginary)")
        return

    ax.set(title=r"Plot of " + psi_label)
    ax.set(xlabel=r"Position ($x$)", ylabel=r"Time ($t$)", zlabel=psi_label)
    ax.set(zlim=( -5, 5))
    ax.grid()
    return ax

def create_cbar(ax, x=x, t=t):
    # create colorbar
    phase_range = k*x - omega*t # as we need it 1 -dimensional
    norm=plt.Normalize(float(np.min(phase_range)), float(np.max(phase_range)))
```

```
    sm = cm.ScalarMappable(norm=norm, cmap=cm.cool)
    sm.set_clim(0, 2*np.pi)
    cbar = plt.colorbar(sm, ax=ax, orientation="horizontal")
    cbar.set_ticks([0, np.pi/2, np.pi, 3*np.pi/2, 2*np.pi])
    cbar.set_ticklabels(["0", r"$\pi/2$", r"$\pi$", r"$3\pi/2$", r"$2\pi$"])
    cbar.set_label("Phase")

fig, (ax1, ax2) = plt.subplots(1, 2, subplot_kw={"projection": "3d"})
fig.tight_layout()

# add each of the graphs
wavefunction_component_ax3d(ax1, ptype="real")
wavefunction_component_ax3d(ax2, ptype="imaginary")

# colorbar
create_cbar(ax1)
create_cbar(ax2)

# plot results
plt.show()
```

Let us now find a *physical interpretation* of k and ω. We note by dimensional analysis, in SI units, k must have units of inverse meters. There is one quantity we know of that has inverse meters, from our study of electromagnetic waves - the *wavenumber*, where $k = \frac{2\pi}{\lambda}$. This suggests that surprisingly, quantum particles have some sort of "wavelength", a characteristic feature of a wave. This wavelength is called the **de Broglie wavelength**, and means that like waves, quantum particles can travel around obstacles, interefere and diffract, and even pass right through each other. This is why the Schrödinger equation is considered a *wave equation*, and as a result, quantum particles are not localized in space before measurement. Even more bizarrely, the de Broglie wavelength gives rise to an equation for a quantum particle's momentum:

$$p = \frac{h}{\lambda} = \hbar k \tag{859}$$

which means that a quantum particle can have momentum *even if it does not have mass*. Photons, for instance, are massless particles, yet they can exert momentum (radiation pressure) on objects.

Let us now return to the other constant factor, ω. By similar dimensional analysis, we note that ω must have units of inverse seconds. This suggests some form of *frequency*. In fact, it is the *angular frequency*. Now, recall we previously derived that $\hbar\omega = \frac{\hbar^2 k^2}{2m}$. Using the substitution that $p = \hbar k$, we can rewrite this as:

$$\hbar\omega = \frac{\hbar^2 k^2}{2m} = \frac{p^2}{2m} = \frac{(mv)^2}{2m} = \frac{1}{2}mv^2 = E \tag{860}$$

Using the special case of photons, we can justify why ω must be the angular frequency. Recall that $E = hf$ expresses the energy carried by a single photon. But we also know that $E = \hbar\omega$ and $\hbar = h/2\pi$. Therefore, equating all our expressions together, we have:

$$E = hf = \hbar\omega = \frac{h}{2\pi}\omega \tag{861}$$

$$hf = \frac{h}{2\pi}\omega \tag{862}$$

$$f = \frac{\omega}{2\pi} \Rightarrow \omega = 2\pi f \tag{863}$$

And therefore we have shown that ω must be the angular frequency. Note that the formulas we found unintentionally while solving the Schrödinger equation, $p = \hbar k$ and $E = \hbar\omega$, show that there is a fundamental connection between **momentum and** k and similarly between **energy and** ω. This is, in fact, a relationship that will hold even when the Schrödinger equation itself becomes inaccurate in certain physical scenarios.

Now, there is *one* particular issue in our solution to the Schrödinger equation: it's physically impossible. Recall that the *normalization condition*, which guarantees that a particle is at least *somewhere* in space, requires that:

$$\int_{-\infty}^{\infty} |\Psi(x,t)^2| dx = 1 \tag{864}$$

But if we attempt to integrate our plane-wave solution $\Psi(x,t) = e^{i(kx-\omega t)}$ we will find that this condition is **not** satisfied:

$$\int_{-\infty}^{\infty} |\Psi(x,t)^2| dx = \int_{-\infty}^{\infty} \Psi(x,t)\Psi^*(x,t)\, dx \tag{865}$$

$$= \int_{-\infty}^{\infty} e^{i(kx-\omega t)} e^{-i(kx-\omega t)} dx \tag{866}$$

$$= \int_{-\infty}^{\infty} dx \tag{867}$$

$$= \infty \tag{868}$$

Therefore, the particle would have infinite probability of being somewhere in space - nonsensical! However, it turns out that there *is* a way to make sense out of a plane-wave solution. If we instead take a *sum* of plane waves of different wavelengths (and therefore a unique k for each wave), then certain portions of each wave will cancel out with the others and certain portions will combine together - that is, the waves would interfere. Therefore, a solution in the form:

$$\Psi(x,t) = \sum_n e^{i(k_n x-\omega t)} \tag{869}$$

does in fact have a finite probability of being somewhere in space. In the continuum limit where we sum over an infinite number of plane waves (and therefore an infinite number of different k's) the sum becomes an integral, so we have:

$$\Psi(x,t) = \frac{1}{\sqrt{2\pi}} \int_{\infty}^{\infty} A(k) e^{i(kx-\omega t)} dx \tag{870}$$

This is known as a **wave packet solution** for a free (quantum) particle and in fact is perfectly normalizable - the factor of $\frac{1}{\sqrt{2\pi}}$ is needed to satisfy the normalization condition (i.e. that the total probability of the particle existing *something* in space is 100%). The function $A(k)$ gives the distribution of k (and therefore the distribution of momenta) for $\Psi(x,t)$. The exact form of $A(k)$ is dependent on the physical scenario we are considering, but one very common choice of $A(k)$ that is widely-applicable is the **Gaussian**:

$$A(k) = \sqrt{\sigma} \left(\frac{2}{\pi}\right)^{1/4} e^{-\sigma^2(k-k_0)^2} \tag{871}$$

For which $\Psi(x,t)$ becomes, after evaluation of the integral:

$$\psi(x,t) = \left(\frac{1}{2\pi}\right)^{1/4} \frac{1}{\sqrt{\sigma}} e^{ik_0 x - \frac{x^2}{4\sigma^2}} e^{-i\omega t} \tag{872}$$

Solving by separation of variables Recall that since the Schrödinger equation is separable, we may use the technique of the **separation of variables** to make it solvable. To do so, we first assume a solution in the form $\Psi(\mathbf{r},t) = \psi(\mathbf{r})T(t)$, where $\psi(\mathbf{r})$ is the purely-spatial component, and $T(t)$ is the purely-temporal (time) component. Then, we can take the derivatives to have:

$$\frac{\partial\Psi}{\partial t} = \frac{\partial}{\partial t}\psi(\mathbf{r})T(t) = \frac{dT}{dt}\psi(\mathbf{r}) \tag{873}$$

$$\nabla\Psi = \nabla[\psi(\mathbf{r})T(t)] = T(t)\nabla\psi(\mathbf{r}) \tag{874}$$

$$\nabla^2\Psi = \nabla(\nabla[\psi(\mathbf{r})T(t)]) = T(t)\nabla^2\psi(\mathbf{r}) \tag{875}$$

$$\tag{876}$$

> **Note**
> As $T(t)$ depends on only one variable (t) the partial derivative becomes an ordinary derivative.

Plugging these back into the Schrödinger equation we have:

$$i\hbar\frac{\partial}{\partial t}\Psi = \left(-\frac{\hbar^2}{2m}\nabla^2 + V(\mathbf{r})\right)\Psi \tag{877}$$

$$i\hbar\frac{dT}{dt}\psi(\mathbf{r}) = -\frac{\hbar^2}{2m}T(t)\nabla^2\psi + V(\mathbf{r})\psi(\mathbf{r})T(t) \tag{878}$$

Dividing both sides by $\psi(\mathbf{r})T(t)$ we have:

$$\frac{1}{\psi(\mathbf{r})T(t)}i\hbar\frac{dT}{dt}\psi(\mathbf{r}) = \frac{1}{\psi(\mathbf{r})T(t)}\left[-\frac{\hbar^2}{2m}T(t)\nabla^2\psi + V(\mathbf{r})\psi(\mathbf{r})T(t)\right] \tag{879}$$

$$\frac{i\hbar}{T(t)}\frac{dT}{dt} = \frac{1}{\psi(\mathbf{r})}\left[-\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r})\psi(\mathbf{r})\right] \tag{880}$$

Remember that just as previously when we did separation of variables, the left and right-hand sides of the last equation are equal to a constant as two derivatives can only be equal in value if they are equal to a constant. We will call this constant E, and by dimensional analysis we can find that this is indeed equal to the *total energy* of the particle (there is a more elegant argument for why we may interpret the separation constant E as the energy, that we will cover later). Thus, we are able to simplify the Schrödinger equation into two simpler equations, one only depend on space and one only dependent on time:

$$\frac{i\hbar}{T(t)}\frac{dT}{dt} = \frac{1}{\psi(\mathbf{r})}\left[-\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r})\psi(\mathbf{r})\right] = E \tag{881}$$

$$\frac{i\hbar}{T(t)}\frac{dT}{dt} = E \quad \Rightarrow \quad i\hbar\frac{dT}{dt} = E\,T(t) \tag{882}$$

$$\frac{1}{\psi(\mathbf{r})}\left[-\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r})\psi(\mathbf{r})\right] = E \tag{883}$$

$$\Rightarrow -\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r})\psi(\mathbf{r}) = E\psi(\mathbf{r}) \tag{884}$$

The differential equation in time, $i\hbar\dfrac{dT}{dt} = E\,T(t)$, is an equation that we are experienced in solving. Its solution is a complex exponential:

$$T(t) = e^{-iEt/\hbar} \tag{885}$$

Which we can verify by substituting by into the differential equation $T(t)$. Note that as we found that $E = \hbar\omega$, then we may rearrange to find that $\frac{E}{\hbar} = \omega$, and thus it is also acceptable to write:

$$T(t) = e^{-i\omega t} \tag{886}$$

Which is a form we will also use in the following sections. The differential equation in time, as we have seen, is straightforward to solve and yields a complex exponential as its solution.

The differential equation in space, however, has much richer and more complex solutions. In fact, this is the equation we primarily focus on when we say we are "solving the Schrödinger equation". It is so crucial that it has its own name: the **time-independent Schrödinger equation**:

$$-\frac{\hbar^2}{2m}\nabla^2\psi + V(\mathbf{r})\psi(\mathbf{r}) = E\psi(\mathbf{r}) \tag{887}$$

The exact solution $\psi(\mathbf{r})$ to the time-independent Schrödinger equation depends on the physics of the situation to analyze. After solving the time-independent Schrödinger equation, recalling that $\Psi(\mathbf{r}, t) = \psi(\mathbf{r})T(t)$, we find that the time-dependent wavefunction is given by:

$$\Psi(\mathbf{r}, t) = \psi(\mathbf{r})e^{-iEt/\hbar} \tag{888}$$

But if ψ_1 is a solution to the time-independent Schrödinger equation, then by linearly, so is ψ_2, and so is $\psi_3, \psi_4, \psi_5, \ldots, \psi_n$, and each solution ψ_n has a corresponding energy E_n. Therefore, the most general solution is found by summing over all ψ, with constant-valued coefficients c_n for each ψ_n:

$$\Psi(\mathbf{r}, t) = \sum_n c_n\psi_n(\mathbf{r})T_n(t) = \sum_i c_n\psi_n(\mathbf{r})e^{-iE_nt/\hbar} \tag{889}$$

This is not simply a mathematical trick; the **physical significance** of the general solution as a sum of individual solutions ψ_n is that each quantum system, which has a wavefunction $\Psi(\mathbf{r}, t)$, is composed of *states* ψ_n. The wavefunction is such that the system has some probabiliy $P_n = |c_n|^2$ of being in the nth-state. The particle can be in *any one* of the states, but we can only predict probabilities it is in a particular state; we cannot predict *which exact state* it is in.

The theoretical approach to solving for the wavefunction, it seems, is not too hard: simply solve the time-independent Schrödinger equation for all the states $\psi_n(\mathbf{r})$, add a time factor $e^{-iE_nt/\hbar}$, and then write the general solution as a sum over all the states with appropriate coefficients c_n, which we can find using the requirement that:

$$\sum_n |c_n|^2 = |c_1|^2 + |c_2|^2 + |c_3|^2 + \cdots + |c_n|^2 = 1 \tag{890}$$

However, the simplicty is deceptive; as with most differential equations, getting to an exact solution is not easy, and sometimes impossible. Thus, for real-life applications (where we cannot assume idealized theoretical systems), Schrödinger's equation is usually solved numerically, using the finite difference method, the finite volume method, or the finite element method, which we will discuss later.

Observables, operators, and eigenvalue problems Just like how we explored the quantization of energy for photons, other measurable quantities, such as the energy and momentum of quantum particles, can also be quantized. In fact, we just derived two cases of quantized quantities: $E = \hbar\omega_0$ and $p = \hbar k$. In the general case, we call measurable quantities **observables** in quantum mechanics. Observables have defined real values, but those values are discrete values, not continuous functions. Therefore, since observables are discrete numbers, then momentum, energy, and angular momentum, as well as other properties cannot be represented by functions, as functions have continuous outputs.

But we know of one different mathematical representation that can work to give discrete numbers - *eigenfunctions and eigenvalues*. Indeed, in quantum mechanics, any observable A has an associated *operator* $\hat{A}$. The operator follows an eigenvalue equation, such that:

$$\hat{A}\psi = a\psi \tag{891}$$

Here, $\hat{A}$ is a linear operator acting on ψ. The most common type of linear operator for eigenfunctions are derivative operators. For instance, the momentum operator (in one dimension) is given by:

$$\hat{p} = -i\hbar\frac{\partial}{\partial x} \tag{892}$$

So if we wanted to find the measured value of the momentum p, we would need to solve the *eigenvalue equation*:

$$\hat{p}\psi = p\psi \tag{893}$$

By substitution of the explicit form of the momentum operator we have:

$$-i\hbar\frac{\partial\psi}{\partial x} = p\psi \tag{894}$$

This is a differential equation with the solution $\psi(x) = C_1 e^{ipx/\hbar}$. But you may think, that's just a plane wave - which we know is unphysical! Indeed, that is true, but recalling our trick with the free particle, we may sum infinite numbers of plane waves together such that we have a normalizable solution, which becomes an integral in the continuous limit:

$$\psi(x) = \frac{1}{\sqrt{2\pi\hbar}}\int_{\infty}^{\infty} B(p)e^{ipx}dp \tag{895}$$

Where $B(p)$ is the distribution of momenta. This ensures that the observable - in this case, the measured value of momentum - stays a discrete number, rather than a continuous function, satisfying the eigenvalue equation for the momentum operator, while $\psi(x)$ also satisfies normalizablity.

From the momentum operator, we may derive the **kinetic energy operator**. Recall that in classical mechanics, the kinetic energy is defined by $K = \frac{1}{2}mv^2 = \frac{p^2}{2m}$. In an analogous fashion, in quantum mechanics, the kinetic energy can be found from the momentum *operator* by the same general formula:

$$\hat{K} = \frac{\hat{p}^2}{2m} = \frac{\hat{p}\cdot\hat{p}}{2m} = \frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} \tag{896}$$

What about the potential energy? Remember that the potential energy in classical physics does *not* depend on the velocity of a particle or time; it depends only on its position. Therefore, the potential energy operator remains unchanged in quantum mechanics, and we have:

$$\hat{V} = V(x) \tag{897}$$

So, the eigenfunction-eigenvalue equations for kinetic energy and potential energy are given by:

$$\hat{K}\psi = -\frac{\hbar^2}{2m}\frac{\partial^2\psi}{\partial x^2} = K\psi \tag{898}$$

$$\hat{V}\psi = V(x)\psi \tag{899}$$

But what about *continuous distributions* of momentum and energy, you may ask? Why does the potential energy operator, for instance, have a *function* rather than a discrete eigenvalue. The answer is that these distributions still have discrete and quantized eigenvalues, just infinitely many of them.

Now, let us take another look at the kinetic energy and potential energy operators:

$$\hat{K} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} \tag{900}$$

$$\hat{V} = V(x) \tag{901}$$

But remember that the *total* energy, which is represented by the Hamiltonian operator, is given by:

$$\hat{H} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x) = \hat{K} + \hat{V} \tag{902}$$

This is simply the conservation of energy! Notice the energy operator is the sum of the kinetic energy operator and the potential energy. But since we have $(\hat{K} + \hat{V})\psi = \hat{H}\psi$ for the operators, then we *also* have $(K + V)\psi = E\psi$ for the eigenvalues. Combining these together we get:

$$\hat{H}\psi = E\psi \tag{903}$$

This is the **time-independent Schrödinger equation**. It offers a crucial insight into the nature of quantum systems: to find the *possible states* of a quantum system, we need to find the eigenfunctions of the Hamiltonian, and knowing the possible states also gives us the possible *energies* of the system (which are often quantized). This is a very important insight as it generalizes to more advanced quantum mechanics *and* the research that we do.

The uncertainty principle Knowing more about a particular observable in general means that we know less about another observable. This is the **uncertainy principle** - when we express the uncertainty of position and momentum in terms of their standard deviations σ_x, σ_p, then:

$$\sigma_x \sigma_p \geq \frac{\hbar}{2} \tag{904}$$

To derive this, let's consider the position and momentum operators. They are given respectively by:

$$\hat{x} = x, \quad \hat{p} = -i\hbar\frac{\partial}{\partial x} \tag{905}$$

Now, it is important to recognize that these two operators do not commute - if we switch the order we apply these operations, we get different results. That is:

$$\hat{x}\hat{p}\psi \neq \hat{p}\hat{x}\psi \tag{906}$$

We call the difference between applying these two operators in either order the **commutator**, denoted by square brackets:

$$[\hat{x}, \hat{p}] = \hat{x}\hat{p} - \hat{p}\hat{x} \tag{907}$$

If we compute this commutator by substituting the operators in, we get:

$$[\hat{x}, \hat{p}] = i\hbar \tag{908}$$

We can now make use of a theorem from statistics, which says that:

$$\sigma_a \sigma_b \geq \frac{1}{2i}[\hat{a}, \hat{b}] \tag{909}$$

Plugging in our result for the commutator, we get the **Heisenberg uncertainty principle**:

$$\sigma_x \sigma_p \geq \frac{\hbar}{2} \tag{910}$$

The Heisenberg uncertainty principle places further restrictions on the limits of observational and experimental precision, meaning that position and momentum cannot *both* be perfectly known. When measuring the **position** of the particle, the momentum is *not* known precisely as there is uncertainty in momentum, meaning that the particle's momentum could be fluctuate within the bounds of the uncertainty; similarly, when measuring the **momentum** of a particle, the position is *not* known precisely as there is uncertainty in position, meaning that the particle's position could be anywhere within the bounds of the uncertainty in position. Further, when position is perfectly known, the

uncertainty in momentum is infinite, and so the momentum is not known at all! The same goes for momentum - when it is perfectly known, the uncertainty in position is infinite, meaning that the particle could be anywhere!

Quantum mechanics, Part 2

Building on what we've learned in the first part of quantum mechanics, we can now explore more complex concepts in quantum theory. In this second part, we will cover the theory that allows us to do more advanced calculations in quantum mechanics.

Introduction to matrix mechanics In studying complex multi-state quantum systems, numerical methods are often the only way to solve a variety of problems, as solving the Schrödinger equation by hand becomes impossible. One important feature of these numerical methods is utilizing the Heisenberg picture of quantum mechanics, also known as **matrix mechanics**.

In matrix mechanics, we describe the system not through its total wavefunction, but by its **operators**. The quantum state $|\Psi\rangle$ stays constant; the **operators** (energy, momentum, etc.) are what evolve through time. In particular, the operators can be expressed in specific *bases* (bases are plural of "basis"). For discrete operators, as we have seen for the spin matrices, we can choose a discrete basis. In our case, the $\hat{S}_z$ operator can be expressed using the basis of the two spin states, as given by:

$$\chi_{z^+} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad \chi_{z^-} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{911}$$

Meanwhile, for continuous (differential) operators, we can choose a functional basis. For instance, we can use the Legendre, Laguerre, and Hermite polynomials, all of which form a complete basis set, to find the **matrix representation** of the operator. Within this section, we will give a brief introduction on determining the matrix representation of various operators, based on the book *Basic Theory of Lasers and Masers* by Jacques Vanier.

The Hamiltonian in general bases In the time-independent regime, the state of a quantum system is represented by its state-vector $|\Psi\rangle$, which can be expanded into a set of eigenstates $|\psi_n\rangle$ by $|\Psi\rangle = \sum_n |\psi_n\rangle$. Each eignestate satisfies eigenvalue equation $\hat{H}|\psi_n\rangle = E_n|\psi_n\rangle$, where $\hat{H}$ is the Hamiltonian and E_n is the energy eigenvalue of a particular eigenstate. We may extract the energy eigenvalues E_n as follows. First, we multiply both sides by a bra $\langle\psi_m|$:

$$\langle\psi_m|\hat{H}|\psi_n\rangle = \langle\psi_m|E_n|\psi_n\rangle \tag{912}$$

Now, the eigenstates $|\psi_n\rangle$ can theoretically be expressed in *any* basis, but we typically choose to use a **complete orthonormal basis**. Such bases include Fourier series as well as the Legendre, Laguerre, and Hermite polynomials. The specific basis doesn't matter; it simply matters that a complete and orthonormal basis satisfies $\langle\psi_m|\psi_n\rangle = \delta_{mn}$. Therefore, we have $\langle\psi_m|E_n|\psi_n\rangle = E_n\langle\psi_m|\psi_n\rangle$ (since E_n is a constant and can be factored out of the expression). Given that we have an orthonormal basis, $E_n\langle\psi_m|\psi_n\rangle$ is only nonzero when $m = n$, in which case we have $E_n\langle\psi_n|\psi_n\rangle = E_n$, and thus:

$$\begin{aligned}
\langle\psi_m|\hat{H}|\psi_n\rangle &= \langle\psi_m|E_n|\psi_n\rangle && (913)\\
&= E_n\langle\psi_m|\psi_n\rangle && (914)\\
&= E_n\,\delta_{mn} && (915)\\
\langle\psi_m|\hat{H}|\psi_n\rangle &= E_n\delta_{mn} \quad \text{(zero if } m \neq n\text{)} && (916)\\
\langle\psi_n|\hat{H}|\psi_n\rangle &= E_n && (917)\\
& && (918)
\end{aligned}$$

Thus, to find E_n, we "only" need to calculate $\langle\psi_n|\hat{H}|\psi_n\rangle$, which can also be written in terms of the Schrödinger formalism as:

$$E_n = \int_{-\infty}^{\infty} \psi_n^*(x)\hat{H}\psi_n(x)\,dx \tag{919}$$

But returning to the matrix representation - we may write the set of eigenvalues E_n with the following matrix equation:

$$\begin{pmatrix} \langle\psi_1|\hat{H}|\psi_1\rangle & 0 & 0 & \dots & 0 \\ 0 & \langle\psi_2|\hat{H}|\psi_2\rangle & 0 & \dots & 0 \\ 0 & 0 & \langle\psi_3|\hat{H}|\psi_3\rangle & \dots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \dots & \langle\psi_n|\hat{H}|\psi_n\rangle \end{pmatrix} = \begin{pmatrix} E_1 & 0 & 0 & \dots & 0 \\ 0 & E_2 & 0 & \dots & 0 \\ 0 & 0 & E_3 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \dots & E_n \end{pmatrix} \tag{920}$$

This is a diagonal matrix - so one might wonder, why bother writing this when each side can just be written as a vector multiplied by an identity matrix? The reason is that if we *don't* know the eigenstates, we can always choose to expand our quantum state $|\Psi\rangle$ in some other complete and orthonormal basis instead, which we will refer to as $|\phi_n\rangle$ (since we can always choose any basis to write out our state-vector). As both bases are complete and orthogonal, we can then express a state in our new basis $|\phi_n\rangle$ in terms of our eigenstate basis $|\psi_n\rangle$ as a linear sum:

$$|\phi_n\rangle = \sum_k a_{nk}|\psi_k\rangle \tag{921}$$

$$= a_{n1}|\psi_1\rangle + a_{n2}|\psi_2\rangle + a_{n3}|\psi_3\rangle + \cdots + a_{nk}|\psi_k\rangle \tag{922}$$

Where all the a_{nk}'s are constant coefficients. We can write this in matrix form as:

$$\underbrace{\begin{pmatrix} |\phi_1\rangle \\ |\phi_2\rangle \\ |\phi_3\rangle \\ \vdots \\ |\phi_n\rangle \end{pmatrix}}_{\text{new basis}} = \underbrace{\begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \dots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \dots & a_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nk} \end{pmatrix}}_{\text{matrix } A_{nk}} \underbrace{\begin{pmatrix} |\psi_1\rangle \\ |\psi_2\rangle \\ |\psi_3\rangle \\ \vdots \\ |\psi_k\rangle \end{pmatrix}}_{\text{old basis}} \tag{923}$$

We can extract out the components of A_{nk} using another orthogonality argument. If we take the components of our new basis $|\phi_n\rangle$, then multiply by a bra $\langle\psi_j|$, then we have:

$$|\phi_n\rangle = \sum_k A_{nk}|\psi_k\rangle \tag{924}$$

$$|\phi_n\rangle\langle\psi_j| = \langle\psi_j|\sum_k A_{nk}|\psi_k\rangle \tag{925}$$

$$= \sum_k A_{nk}\langle\psi_j|\psi_k\rangle \tag{926}$$

$$= A_{nk}\delta_{jk} \tag{927}$$

Where $\langle\psi_j|\psi_k\rangle = \delta_{jk}$ since our bases are othogonal. The only case were $\langle\psi_j|\phi_n\rangle = A_{nk}\delta_{jk}$ does *not* vanish is when $j = k$ and therefore we have:

$$|\phi_n\rangle\langle\psi_k| = A_{nk}\delta_{kk} = A_{nk} \tag{928}$$

Thus we have $A_{nk} = |\phi_n\rangle\langle\psi_k|$. We may also derive an expression for the *inverse*, i.e. A^{-1}. Recall that the inverse of A must satisfy $AA^{-1} = I$ where I is the identity matrix. We recall that the identity matrix is given by $I = \sum_n |\phi_n\rangle\langle\phi_n|$. Then, we have:

$$AA^{-1} = I \tag{929}$$

$$|\phi_n\rangle\langle\psi_k|A^{-1} = \sum_i |\phi_i\rangle\langle\phi_i| \tag{930}$$

$$\tag{931}$$

Now, if we take the inner product of both sides with the ket $|\psi_k\rangle$, then:

$$|\psi_k\rangle\Big(|\phi_n\rangle\langle\psi_k|\Big)A^{-1} = |\psi_k\rangle\sum_i |\phi_i\rangle\langle\phi_i| \tag{932}$$

$$\Big(|\phi_n\rangle\langle\psi_k|\Big)|\psi_k\rangle A^{-1} = \left(\sum_i |\phi_i\rangle\langle\phi_i|\right)|\psi_k\rangle \tag{933}$$

$$|\phi_n\rangle\langle\psi_k|\psi_k\rangle A^{-1} = \sum_i |\phi_i\rangle\langle\phi_i|\psi_k\rangle \tag{934}$$

$$|\phi_n\rangle\langle\psi_k|\psi_k\rangle A^{-1} = \sum_i |\phi_i\rangle\langle\phi_i|\psi_k\rangle \tag{935}$$

$$|\phi_n\rangle A^{-1} = \sum_i |\phi_i\rangle\langle\phi_i|\psi_k\rangle \tag{936}$$

$$\tag{937}$$

Where we were able to swap the order of the inner product since the inner product is commutative, and we were able to pull the $|\psi_k\rangle$ into the sum since the sum does not sum over the index k (so $|\psi_k\rangle$ can essentially be treated as a constant in the sum). Now if we take the inner product with the bra $\langle\phi_n|$ we have:

$$\langle\phi_n|\phi_n\rangle A^{-1} = \langle\phi_n|\sum_i |\phi_i\rangle\langle\phi_i|\psi_k\rangle \tag{938}$$

$$\langle\phi_n|\phi_n\rangle A^{-1} = |\psi_k\rangle \underbrace{\langle\phi_n|\sum_i |\phi_i\rangle\langle\phi_i|}_{=\delta_{ni}} \tag{939}$$

$$A^{-1} = |\psi_k\rangle\delta_{ni}\langle\phi_i\rangle \tag{940}$$

$$A^{-1} = |\psi_k\rangle\langle\phi_n| \tag{941}$$

Where again, since $|\psi_k\rangle$ is not summed over, we were able to pull it out of the sum, and we used orthogonality to collapse the sum into a single term.

Having computed A and A^{-1}, we will now show that our transformation of basis actually leads to a very nice expression for the matrix representation of the Hamiltonian. The Hamiltonian in our new basis, which we write as $\hat{H}$, is given by $\hat{H} = A\hat{H}A^{-1}$ (this is a standard result of linear algebra). If we then substitute our expressions for A and A^{-1}, we have:

$$\hat{H} = A\hat{H}A^{-1} \tag{942}$$

$$= |\phi_n\rangle\langle\psi_k|\hat{H}|\psi_k\rangle\langle\phi_n| \tag{943}$$

$$|\phi_n\rangle\hat{H} = |\phi_n\rangle\Big(|\phi_n\rangle\langle\psi_k|\hat{H}|\psi_k\rangle\langle\phi_n|\Big) \tag{944}$$

$$= \Big(|\phi_n\rangle\langle\psi_k|\hat{H}|\psi_k\rangle\langle\phi_n|\Big)|\phi_n\rangle \tag{945}$$

$$= |\phi_n\rangle\langle\psi_k|\hat{H}|\psi_k\rangle\langle\phi_n|\phi_n\rangle \tag{946}$$

$$\langle\phi_n|\phi_n\rangle\hat{H} = \langle\phi_n|\phi_n\rangle\langle\psi_k|\hat{H}|\psi_k\rangle \tag{947}$$

$$\langle\phi_n|\phi_n\rangle\hat{H} = \langle\phi_n|\phi_n\rangle\langle\psi_k|\hat{H}|\psi_k\rangle \tag{948}$$

$$\hat{H} = \langle\psi_k|\hat{H}|\psi_k\rangle \tag{949}$$

$$= E_k \tag{950}$$

Important
Here, E_k is shorthand notation for $E_k \cdot I$, since it represents a matrix with the energy eigenvalues along its diagonal and zero everywhere else.

Where the last step comes from $\langle\psi_n|\hat{H}|\psi_n\rangle = E_n$ (the index does not matter at this point because we use only one index in the expression). Thus we find that:

$$\hat{H} = A\hat{H}A^{-1} = E_n \tag{951}$$

Let us demonstrate the same with, in the functional picture. Consider a system described by wavefunction $\psi(\mathbf{r})$. By the postulates of quantum mechanics, the wavefunction may be expanded in any complete and orthonormal set of functions $\phi_n(\mathbf{r})$, such that:

$$\psi(\mathbf{r}) = \sum_{n=1}^{\infty} c_n\phi_n(\mathbf{r}) = c_1\phi_1 + c_2\phi_2 + c_3\phi_3 + \cdots + c_n\phi_n \tag{952}$$

Where we assume all ϕ_n are normalized. From here, we may prepare a matrix A_{mn} with elements given by:

$$A_{mn} = \int_{-\infty}^{\infty} \phi_m^* \hat{H}\phi_n dV \tag{953}$$

Now, A_{mn} is not necessarily a diagonal matrix, so it does not necessarily give the energy eigenvalues. However, if we diagonalize it, we are left with the matrix E_{mn}, which *is* diagonal (that is, for all $m \neq n$, $E_{mn} = 0$, and whose diagonals $E_{nn} = E_n$ are equal to the energy eigenvalues.

The matrix operator approach therefore condenses the difficult problem of solving the Schrödinger equation into a more straightforward problem of finding the correct matrix A that diagonalizes the Hamiltonian in a particular basis; from there, we can simply read off the energy eigenvalues from the diagonal.

Example: quantum harmonic oscillator We will use the matrix representation approach to solve for the quantum harmonic oscillator, which will act as a toy model. We want to obtain a matrix representation for the Hamiltonian of the quantum harmonic oscillator and find the energy eigenvalues.

The well-known Hamiltonian for the quantum harmonic oscillator is given by:

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{x}^2 \tag{954}$$

We will now need to pick a basis to be able to obtain its matrix representation. In theory, when we don't know the precise eigenstates of the matrix, any set of basis functions would do (as long as they are complete and orthogonal) - but luckily for us, we already know the eigenstates of the Hamiltonian. So for demonstrative purposes, it is easiest to choose the basis of the eigenstates of the Hamiltonian, which we can write as $|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle, \ldots, |\psi_n\rangle$. Then, we have $E_n = \langle\psi_n|\hat{H}|\psi_n\rangle$. But recall that in the example of the quantum harmonic oscillator, $\hat{H}|\psi_n\rangle = \hbar\omega\left(n + \frac{1}{2}\right)|\psi_n\rangle$. Thus, $\langle\psi_n|\hat{H}|\psi_n\rangle$ is given by:

$$\langle\psi_n|\hat{H}|\psi_n\rangle = \langle\psi_n|\hbar\omega\left(n + \frac{1}{2}\right)|\psi_n\rangle \tag{955}$$

$$= \hbar\omega\left(n + \frac{1}{2}\right)\langle\psi_n|\psi_n\rangle \tag{956}$$

$$= \hbar\omega\left(n + \frac{1}{2}\right)\delta_{nn} \tag{957}$$

Thus our resulting energy matrix becomes[2]:

[2] https://quantummechanics.ucsd.edu/ph130a/130_notes/node258.html

$$\langle\psi_n|\hat{H}|\psi_n\rangle = \hbar\omega \begin{pmatrix} \frac{1}{2} & 0 & 0 & \dots & 0 \\ 0 & \frac{3}{2} & 0 & \dots & 0 \\ 0 & 0 & \frac{5}{2} & \dots & 0 \\ 0 & 0 & 0 & \ddots & \vdots \\ 0 & 0 & 0 & \dots & (n+\frac{1}{2}) \end{pmatrix} \tag{958}$$

And thus our energy eigenvalues can be found (as expected) by just reading off the diagonals the diagonals, from which we can surmise that:

$$E_n = \left(n + \frac{1}{2}\right)\hbar\omega \tag{959}$$

Let us now show that even though this matrix representation method for the quantum harmonic oscillator is a "toy model", it can still predict real-world results. The transition energy ΔE from our result is the difference between energy levels n_1, n_2, and thus:

$$\Delta E = \left(n_2 + \frac{1}{2}\right)\hbar\omega - \left(n_1 + \frac{1}{2}\right)\hbar\omega \tag{960}$$

$$= (n_2 - n_1)\hbar\omega \tag{961}$$

From which we may derive the spectral lines for various different transitions, given that $\Delta E = hc/\lambda$, with:

$$\frac{1}{\lambda} = \frac{\omega}{2\pi c}(n_2 - n_1) \quad \Rightarrow \quad \lambda = \frac{2\pi c}{(n_2 - n_1)\omega} \tag{962}$$

The angular frequency of the vibrations, ω, can be found through $\omega = \sqrt{k/\mu}$ where μ is the reduced mass of the molecule and k is its force constant. Let us compute the vibrational transitions of carbon dioxide, which, although not strictly speaking a *diatomic* molecule, can be somewhat treated as such. The reduced mass of the triatomic carbon dioxide molecule is given by[3]:

$$\mu = \frac{2m_C m_O + m_O^2}{2m_C + m_O} \approx 2.567\text{E} - 27\text{kg} \tag{963}$$

Where m_C is the mass of the carbon atom and similarly m_O is the mass of the oxygen atom. The force constant of $CO2$ is approximately 1680N/m^4 and thus $\omega \approx 251.47\text{THz}$. Substituting these values, we find that the spectral lines are given by:

Transition	Spectral line	Spectrum
$\|1\rangle \to \|0\rangle$	7596 nm	Mid-infrared
$\|2\rangle \to \|0\rangle$	3748 nm	Mid-infrared
$\|3\rangle \to \|0\rangle$	2499 nm	Near-infrared
$\|4\rangle \to \|0\rangle$	1874 nm	Near-infrared
$\|5\rangle \to \|0\rangle$	1499 nm	Near-infrared
$\|6\rangle \to \|0\rangle$	1249 nm	Near-infreared

Note
All of these transitions are vibrational transitions, and they all occur in the *infrared spectrum*. This has great physical significance; it means that carbon dioxide (just like other greenhouse gases) is very good at absorbing and re-emitting infrared light, which is (one of) the mechanisms that leads to the greenhouse effect, which in turn drives climate change.

[3] https://www.quora.com/Is-there-any-way-to-calculate-reduced-mass-for-more-than-two-atoms-in-a-molecule

[4] https://chemistry.stackexchange.com/questions/70858/how-do-i-determine-the-molecular-vibrations-of-linear-molecules

Example 2: hydrogen atom We will now produce a computational example to solve for the **hydrogen atom**. We use the normalized Chebyshev polynomials as our basis for $R(r), \Theta(\theta)$, and $\Phi(\phi)$ (that is, each of these is approximated by a particular Chebyshev polynomial). We code this all with python, using SciPy for the special functions.

First, we import out libraries and set up our basis:

```
from scipy.special import legendre
from scipy.differentiate import derivative
from scipy.constants import hbar, c, mp, me, elementary_charge
import numpy as np
```

Then, we prepare our Hamiltonian:

```
# TODO convert to spherical coords
def Hamiltonian(psi, x, y, z, mu=None, eps=1E -5):
 # mu is reduced mass
 if not mu:
  mu = (mp * me)/(mp + me)
 laplacian =
 K = -hbar**2 / (2*mu) * derivative(psi, )
 r2 = x**2 + y**2 + z**2
 # smoothed coulomb potential to avoid divide by zero
 # similar to the Plummer potential
 potential = -k*elementary_charge**2 / np.sqrt(r2 + eps**2)
 return K + potential
```

Example 3: helium atom Let us now consider a case of a system that does not possess an analytic solution - the helium atom. Since a (neutral) helium atom is composed of two electrons around a nucleus, the atomic Hamiltonian (we ignore spin, vibrational, and rotational degrees of freedom) becomes:

$$\hat{H}_{He} = \underbrace{-\frac{\hbar^2}{2m}(\nabla_1^2 + \nabla_2^2)}_{\text{electron kinetic energy}} - \underbrace{\frac{Z}{4\pi\varepsilon_0}\left(\frac{e^2}{r_1} + \frac{e^2}{r_2}\right)}_{\text{nucleus-electron attraction}} + \underbrace{\frac{1}{4\pi\varepsilon_0}\frac{e^2}{|r_2 - r_1|}}_{\text{electron-electron repulsion}} \tag{964}$$

Where here we assumed that the nucleus moves so slowly that we can effectively consider it non-moving (as with the solution to the hydrogen atom). The third term (interaction term between different electrons) means that the solution to the helium atom *cannot* be written as a linear superposition of the solutions of the hydrogen atom.

To be able to solve for the energy levels, since we don't know the exact eigenstates, we will use the eigenstates of the *hydrogen atom* instead as our orthonormal basis. Recall that the hydrogen atom's eigenstates are parametrized by quantum numbers n, ℓ, m and are given by:

$$\psi_{n,\ell,m}(r,\theta,\varphi) = \sqrt{\left(\frac{2}{na_0^*}\right)^3 \frac{(n-\ell-1)!}{2n(n+\ell)!}} \mathrm{e}^{-\rho/2}\rho^\ell L_{n-\ell-1}^{2\ell+1}(\rho) Y_\ell^m(\theta,\varphi) \tag{965}$$

Where a_0^* is the reduced Bohr radius, $\rho = 2r/na_0^*$, $L(\rho)$ is a generalized Laguerre polynomial, and $Y(\theta,\varphi)$ is a spherical harmonic. To be able to find the energy levels requires computing the diagonals of the matrix $\langle\psi_k|\hat{H}|\psi_k\rangle$, which, in this case, becomes:

$$E_k = \langle\psi_k|\hat{H}|\psi_k\rangle = \int_V \psi_{n,\ell,m}\ \hat{H}_{\mathrm{He}}\ \psi_{n,\ell,m}^* dV \tag{966}$$

These integrals are considerably simplified by the orthogonality of the solutions to the hydrogen atom (since they form a complete orthonormal basis).

Quantum mechanics, Part 3

Transition frequencies and spectroscopy One of the first great successes of quantum theory was in its correct prediction of what light is emitted and absorbed by particular atoms (and molecules and crystals). The wavelengths a particular atom (or molecule or crystal) will absorb and emit are known as its **spectra** (the individual wavelengths are called *spectral lines*). The field devoted to the study of spectra is **spectroscopy**. Precise spectra are determined through experimental measurements with specialist instruments. However, high-accuracy calculation of spectra from quantum theory is also possible, although these need experimental verification to check that the spectra are indeed correct. In this and the following sections, we will be focused on the latter and we will discuss methods of calculating spectra for different material species (here, *species* means the same thing as "types").

Calculating spectra from theory, also called *first-principles* or *ab priori* calculations, is a difficult task. Since calculating spectra *at minimum* usually requires solving the Schrödinger equation, which has very few analytical solutions, approximation schemes, numerical methods, and a variety of techniques have grown around trying to solve the Schrödinger equation to find (among other things) the spectra of different material species.

One of the earliest successful theoretical calculations of spectra was that of atomic hydrogen. The **Rydberg formula**, which can derived from the solution to the Schrödinger equation for the hydrogen atom, predicts spectral lines at wavelengths λ for which:

$$\frac{1}{\lambda} = R_H \left(\frac{1}{{n_1}^2} - \frac{1}{{n_2}^2} \right) \tag{967}$$

Where n_1, n_2 are positive integers, $n_1 < n_2$, and R_H is the **Rydberg constant** for hydrogen, given by:

$$R_H = \frac{m_p}{m_e + m_p} \frac{m_e e^4}{8\varepsilon_0^2 h^3 c} \approx 1.097 \times 10^7 \text{ m}^{-1} \tag{968}$$

Where m_p is the proton mass, m_e is the electron mass, e is the elementary charge, ε_0 is the electric constant, h is the reduced Planck constant, and c is the speed of light. A very similar formula exists for the helium ion (*not* helium atom).

While the Rydberg formula does not predict *all* the spectral lines of hydrogen and has been supplemented by more precise calculations, it represented a major step in our ability to predict atomic spectra. However, the Rydberg formula could only be found from theory thanks to the fact that an analytical solution could be found to the hydrogen atom.

Of course, the Rydberg formula, while a historical landmark in theoretical spectra analysis, is a specific result that makes rigorous predictions for the hydrogen atom (and a few other hydrogen-like atoms). One can attempt to apply the Rydberg formula to non-hydrogen atoms, but the predictions are inconsistent; while it works for some spectra lines of some atoms, it doesn't work for others.

The lack of a general formula to derive the spectra lines of arbitrary atoms, much less molecules or crystal solids, means that in general, spectral lines of non-hydrogenic atoms must be calculated on a case-by-case basis. However, we find from both approximate calculations and experimental data that we can categorize the wavelengths of transitions according to certain heuristic rules (*heuristic* here means that the results are non-rigorous but are usually in the right direction). We summarize these rules using the table below:[5][6]

Type of transition	Associated wavelength of emitted light
Electronic (electron transition between atomic orbitals)	Mostly UV or visible, up to infrared
Vibrational (molecular-only)	Mid-Infrared
Rotational (molecular-only)	Far-Infrared or microwaves

[5] http://chemistry.ncssm.edu/labs/DiatomicMoleculeMath.pdf
[6] https://www.damtp.cam.ac.uk/user/tong/aqm/justsix.pdf

There are also a few other "transitions" of note beyond the ones we have listed in the above table. We say "transitions" because most of them are actually conventional types of transitions (such as electronic transitions), but between different energy levels that were not predicted by the Schrödinger model of simple atoms. These "new" energy levels are caused by spin-related and relativistic effects as well as interactions with electric and magnetic fields. Additionally, many (but not all) of these "transitions" result in the emission of microwave radiation (which is far weaker than the UV/visible/infrared light, making them far less detectable). We compile a table of these "transitions" in the following table[7]:

Type of transition	Occurs due to
Electronic, between "new" energy levels caused by relativistic/spin effects	Fine-structure splitting
Electronic, in presence of a magnetic field	Zeeman effect
Electronic, in presence of an electric field	Stark effect
Spin-flip transition, caused by nuclear spin	Hyperfine splitting
Electronic, between "new" energy levels caused by quantum vacuum fluctuations	Lamb shift (both predicted by quantum electrodynamics)

Note
The majority of these results can be found by solving the relativistic **Dirac equation**, which is more accurate than the Schrödinger equation, with a perturbed potential known as the Uehling potential that captures the contribution of quantum electrodynamics. They can also be found through advanced density-functional theory techniques.

Vibrational spectra We have already seen that certain transitions can be approximately modelled by known quantum systems. For instance, vibrational transitions in a diatomic molecule (which roughly come from the chemical bond joining the two atoms in the molecule) can be modelled by the quantum harmonic oscillator, which has a constant spectral line given by:

$$\frac{1}{\lambda} = \frac{1}{2\pi c}\sqrt{\frac{k}{\mu}} \tag{969}$$

Where k is the force constant of the molecular bond (determined experimentally) and μ is the reduced mass of the molecule, given by:

$$\mu = \frac{m_1 m_2}{m_1 + m_2} \tag{970}$$

However, the quantum harmonic oscillator is only a very *idealized* model. Its results are far from general enough or accurate enough to describe the spectral lines of most material species. To be able to compute more accurate results, we need to use more powerful techniques, which we will introduce in time.

Rotational spectra We will now investigate *rotational* transitions in molecules - these are the dominant microwave-producing transitions. For a diatomic molecule composed of two atoms of masses m_1, m_2, with bond length d, the rotational Hamiltonian is given by the **rigid rotor**:

$$\hat{H}_{\text{rot.}} = \frac{\hat{L}^2}{2I}, \quad I = \mu d^2 \tag{971}$$

Where $\mu = \dfrac{m_1 m_2}{m_1 + m_2}$ is the *reduced mass* of the molecule, and I is its *moment of inertia*. The solution yields eigenstates in terms of the spherical harmonics $Y_{j,m}$ and energy eigenvalues:

[7] Degenerate energy levels#Removing degeneracy

$$E_j = \frac{j(j+1)\hbar^2}{2I} = Cj(j+1), \quad C = \frac{\hbar^2}{2I} \tag{972}$$

For reasons we will cover later, rotational transitions must be between energy eigenstates of $\Delta j = \pm 1$, that is, *only* between two adjacent energy levels. To find the transition wavelengths, we must calculate the energy difference ΔE between E_{j+1} and E_j energy levels:

$$E_j = Cj(j+1) \tag{973}$$
$$E_{j+1} = C(j+1)(j+2) \tag{974}$$
$$\Delta E = E_{j+1} - E_j = 2C(j+1) \tag{975}$$
$$\tag{976}$$

Thus, for a rotational transition between energy levels $j' \to j$, where $j' = j+1$, we have:

$$\Delta E = \frac{hc}{\lambda} = 2Cj' \tag{977}$$
$$\frac{1}{\lambda} = \frac{2C}{hc} j' \tag{978}$$
$$= \frac{\hbar^2}{2I}\frac{2}{hc} j' \tag{979}$$
$$= \frac{h^2}{4\pi^2 hIc} j' \tag{980}$$
$$= \frac{h}{4\pi^2 Ic} j', \quad j' = 1, 2, 3, \ldots \tag{981}$$

Where, remember, j' is the total angular momentum quantum number of the *upper* energy level, and j is that of the *lower* energy level. In the literature, the expression for the reciprocal of the wavelength is more commonly expressed in terms of a molecular constant B, defined as:

$$\frac{1}{\lambda} = 2Bj', \quad B = \frac{h}{8\pi^2 cI} \tag{982}$$

Thus our expression gives the *absorption* spectral lines for molecules *excited to* the j'-th state, and the *emission* spectral lines for molecules *decaying from* the j'-th state. Meanwhile, the relative population of the lower level relative to the ground state is given by:

$$\frac{N_l}{N_0} = (2J_l + 1)\exp\left(-\frac{hcBJ_l(J_l+1)}{k_B T}\right) = (2J_l+1)\exp\left(-\frac{h^2 J_l(J_l+1)}{8\pi^2 I k_B T}\right) \tag{983}$$

The general problem and selection rules In the most general case, the energy levels of a system are given by taking the inner product of the system Hamiltonian acting on the system's wavefunction, with the conjugate of the wavefunction:

$$E_{n,\ell,m} = \int_\Omega \psi^*_{n,\ell,m}(\vec{x})\hat{H}\psi_{n,\ell,m}(\vec{x})\, dV \tag{984}$$

However, due to *selection rules*, we find that such integrals often evaluate to zero, meaning that the transition is impossible, so we say that the transition is *forbidden*. There are some cases in which other quantum effects cause a forbidden transition to occur, but we will not discuss them here.

Theoretical and numerical computation of spectra It's important to note that simply calculating the energy levels *isn't* what gives the spectral lines. This is because spectra are the result of *transitions* between different energy levels, not the energy levels themselves, so we need to find the *differences* between energy levels; those differences are what are actually proportional to the wavelength.

Also go through the Hartree-Fock method and its generalization to the Dirac equation (Dirac-Hartree-Fock)

General DFT for spectral calculations: https://adpcollege.ac.in/online/attendence/classnotes/files/1587884742

Relativistic + QED DFT: https://arxiv.org/abs/2102.10465

0.2.8 Fundamentals of lasers

Laser technology is at the core of so much of our modern-day world. They are quintessential, found everywhere from barcode scanners to laser pointers to fusion laboratories. Yet the principles behind their operation are surprisingly non-trivial. In this sequence of chapters, we'll explore how lasers work at the fundamental level; information that will be essential once we discuss Project Elara's laser research.

Laser physics, part 1

As we've covered previously, a key component in our proposed solar power satellite system is the maser (microwave laser) that transmits the captured energy of sunlight back to Earth. Designing a laser to meet the high demands of our use case is an immense engineering challenge, which we hope to accomplish with research breakthroughs and technological innovation. But before examining the specifics of state-of-the-art laser technology, let us first go over the essential theory behind how lasers work.

A review of stimulated and spontaneous emission We encountered and briefly discussed the phenomenon of **stimulated emission** that underlies lasers, but let us review the topic again to gain greater familiarity for the heavy quantum theory that follows.

Remember that **stimulated emission** is one of two modes of light emission (the emission of photons, the quantum particle associated with light). The other, more "conventional" way that atoms emit photons is the process of **spontaneous emission**. This is a three-step process[8]:

1. An atom absorbs a photon, and is excited to a higher-energy state with energy E_2
2. The atom then decays to a lower-energy state, which has energy E_1.
3. A new photon is released in the process, with energy $E_{\text{photon}} = E_2 - E_1$ and wavelength $\lambda = hc/E_{\text{photon}}$

Spontaneous emission is how most light (which includes UV, infrared, microwave, radio wave, etc.) in the universe is produced, from starlight to incandescent lightbulbs[9]. It has several important characteristics:

- The decay from the upper state to the lower state happens *spontaneously*, without anything to trigger it.[10]
- Although the decay rate (probability of a decay) *can* be predicted, *when* exactly the decay occurs is **random**.
- The photons that are emitted from the decay travel in random directions and their polarization, frequency, and wavelength cannot be predicted in advance

Spontaneous emission is not helpful for building a laser, because there is no way to pre-determine the characteristics of the photon to be emitted. Thus, spontaneously-emitted light is not **monochromatic** (monochromatic means that the light is of *only one frequency*), but rather, spread across a wide range of frequencies, and the emitted photons travel away in random directions[11], meaning that the light is not **collimated** and cannot form a tight beam. These two undesirable results defeat the point of a laser, where we want to produce light of a *single frequency* in a tightly-focused beam.

But there exists an *alternative* means of light emission, known as **stimulated emission**. Stimulated emission is also a multi-step process, but with different steps:

1. An atom is excited by some external energy source (this can be in the form of electric, electromagnetic/light, chemical, thermal or even nuclear energy). The energy source raises the atom into a higher-energy state with energy $E_2 = E_1 + \Delta E$.
2. The already-excited atom absorbs a photon that *also* has energy ΔE
3. The atom then decays down to a lower-energy state, which has energy E_1.
4. *Two* identical photons are released in the process, each with energy $E_{\text{photon}} = E_2 - E_1 = \Delta E$ and thus wavelength $\lambda = hc/E_{\text{photon}}$

[8] https://www.rp-photonics.com/spontaneous_emission.html

[9] https://ecampus.matc.edu/mihalj/scitech/unit5/incandescence/incandescence.htm

[10] Technically speaking, the emission of a photon from the state transition (decay from the upper state to a lower state) *doesn't* happen completely on its own. The quantum electrodynamical vacuum is what mediates the transition and thereby the release of a photon, but that is an advanced topic we'll cover in the expert guide

[11] https://physics.stackexchange.com/questions/338038/why-laser-is-a-collimated-parallel-beam?rq=1

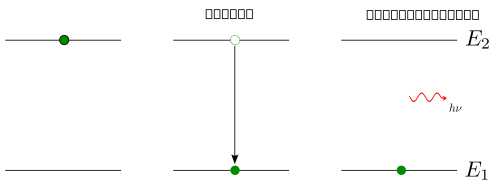
Note
Typically (but not always), the energy source is an applied electromagnetic field that supplies energy in the form of electromagnetic waves. In that case, then $\Delta E = hf$, where f is the frequency of the electromagnetic waves. What this means is that you can use light as an energy source for a laser! In fact, you can use *another laser* as an energy source for a laser!

Crucially, stimulated emission is *different* from spontaneous emission in the following ways:

- The atom is already in an upper state even **before** a photon strikes the atom, due to the external energy source
- This means that *two* photons are emitted from the decay, unlike in spontaneous emission, where only a single photon is released
- The two emitted photons are *identical*, with the same predictable frequency and direction

We call this process *stimulated* emission due to the fact that this entire process is *stimulated* due to the energy source. Since the atom emits two photons, the two outgoing photons can then strike *two* new atoms, which *each* emit two photons, which is four photons total. This doubling process continues, as four photons becomes eight photons becomes sixteen photons, causing a continuous chain reaction. Even better, the resulting light produced is **monochromatic** and in the **same direction**, which are fantastic for building lasers!

Operating principles of lasers After our conceptual review, let us reformulate what we know about light emission and absorption in a more rigorous, mathematical way. Again, we know from the quantum model of the hydrogen atom that electrons can have different states. As each state is an eigenstate of the Hamiltonian, different states (usually) have different energies. If an atom absorbs a photon, for instance, the atom can jump from its lower-energy state, which we write as $|1\rangle$, to a higher-energy upper state, which we write as $|2\rangle$. Meanwhile, an atom can also decay to its lower state by emitting a photon, with the difference in the energies $E_2 - E_1$ between the upper state and the lower state being the energy of this photon. While atoms, in general, have a multitude of states (and more than one upper state), this two-state approximation is good enough for a lot of theoretical analysis. A diagram of the two-state atomic system is shown below:



An atomic system with two possible states ($|1\rangle$ and $|2\rangle$). A transition from $|2\rangle$ to $|1\rangle$ is accompanied by the emission of a photon with energy $E = E_2 - E_1$.

Whether the photon is emitted via stimulated emission or spontaneous emission, precisely *when* the decay happens is random. We do know, however, that this process follows a probabilistic law, first derived by Einstein in 1916. Let us assume we are studying a certain group of atoms. Let $N_2(t)$ be the expected number of atoms in a higher energy state $|2\rangle$. Over time, these electrons will spontaneously decay to a lower energy state $|1\rangle$, such that $N_2(t)$ follows the differential equation:

$$\frac{dN_2}{dt} = -A_{21}N_2 \tag{985}$$

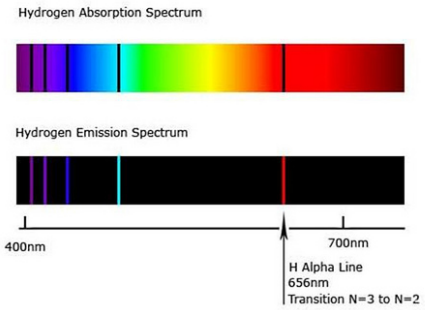
Where A_{21} is called the Einstein A coefficient, and is approximately the probability that an atom will decay from $|2\rangle \to |1\rangle$ per unit time. Remember, this is probabilistic, as in $N_2(t)$ is the *expected* number of electrons in the upper state, as in, it is *most likely* that at time t there are N_2 electrons that are in the upper state $|2\rangle$. This differential equation is just an exponential decay whose solution is:

$$N(t) = N_0 e^{-A_{21}t} \tag{986}$$

Remember that spontaneous emission is fundamentally *random* in nature; they follow general probabilistic rules but those are simply probabilities. But we don't want that, we want emission of photons that we can control.

For this, we need *stimulated emission*. First, we bring the atom to its upper state by surrounding it with an externally-applied energy source. When this energy source is another electromagnetic field (e.g. flash lamp, arc lamp, sunlight, even an LED) is called *optical pumping*. It is in fact possible to use one laser to optically pump another laser - in fact, this is often the most efficient method of pumping a laser, and can be used to take a laser beam of one wavelength to be able to produce a laser beam of another wavelength.

The wavelength used in optical pumping does not necessarily have to be the same wavelength that is emitted. The chief requirement is that the optical source used in optical pumping matches one of the absorption lines of the atom (or molecule, or molecular gas lasers). For instance, hydrogen absorbs (and correspondingly, also emits) wavelengths of 656 nm (red), 486 nm (cyan), 434 nm (blue), 410 nm (violet), and a variety of other wavelengths in the UV band, among others, as shown below:



The spectrum of hydrogen, showing the wavelengths of light hydrogen absorbs and emits. Black lines, also called *spectral lines*, indicate atomic transitions.

But while these absorption lines at the most well-known, they are not its *only* absorption lines, because, due to fine structure and hyperfine structure (the splitting of energy levels due to complex quantum effects), other types of transitions are possible, including the 21 cm absorption line (and emission line) in the microwave spectrum.

When an atom has been raised to its upper state by some energy source, a passing photon that has energy $\Delta E = E_2 - E_1$ will *stimulate* the decay of the atom from its excited-state $|2\rangle$ to its ground-state $|1\rangle$, releasing another further photon of energy ΔE with the *exact same properties* as the passing photon. The passing photon that stimulated the transition between the states, however, is *not absorbed*, meaning that now there are *two* photons of energy ΔE. These two photons can then pass by one atom each, triggering the release of two more photons from each atom, so four photoms after. The process continues, with emitted photons stimulating another atomic transition which would emit another photon, repeating the process over and over to create a cascade of electromagnetic radiation. Thus, we have the laser: an acronym for **light amplification by *simulated emission* of radiation**.

Crucially, stimulated emission needs a large population (number of atoms) to be already in the upper state - otherwise, we would see mostly spontanenous emission rather than stimulated emission. Pumping - that is, introducing an external energy source - is what allows stimulated emission to dominate over spontaneous emission, as the population of the upper state is greater than the population of the lower state, which we call a **population inversion**. We may quantify the expected number of atoms N_2 in the higher energy state $|2\rangle$ with the following differential equation:

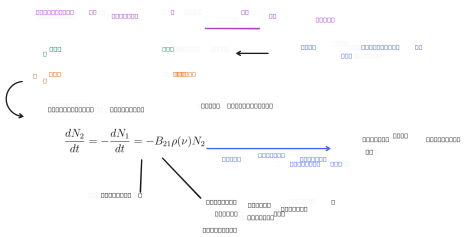
$$\frac{dN_2}{dt} = -B_{21} N \rho(\nu) \tag{987}$$

Where ν is the frequency of the radiation, $\rho(\nu)$ is Planck's law at temperature T, and B_{21} is called the Einstein B coefficient (which we'll derive later), which again is a probabilistic measure of decay

from $|2\rangle \to |1\rangle$:

$$\rho(\nu) = \frac{2h\nu^3}{c^3}\left(\exp\left(\frac{h\nu}{k_B T}\right) - 1\right)^{-1} \tag{988}$$

An annotated sketch showing the inter-relationships between each of these mathematical relations is shown below:



N_1 and N_2, the populations of the upper and lower energy levels, are related via a coupled system of differential equations.

As an example of how this works in practice, the He-Ne laser, a common type of laser operating in the visible spectrum, uses a mixture of helium and neon gas within an optical chamber. An electrical discharge is created in the chamber between the cathode (positive end) and anode (negative end), which acts as an energy source (laser pump source), causing the gas to become a plasma where the electrons are free to move around. The electrons randomly collide with the helium atoms, transferring energy and bringing the helium to an upper state. The helium atoms also collide with the neon atoms, bringing the neon atoms to an upper state and allowing the helium atoms to decay to a lower state. These atoms in upper states provide the conditions for stimulated emission to occur: when one atom decays to lower states and release a photon, another atom would release another photon by stimulated emission. A reflective mirror at one end of the chamber and a semi-transparent mirror at the end reflects the light back and forth, repeating this process over and over and amplifying the light by re-concentrating energy into the gain medium, ensuring that atoms are raised to the upper state, and continuing the cyle of stimulated emission. At a certain point, this cycle of continuous amplification through stimulated emission has progressed far enough that photons escape the optical cavity and begin to pass through the semi-transparent mirror, which is the laser beam we see.

In a laser, light is fundamentally quantized - that is the prerequisite that allows stimulated emission in the first place. Given that photons are quanta of the electromagnetic field, the full picture of laser dynamics would *in theory* require a quantum treatment of electromagnetism, that is, quantum electrodynamics. However, the actual quantum-mechanical workings of lasers can be approximately treated as *separate* from the electromagnetic field produced within it. This means that we can actually describe use *classical* electromagnetic theory - and specifically the **Gaussian beam** solution to the Helmholtz equation - and use it to derive quantum results. And indeed, this is what we will do.

Theoretical analysis **Lasers** are devices that rely on *stimulated emission* to emit light - in fact, LASER is an acronym for "*light amplification by stimulated emission of radiation*". This is in contrast with lightbulbs, stars, or blackbody radiators, which operate by either stimulated emission or absorption. A laser relies on creating the optimal conditions for spontaneous emission to occur.

Quantum mechanically-speaking, a laser can be classified as a multi-state system that undergoes transitions between its states. This requires more advanced methods compared to time-independent systems, which do not have transitions between states, and therefore have constant probabilities to be in each of their possible states . To analyze lasers, we must use **time-dependent perturbation theory**, where the probabilities of each state *are* dependent on time. But before we go into time-dependent perturbation theory, let us review the quantum mechanics background required to understand it.

Recall that in quantum mechanics, every system has an associated quantum state, denoted $|\Psi(t)\rangle$. This quantum state is the solution to the time-dependent Schrödinger equation:

$$i\hbar\frac{\partial}{\partial t}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle \tag{989}$$

Where $\hat{H} = \hat{K} + V$ is the Hamiltonian, which is the sum of the kinetic energy operator $\hat{K}$ and the potential V. The allowed energies E of a quantum system, meanwhile, is governed by the time-independent Schrödinger equation:

$$\hat{H}|\psi\rangle = E|\psi\rangle \tag{990}$$

Where E is the energy and $|\psi\rangle = |\Psi(0)\rangle$ is the state at $t = 0$. The state $|\psi\rangle$ is itself a *superposition* of numerous other states $|\psi_n\rangle$:

$$|\psi\rangle = \sum_n c_n|\psi_n\rangle \tag{991}$$

Where each $|\psi_n\rangle$ individually satisfies the time-independent Schrödinger equation equation:

$$\hat{H}|\psi_1\rangle = E_1|\psi_1\rangle \tag{992}$$
$$\hat{H}|\psi_2\rangle = E_2|\psi_2\rangle \tag{993}$$
$$\vdots \quad \vdots \tag{994}$$
$$\hat{H}|\psi_{n-1}\rangle = E_{n-1}|\psi_{n-1}\rangle \tag{995}$$
$$\hat{H}|\psi_n\rangle = E_n|\psi_n\rangle \tag{996}$$

We find that in many cases, the values of E_n take very specific values: such states are known as *bound states* as they arise when a system is situated within a potential well (such as a Coulomb potential or harmonic potential well). In the well-known case of hydrogen, E_n takes the values:

$$E_n = \frac{-13.6\text{eV}}{n^2} \tag{997}$$

The emission and absorption spectra of hydrogen are based off its values of E_n. This is because the energy absorbed or emitted by a hydrogen atom must be equal to the *energy difference* ΔE between two energy levels. This happens when an electron in a hydrogen atom (or molecule) "jumps" between two orbitals - this can either happen because the atom is excited by an absorbed photon, or an electron releases a photon via either spontaneous or stimulated emission. The energy levels of hydrogen are the simplest of all atoms, but even still, they are diverse:

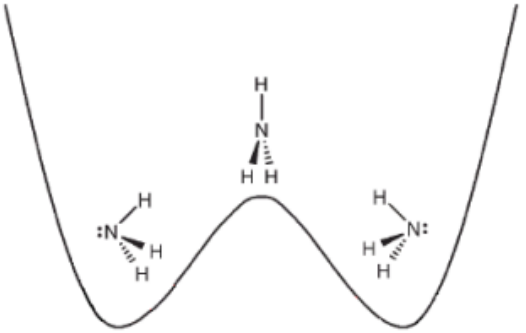
- Orbital transitions (in atomic hydrogen) generally have a ΔE in the UV and visible range
- Molecular transitions (in diatomic hydrogen) generally have a ΔE in the far-infrared range to short microwave range
- Fine transitions have a ΔE in the millimeter-wave range (which are also microwaves)
- Hyperfine transitions have a ΔE in the long microwave range

As we can see, the much smaller energy gaps between the latter three types of transitions - which split the energy levels - comprise the majority of the microwave-producing transitions. While an atom (or molecule) can have a large number of states, we are interested in only in transitions that are microwave-producing. We will now cover one of the simplest analytical solutions that corresponds to a real-world maser system: the ammonia maser.

The ammonia maser Consider a two-state laser whose gain medium is pumped by an external electromagnetic field. One example is the **ammonia molecule** - ammonia has a large number of spectral lines, encompassing near-UV, visible light, far-infrared, and microwaves, coming from electronic (i.e. atomic) transitions, vibrational (molecular) transitions, rotational (also molecular) transitions, among others. However, we are interested in only the transitions that produce microwaves.

Specifically, we consider a specific type of transition called a umbrella inversion. This transition happens when the nitrogen atom in ammonia transitions from being at the "right" of the molecule

to the "left". We can model this as a potential $V(x)$ with two minima, representing each of the two states:



The two states of ammonia are stable equilibria of the potential energy, but the potential can be overcome and result in an atomic transition. Diagram courtesy of LibreTexts.

Such a system can occupy two states: the lower-energy state, which we will call $|1\rangle$ (represented in the above diagram with the ammonia molecule on the left), and the higher-energy state, which we will call $|2\rangle$ (represented by the ammonia molecule on the right). It is also common to refer to $|1\rangle$ as the **lower state** and $|2\rangle$ as the **upper state**, and we will adopt this naming convention for the rest of this chapter. The general state of the system, assuming that transitions are forbidden (and thus $c_1 = c_2 = \text{const.}$), is given by:

$$|\Psi\rangle = c_1|1\rangle e^{-iE_1 t/\hbar} + c_2|2\rangle e^{-iE_2 t/\hbar} \tag{998}$$

where $|1\rangle, |2\rangle$ are eigenstates of the system's Hamiltonian assuming no transitions (which we will call $\hat{H}_0$):

$$\hat{H}|1\rangle = E_1|1\rangle \tag{999}$$

$$\hat{H}|2\rangle = E_2|2\rangle \tag{1000}$$

Now, this is only the case if transitions are forbidden (which is a requirement of time independence) - but we know that transitions between the lower state and upper states do certainly exist. Therefore, we must add time-dependence to the system, which means c_1, c_2 must become functions of time $c_1(t), c_2(t)$:

$$|\Psi\rangle = c_1(t)|1\rangle e^{-iE_1 t/\hbar} + c_2(t)|2\rangle e^{-iE_2 t/\hbar} \tag{1001}$$

Generally, however, these transitions happen randomly, resulting in spontaneous emission, which we don't want for lasers. This means that somehow, we must keep more ammonia molecules in the upper state and less in the lower state, a **population inversion**, for stimulated emission to occur frequently enough that it becomes the dominant mode of light (electromagnetic radiation) production, which is a prerequisite for lasing.

Let us now examine how to formulate what we have described using the theoretical framework of quantum mechanics. We are primarily interested in stimulated emission, so we will not describe spontaneous emission in this section, although the calculations are actually rather similar. We will use the treatment originating with Griffiths (in *Introduction to Quantum Mechanics*) for this.

Consider an applied electromagnetic field $\mathbf{E} = E_0 \cos(\omega t)\hat{k}$, where $\omega \equiv 2\pi f$ and f is the frequency of the field. This is a classic (idealized) solution to Maxwell's equations of electromagnetism. The Hamiltonian must then include both the "standard" Hamiltonian $\hat{H}_0 = \hat{p}^2/2m + V(\mathbf{r})$ as well as the contribution from the electromagnetic field $\hat{H}_1(t) = -qE_0 z \cos\omega t$, which has a dependence on time due to the EM field. Thus the complete Hamiltonian is given by:

$$\hat{H} = \hat{H}_0 + \hat{H}_1(t) \tag{1002}$$
$$= \hat{H}_0 - qE_0 z\cos\omega t \tag{1003}$$

The inclusion of the external EM field Hamiltonian is **crucial**. Recall how we previously saw that an applied EM field raises an atom's electrons to an upper state, allowing stimulated emission to occur. The $\hat{H}_1(t)$ term in the Hamiltonian, also called a **perturbation term**, expresses this fact.

Now, let us assume we have already solved for the eigenstates of $\hat{H}_0$, and these are given by $|\psi_1\rangle, |\psi_2\rangle, |\psi_3\rangle, \dots$ where $|\psi_1\rangle$ is the lower state and $|\psi_2\rangle, |\psi_3\rangle, \dots$ are the upper states. The eigenstates are orthonormal and thus satisfy $\langle\psi_i|\psi_j\rangle = \delta_{ij}$. The state of the system can be written in terms of these eigenstates as:

$$|\Psi(t)\rangle = c_1|\psi_1\rangle e^{-iE_1t/\hbar} + c_2|\psi_2\rangle e^{-iE_2t/\hbar} + c_3|\psi_3\rangle e^{-iE_3t/\hbar} + \cdots + c_n|\psi_n\rangle e^{-iE_nt/\hbar} \tag{1004}$$

Where $c_1, c_2, \dots c_n$ are time-dependent coefficients (also called **transition amplitudes**) whose squared norm $|c_n(t)|^2$ is the probability of finding each eigenstate at time t. For simplicity, let's consider a system with only two states: the lower state $|\psi_1\rangle$, which has energy E_1, and one upper state $|\psi_2\rangle$, which has energy E_2. This may seem like a ridiculous simplification, given that atoms often have dozens of energy levels, but often, the electron transitions relevant to lasers happen only between two energy levels, so this is a *reasonable* assumption (indeed this is true for the famous ammonia laser[12]). Then, the state of the system would be given by:

$$|\Psi(t)\rangle = c_1|\psi_1\rangle e^{-iE_1t/\hbar} + c_2|\psi_2\rangle e^{-iE_2t/\hbar} \tag{1005}$$

Where once again, remember that c_1, c_2 are both functions of time. We now aim to solve for $c_1(t)$ and $c_2(t)$, the transition amplitudes. To do so, we plug $|\Psi(t)\rangle$ into the Schrödinger equation $i\hbar\dfrac{\partial}{\partial t}|\Psi(t)\rangle = \hat{H}|\Psi(t)\rangle$, where, remember, $\hat{H} = \hat{H}_0 + \hat{H}_1(t)$. The resulting expression is rather long:

$$c_1(t)\hat{H}_0|\psi_1\rangle e^{-iE_1t/\hbar} + c_2(t)\hat{H}_0|\psi_2\rangle e^{-iE_2t/\hbar} + c_1(t)\hat{H}_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} \tag{1006}$$
$$+c_2(t)\hat{H}_1(t)|\psi_2\rangle e^{-iE_2t/\hbar} = i\hbar\Big[\frac{dc_1}{dt}|\psi_1\rangle e^{-iE_1t/\hbar} + \frac{dc_2}{dt}|\psi_2\rangle e^{-iE_2t/\hbar} \tag{1007}$$
$$-\frac{iE_1}{\hbar}c_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} - \frac{iE_2}{\hbar}c_2(t)|\psi_2\rangle e^{-iE_2t/\hbar}\Big] \tag{1008}$$

Which we can slightly simplify (by expanding the brackets) to:

$$c_1(t)\hat{H}_0|\psi_1\rangle e^{-iE_1t/\hbar} + c_2(t)\hat{H}_0|\psi_2\rangle e^{-iE_2t/\hbar} + c_1(t)\hat{H}_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} \tag{1009}$$
$$+c_2(t)\hat{H}_1(t)|\psi_2\rangle e^{-iE_2t/\hbar} = i\hbar\left[\frac{dc_1}{dt}|\psi_1\rangle e^{-iE_1t/\hbar} + \frac{dc_2}{dt}|\psi_2\rangle e^{-iE_2t/\hbar}\right] \tag{1010}$$
$$-i\hbar\frac{iE_1}{\hbar}^{E_1}c_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} - i\hbar\frac{iE_2}{\hbar}^{E_2}c_2(t)|\psi_2\rangle e^{-iE_2t/\hbar} \tag{1011}$$

But recall that since $|\psi_1\rangle, |\psi_2\rangle$ are eigenstates of $\hat{H}_0$, they satisfy:

$$\hat{H}_0|\psi_1\rangle = E_1|\psi_1\rangle \tag{1012}$$
$$\hat{H}_0|\psi_2\rangle = E_2|\psi_2\rangle \tag{1013}$$

So, substituting in, we find that the terms actually cancel quite nicely:

[12] https://www.britannica.com/technology/ammonia-maser

$$c_1(t)\hat{H}_0|\psi_1\rangle e^{-iE_1t/\hbar} + c_2(t)\hat{H}_0|\psi_2\rangle e^{-iE_2t/\hbar} + c_1(t)\hat{H}_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} \tag{1014}$$

$$+c_2(t)\hat{H}_1(t)|\psi_2\rangle e^{-iE_2t/\hbar} = i\hbar\left[\frac{dc_1}{dt}|\psi_1\rangle e^{-iE_1t/\hbar} + \frac{dc_2}{dt}|\psi_2\rangle e^{-iE_2t/\hbar}\right] \tag{1015}$$

$$c_1(t)E_1|\psi_1\rangle e^{-iE_1t/\hbar} + c_2(t)E_2|\psi_2\rangle e^{-iE_2t/\hbar} \tag{1016}$$

Meaning that we are left with simply:

$$c_1\hat{H}_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} + c_2\hat{H}_1(t)|\psi_2\rangle e^{-iE_2t/\hbar} \tag{1017}$$

$$= i\hbar\left[\frac{dc_1}{dt}|\psi_1\rangle e^{-iE_1t/\hbar} + \frac{dc_2}{dt}|\psi_2\rangle e^{-iE_2t/\hbar}\right] \tag{1018}$$

Where again $c_1 = c_1(t)$ and $c_2 = c_2(t)$. Since the eigenstates are orthonormal and thus obey $\langle\psi_i|\psi_j\rangle = \delta_{ij}$, if we multiply by $\langle\psi_1|$ on all sides, we would have:

$$c_1\langle\psi_1|\hat{H}_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} + c_2\langle\psi_1|\hat{H}_1(t)|\psi_2\rangle e^{-iE_2t/\hbar} \tag{1019}$$

$$= i\hbar\left[\frac{dc_1}{dt}\langle\psi_1|\psi_1\rangle^1 e^{-iE_1t/\hbar} + \frac{dc_2}{dt}\langle\psi_1|\psi_2\rangle^0 e^{-iE_2t/\hbar}\right] \tag{1020}$$

Which reduces to:

$$c_1\langle\psi_1|\hat{H}_1(t)|\psi_1\rangle e^{-iE_1t/\hbar} + c_2\langle\psi_1|\hat{H}_1(t)|\psi_2\rangle e^{-iE_2t/\hbar} \tag{1021}$$

$$= i\hbar\frac{dc_1}{dt}e^{-iE_1t/\hbar} \tag{1022}$$

We can do some rearranging (dividing by $i\hbar$ and multiplying by $e^{iE_1t/\hbar}$ on both sides) to get $\frac{dc_1}{dt}$ on one side, leaving us with an ODE for c_1:

$$\frac{dc_1}{dt} = -\frac{i}{\hbar}\left[c_1\langle\psi_1|\hat{H}_1(t)|\psi_1\rangle + c_2\langle\psi_1|\hat{H}_1(t)|\psi_2\rangle\right]e^{-i(E_2-E_1)t/\hbar} \tag{1023}$$

> **Note**
> If it is unfamiliar, recall that $\frac{1}{i} = -i$

Repeating the same process, only multiplying by $\langle\psi_2|$ rather than $\langle\psi_1|$, gives us an ODE for c_2:

$$\frac{dc_2}{dt} = -\frac{i}{\hbar}\left[c_2\langle\psi_2|\hat{H}_1(t)|\psi_2\rangle + c_1\langle\psi_2|\hat{H}_1(t)|\psi_1\rangle\right]e^{i(E_2-E_1)t/\hbar} \tag{1024}$$

The system of ODEs for c_1 and c_2 can be rewritten in matrix form (where here, $\hat{H}_1 = \hat{H}_1(t)$ as with before), just as we previously saw in the matrix representation section:

$$\begin{pmatrix}\dot{c}_1\\ \dot{c}_2\end{pmatrix} = -\frac{i}{\hbar}\begin{pmatrix}\langle\psi_1|\hat{H}_1|\psi_1\rangle & \langle\psi_1|\hat{H}_1|\psi_2\rangle\\ \langle\psi_2|\hat{H}_1|\psi_1\rangle & \langle\psi_2|\hat{H}_1|\psi_2\rangle\end{pmatrix}\begin{pmatrix}c_1e^{-i(E_2-E_1)t/\hbar}\\ c_2e^{i(E_2-E_1)t/\hbar}\end{pmatrix} \tag{1025}$$

Let us assume that at time $t = 0$, the atom is in its upper state $|\psi_2\rangle$ with energy E_2. Thus the initial condition would be 100% probability of the $|\psi_2\rangle$ state and 0% probability of the $|\psi_1\rangle$ state:

$$c_1(0) = 0, c_2(0) = 1 \tag{1026}$$

We want to solve for $c_1(t)$, which will give us the probabilities of the atom decaying to the lower state at some future time t (remember, even in stimulated emission, the decay time is random, only the *probability* of a decay is predictable). The differential equations are indeed quite intimidating to solve. There are, however, some steps we can use to simplify. First, the diagonals of the matrix are often zero; see this physical argument on Physics SE which explains why, which reduces each of the ODEs by one term, so that we "only" have:

$$\begin{pmatrix}\dot{c}_1\\ \dot{c}_2\end{pmatrix} = -\frac{i}{\hbar}\begin{pmatrix}c_2\langle\psi_1|\hat{H}_1|\psi_2\rangle e^{i(E_2-E_1)t/\hbar}\\ c_1\langle\psi_2|\hat{H}_1|\psi_1\rangle e^{-i(E_2-E_1)t/\hbar}\end{pmatrix} \tag{1027}$$

We can then make use of a *perturbative expansion*. Let's first *assume* that $\dot{c}_1, \dot{c}_2$ are small, meaning that transition between the states (including decays) happen relatively infrequently. If the transition rates are small enough, we can assume that $\dot{c}_2 \approx 0$. If this is the case, then $c_2(t) \approx c_2(0) = 1$. If we substitute this value of c_2 into the top ODE of the matrix system (the ODE for $\dot{c}_1$, we have:

$$\frac{dc_1}{dt} = -\frac{i}{\hbar}c_2\langle\psi_1|\hat{H}_1|\psi_2\rangle e^{i(E_2-E_1)t/\hbar} \tag{1028}$$

$$\approx -\frac{i}{\hbar}(1)\langle\psi_1|\hat{H}_1|\psi_2\rangle e^{i(E_2-E_1)t/\hbar} \tag{1029}$$

$$\Rightarrow \frac{dc_1}{dt} = -\frac{i}{\hbar}\langle\psi_1|\hat{H}_1|\psi_2\rangle e^{i(E_2-E_1)t/\hbar} \tag{1030}$$

With this simplification, the ODE becomes solvable - the solution is:

$$c_1(t) = -\frac{i}{\hbar}\int_0^t \langle\psi_1|\hat{H}_1|\psi_2\rangle e^{i(E_2-E_1)t'/\hbar}\,dt' \tag{1031}$$

Note
This expansion is a *first-order* expansion, but in theory we can expand to any arbitrary order. The in-depth study of solving time-dependent quantum systems (systems that have some sort of time-dependent Hamiltonian) is known as *time-dependent perturbation theory*; this is just a very introductory treatment.

If we substitute our applied EM field Hamiltonian, which has $\hat{H}_1(t) = -qE_0 z\cos\omega t$, then the solution (once you perform the integral) is:

$$c_1(t) = \frac{i}{\hbar}qE_0\langle\psi_1|z|\psi_2\rangle\frac{\sin[(\omega_0-\omega)t/2]}{\omega_0-\omega}e^{i(\omega_0-\omega)t/2}, \quad \omega_0 = \frac{E_2-E_1}{\hbar} \tag{1032}$$

Taking the squared norm of c_1 yields the *probability* of the transition from the upper state to the lower state, which we will denote P_{21}:

$$P_{21}(t) = |c_1|^2 = |q\langle\psi_1|z|\psi_2\rangle|^2\left(\frac{E_0}{\hbar}\right)^2\frac{\sin^2[(\omega_0-\omega)t/2]}{(\omega_0-\omega)^2} \tag{1033}$$

Note that we can also write this in terms of the *energy density* of an electromagnetic wave, $u_E = \frac{\varepsilon_0}{2}E_0^2$, as:

$$P_{21}(t) = \frac{2u_E}{\varepsilon_0\hbar^2}|q\langle\psi_1|z|\psi_2\rangle|^2\frac{\sin^2[(\omega_0-\omega)t/2]}{(\omega_0-\omega)^2} \tag{1034}$$

Reminder
If we know the wavefunction representation of $|\psi_1\rangle, |\psi_2\rangle$, then the inner product becomes an integral, so we have $\langle\psi_1|z|\psi_2\rangle = \int \psi_1^*(\mathbf{r})z\,\psi_2(\mathbf{r})\,dV$.

We should add one more caveat: this is for a **monochromatic** applied electric field with a strict *linear polarization*. In practice, an applied electric field would be composed of many different frequencies, and would be a mix of different polarizations (for example, sunlight ranges from 300 to 2500 nm and is certainly unpolarized by the time it reaches Earth)[13]. In that case, instead of a single energy density, we instead have a *spectral energy density* $\rho(\nu)$, which gives the electromagnetic energy density at frequency ν. Recall again that Planck's law of blackbody radiation tells us that this can be approximated by:

$$\rho(\nu) = \frac{8\pi h\nu^3}{c^3}\frac{1}{e^{h\nu/k_BT}-1} \Rightarrow \rho(\omega) = \frac{\hbar\omega^3}{\pi^2c^3}\frac{1}{e^{\hbar\nu/k_BT}-1} \tag{1035}$$

Where k_B is the Boltzmann constant T is the temperature, and $\nu = \omega/2\pi$. In this case, the more general expression for the probability of the transition via stimulated emission (which we will not derive) is:

$$P_{21}(t) = |c_1|^2 = \frac{\pi}{3\varepsilon_0\hbar^2}|q\langle\psi_1|\mathbf{r}|\psi_2\rangle|^2\rho(\omega_0)t \tag{1036}$$

Taking the time derivative of the transition probability gives us the **transition rate**, the probability of a transition per unit time. In fact, with the exception of a factor $\rho(\nu)$ to make the units consistent (due to how the Einstein coefficients are defined), this is what we know to be the **Einstein B coefficient**:

$$B_{21}\rho(\nu) = \frac{dP}{dt} = \frac{\pi}{3\varepsilon_0\hbar^2}|q\langle\psi_1|\mathbf{r}|\psi_2\rangle|^2\rho(\omega_0) \tag{1037}$$

And the Einstein coefficients in the rate equations are given by:

$$A_{21} = \frac{\hbar\omega_0^3}{\pi^2c^3}B_{21}, \tag{1038}$$

$$B_{21} = \frac{\pi}{3\varepsilon_0\hbar^2}|q\langle\psi_2|\mathbf{r}|\psi_1\rangle|^2 \tag{1039}$$

From which we may solve the rate equations that govern the population of the $|\psi_1\rangle$ and $|\psi_2\rangle$ states, which, as a reminder, are given by:

$$\frac{dN_2}{dt} = -\frac{dN_1}{dt} = -B_{21}\rho(\nu)N_2 \tag{1040}$$

Note that the **mean lifetime** τ of the upper state, meaning the average amount of time an atom spends in the upper state $|\psi_2\rangle$ before decaying into the lower state $|\psi_1\rangle$, can be calculated from the Einstein A coefficient as follows:

$$\tau = \frac{1}{A_{21}} \tag{1041}$$

Of course, this is for just a two-level system (with one upper state and one lower state). Many lasers are three-level or four-level systems, and thus do not have such simple expressions for finding the transition probability. In the most complicated of cases, numerical methods can be used for solving the matrix ODEs to find the transition rates and the rate equations to solve for the population of each state.

Fermi's golden rule Our quantum model of lasers is actually just one example in the broader field of **time-dependent perturbation theory**, which describes the dynamics of multi-state quantum systems (like lasers!) Just as we saw at the very beginning of our discussion, time-dependent perturbation theory allows for the possibility of *transitions* between states. For such a system, we write out a Hamiltonian in the following form, just like we did for analyzing lasers:

[13] https://www.sciencedirect.com/topics/physics-and-astronomy/solar-spectra

$$\hat{H} = \hat{H}_0 + \hat{H}_1(t) \tag{1042}$$

Where $\hat{H}_0$ is the **time-independent Hamiltonian**, and $\hat{H}_1$ is the **perturbation Hamiltonian**, which is responsible for the time-dependent behavior of the system. Fermi's golden rule says that *to first-order*, the probability of a transition per unit time Γ_{fi} from initial state $|i\rangle$ to final state $|f\rangle$ is given by:

$$\Gamma_{fi} = \frac{2\pi}{\hbar}|\mathcal{M}_{fi}|^2\rho(E_f) \tag{1043}$$

Where $\rho(E_f)$ is the **density of states** at energy E_f, which we'll explain in a little bit, and $\mathcal{M}_{fi}$ is called the **matrix element** for the transition, which is given by:

$$\mathcal{M}_{fi} = \langle f|\hat{H}_1|i\rangle \tag{1044}$$

Fermi's golden rule is applicable to a broad range of multi-state quantum systems, from nuclear physics to scattering processes described in quantum field theory (for those interested, $\mathcal{M}_{fi}$ becomes the first-order term of the S-matrix in QFT). The mean lifetime τ of the state $|f\rangle$ before a decay to state $|i\rangle$ can also be found from the transition rate Γ_{fi} via:

$$\tau = \frac{1}{\Gamma_{fi}} \tag{1045}$$

Important
What is the density of states, $\rho(E)$? It is just the **number of states per unit energy**. We know that a quantum system has multiple states, and that a system can be in different states depending on the total energy of the system. The density of states is important in measuring the **change in the total number of states** of the system, ΔN, when there is a change in the energy, via $\Delta N = \rho(E)\Delta E$. This means that if you increase (or decrease) the energy by an amount ΔE, the density of states tells you that the corresponding change in the number of states is ΔN. In the continuous limit, this becomes $dN = \rho(E)dE$, or $\rho(E) = \frac{dN}{dE}$, at which point physical intuition can fail, but the math stays the same. Note that in some applications, it is customary to write the density of states averaged over the *total volume*, that is, $\rho(E) = \frac{1}{V}\frac{dN}{dE}$, but in Fermi's golden rule we use the form $\rho(E) = \frac{dN}{dE}$.

Using the transition rate, one may then find the following differential equation relating the population N_i of state $|i\rangle$ with the population N_f of the state $|f\rangle$, which are simply the exponential decay/growth equations, *assuming* that there are no other transitions than the transition $|i\rangle \to |f\rangle$:

$$\frac{dN_i}{dt} = -\Gamma_{fi}N_i \tag{1046}$$

$$\frac{dN_f}{dt} = \Gamma_{fi}N_f \tag{1047}$$

This looks very similar to the laser rate equations for a two-level system! Indeed, that is *almost* correct.

In our case of lasers, Γ_{fi} is A_{21}, the Einstein A coefficient, and therefore $\tau = 1/\Gamma_{fi} = 1/A_{21}$ is the lifetime of the upper state. Our entire process of finding $c_1(t)$ and $c_2(t)$ and calculating the transition probability P_{21} could have been avoided, had we used Fermi's golden rule directly. It is a *very* powerful tool to use when doing calculations.

The characteristics of laser light At the very start of our discussion of lasers, we noted that lasers produce light with very specific characteristics:

- **Monochromatic light**: The light is of one frequency (or essentially one frequency)
- **Directionality**: The laser beam travels in one direction and can be focused tightly
- **Amplification:** The emission of one photon triggers the emission of two more, starting a chain reaction that leads to a cascade of light, and making it possible to create powerful beams of light

With all we've learned, we can now answer the question of *why* laser light has these properties. First, the reason why light from a laser is only of one frequency comes directly from the stimulated emission process. Remember that since stimulated emission always produces two **identical** photons, which have identical frequencies. Since those two photons go on to trigger the emission of *two more* identical photons each (so four photons total), we have a chain reaction that continues, doubling the number of photons each time. This means that eventually, (nearly) all the photons in the optical cavity will be produced by stimulated emission, and they will be identical to each other, giving laser light its characteristic monochromaticity. However, this is only possible because lasers maintain a *population inversion*, since stimulated emission is only favored when the upper state has a higher population than the lower state. Normal light sources do not maintain a population inversion, so *spontaneous emission* dominates over stimulated emission, and as a result, their light is spread over a range of frequencies and is not monochromatic.

Second, the reason why laser light exhibits strong directionality is due to the mirrors in the laser's optical cavity. We'll first start by giving a more intuitive but less rigorous explanation. Imagine a photon that is emitted by stimulated emission, inside the optical cavity: if the photon's direction is not exactly normal (90 degrees) to the mirror, it will reflect off the mirror at an angle, causing it to eventually hit the walls of the optical cavity, where it is absorbed and can no longer be reflected. Only photons *normal* to the mirror can get reflected again at the mirror on the other side, where they can continue travelling through the optical cavity.

The more complicated but also more rigorous explanation comes in the form of the wave nature of light. Light is an electromagnetic wave, and electromagnetic waves exhibit *interference*: when two waves are added together, if they have a different phase, they will interfere with each other. An electromagnetic wave that propagates along the optical axis (normal to the mirror) and another wave propagating at an angle to the optical axis would have a different phase, because two waves travelling over different distances will always have a phase difference proportional to the difference in the distances. Normally, we don't notice this, because light travels so fast (it can travel around the Earth in 1/6th of a second!) that any phase shift is far beyond anywhere we could see. But in the closed optical cavity of a laser, surrounded by reflecting mirrors, small differences in phase add up as the electromagnetic waves reflect back-and-forth between the mirrors, quickly (in fact near-instantly) building up to the point that *constructive interference* leads to the waves along the optical axis to add up, and *destructive interference* leads to the waves that are at an angle to be cancelled out. This means we're just left with waves travelling parallel to the optical axis, giving us a highly-directional, straight beam, whose power is concentrated along the direction of the beam.

Laser physics, part 2

Up to this point, we have covered the following principles of how lasers work:

- A laser requires some sort of power source (usually an applied EM field) to bring the atoms (or ions or molecules, etc.) in some material to an upper (higher-energy) state (we call this **laser pumping**)
- Once the laser is in this state, any incident photons lead to the emission of two more photons
- The two photons start a chain reaction that leads to the exponential emission of monochromatic, coherent, strongly-directional light (or other form of EM radiation)

We now introduce some technical terminology. A laser is typically composed of an energy source as well as a (nearly) completely-sealed cavity, filled with some material, called the laser cavity (or *optical cavity*). The material is known as the **gain medium** or **lasing medium**[14]; the atoms of the gain medium are excited by the energy source and emit light. Gain media (*media* is plural of *medium*) can be anything from a solid (e.g. crystalline solid), liquid solution (e.g. organic dyes), or gas (e.g. hydrogen/argon/carbon dioxide gas)[15]. The gain media is then pumped with energy from the laser's power source; typically, this is either a strong electric current, a very, very bright light (which is an EM field, since light is an EM wave), or another laser (which, again, is also an EM field).

To keep as many of the atoms as possible in the upper state, which is necessary for stimulated emission, the laser cavity has mirrors at its ends, which reflect the light back and forth throughout the material, continuously re-injecting energy back through the gain medium. When more than 50% of the atoms within the gain medium are in their upper state[16], we say that a **population inversion** has occured. At this point, stimulated emission takes over, causing the chain reaction that leads to a rapid emission of more and more photons. One end of the laser cavity is a semi-transparent mirror[17] that is designed to let a small fraction of the light through, while the rest of the light reflects off to continue the stimulated emission process inside the cavity. This mirror is often called an **output coupler**. The light that makes it through the output coupler forms the characteristic beam that emerges from one end of the laser.

The four main components of a laser - power source, gain medium, laser (optical) cavity, and output coupler[18] - are each complex topics in and of themselves, and especially for high-performance lasers, each requires meticulous design and engineering. Lasers fine-tuned to specific tasks often have different requirements for the type of beam, wavelength/frequency, and power efficiency of the laser. Hence, we will discuss lasers in *much greater depth* in the following sections.

Lasing transitions and the gain medium When designing a laser, we first want to consider the material composition of the gain medium. The gain medium could be composed of elemental atoms, ions, molecules (like the ammonia molecule used in the ammonia maser), crystals, semiconductors, or even a combination of different materials. For this reason, we will use the generic term "quantum system" to represent the atomic/molecular/ionic/etc. constituents of the laser's gain medium, instead of specific terms like "atom" or "molecule".

States and energy levels First, we want to identify all the states of the quantum system. This is done by solving for the states of the quantum system with the *time-independent part* of the Hamiltonian. Just as we showed earlier in performing our calculations for the ammonia maser, if we let the total Hamiltonian of the system be $\hat{H} = \hat{H}_0 + \hat{H}_1(t)$, where $\hat{H}_0$ is the time-independent portion and $\hat{H}_1(t)$ is the time-dependent portion, then the states $|\psi_n\rangle$ of the system are found by solving the time-dependent Schrödinger equation:

$$\hat{H}_0|\psi_n\rangle = E|\psi_n\rangle \tag{1048}$$

[14] Optical cavity
[15] List of laser types
[16] Stimulated emission#Optical amplification
[17] Output coupler
[18] https://commons.wikimedia.org/wiki/File:Laser.svg

Depending on the system's complexity, an analytical solution may be possible to find (for instance, in the case of the hydrogen atom) or be impossible to find (for instance, in the case of all multi-electron atoms). This step therefore may often require using approximations or numerical methods to calculate the states and find the energy eigenvalues, from which we may obtain the emission spectrum (all the wavelengths the system can emit light).

Selection rules Next, we want to find the *possible transitions* between different energy states. This involves the selection rules. A selection rule defines which transitions between states are possible and which transitions are impossible in a quantum multi-state system. A transition between two states $|\psi_m\rangle \to |\psi_n\rangle$ is said to be **allowed** *if* the quantity $\langle\psi_m|\mathbf{r}|\psi_n\rangle$, called the *matrix element*, satisfies:

$$\langle\psi_m|\mathbf{r}|\psi_n\rangle \neq 0 \tag{1049}$$

However, if $\langle\psi_m|\mathbf{r}|\psi_n\rangle = 0$, a transition is said to be **forbidden**. For instance, the states of the hydrogen atom, which are parametrized by three quantum numbers (n, m, and ℓ) and which we write as $|\psi_{m,n,\ell}\rangle$, satisfy the following selection rules:

$$\langle\psi_{m,n,\ell}|\mathbf{r}|\psi_{m',n',\ell'}\rangle = \begin{cases} \text{nonzero}, & m' = m \text{ and } l' = l \pm 1, \\ 0, & \text{otherwise} \end{cases} \tag{1050}$$

In theory, we must compute the matrix element $\langle\psi_m|\mathbf{r}|\psi_n\rangle$ for all the possible combinations of states $|\psi_m\rangle$ and $|\psi_n\rangle$ to find all possible transitions; in practice, if the states of a quantum system already satisfy certain orthogonality relations, we can quickly tell which matrix elements are zero, and therefore which transitions are allowed or forbidden.

Identifying transitions Next, we need to sort through all the allowed transitions to identify the *ideal transitions*. Atoms usually have many different possible radiative transitions (also called *decay modes*, "radiative" means that the transition leads to a photon being emitted), so we want to find the *best decay modes*. A decay mode is ideal when all the below conditions are satified[19]:

1. The transition wavelength of the decay mode corresponds with the desired wavelength of the laser light. For instance, you would look for transition wavelengths between 780nm-2500nm if your target wavelength for the laser is in the near-infrared range.

2. It should be easy to create a population inversion. In practical terms, it means that the transition rate (probability of a transition per unit time) of stimulated emission should be much higher than the transition rate of spontaneous emission in the system.

3. The upper state in the decay mode should have a generally long lifetime τ, so that the quantum system maintains a higher population in the upper state compared to the lower state (which again is what leads to a stable population inversion)

4. The lower state in the decay mode should be quickly depopulated by some mechanism[20], so that the lower state does not simply re-absorb the same radiation it emitted. This depopulation mechanism should be non-radiative (meaning it doesn't involve releasing a photon), instead causing the lower state to decay by some other means, like the emission of a phonon (a type of structural vibration) for solid gain media or by molecular collisions that transfer away energy in gases[21].

[19] From RP Photonics Encyclopedia, https://www.rp-photonics.com/laser_transitions.html
[20] From RP Photonics Encyclopedia, https://www.rp-photonics.com/lower_state_lifetime.html
[21] From RP Photonics Encyclopedia, https://www.rp-photonics.com/four_level_and_three_level_laser_gain_media.html#Four-Level-Systems

5. The decay mode can lead to another decay, meaning that the system in the lower state can decay to an even lower state. This requires that the lifetime of the lower state is not too long, and that it has allowed transitions (by the selection rules) with a high transition probability to decay to another state. Multiple decays allows building *multi-level* lasers, which have many advantages over two-level lasers (lasers that use a transition from only one upper state to the ground/lower state)

Nearly all of these conditions can be checked against with (tedious) calculations. The transition wavelength comes from calculating the energy eigenvalues associated with each of the states of the system. Using the energy eigenvalue expression $E_j = \langle\psi_j|\hat{H}|\psi_j\rangle$, which we learned from the section on the matrix representation of operators, we can write the transition wavelength for a transition from upper state $|\psi_m\rangle$ to lower state $|\psi_n\rangle$ as follows:

$$\lambda_{mn} = \frac{hc}{|E_m - E_n|} = \frac{hc}{\left|\langle\psi_m|\hat{H}_0|\psi_m\rangle - \langle\psi_n|\hat{H}_0|\psi_n\rangle\right|} \tag{1051}$$

From calculating all possible transition wavelengths for the allowed transitions, we can build up the **spectrum** of the system, giving all the wavelengths of light that the system can (theoretically) emit. We need to then filter the transitions by our other conditions. The transition rates Γ_{fi} for stimulated and spontaneous emission can be calculated using Fermi's golden rule the same method we outlined for the ammonia maser, from which we may easily obtain the lifetime τ of the upper state with $\tau = 1/\Gamma_{fi}$, as we mentioned earlier. Multi-level decays can be checked with the selection rules' matrix elements $\langle\psi_m|\mathbf{r}|\psi_n\rangle$ as well as further computation of transition rates.

In the case of **optically-pumped lasers** (and masers), we also need the gain medium to absorb light strongly around the **pump wavelength**, which is the wavelength of the light used to pump the laser/maser. The wavelength of the laser's light *does not always* match the pump wavelength, and the key reason for this is that **not all lasers are two-level systems**; three-level and four-level lasers (which we'll get to soon) are excited to their highest-energy state by the pump source, but decay in several steps back to the lowest-energy state, where stimulated emission takes place in (usually) one of these transition. Such lasers typically use a mixture of different atomic species or molecules (species means the same as "type"), and especially for solid-state lasers that use a combination of solids (usually crystals) with ions of a different element packed into their crystal lattice. In such materials, one of the species is typically the one that absorbs light and quickly decays to a lower energy state, triggering additional decays in the next species. For solid-state lasers in particular, the species responsible for light absorption is typically the solid, which (again) is typically a crystal such as $Y3Al5O12$ (yttrium aluminum garnet) or (yttrium lithium fluoride).

Three and four-level systems The ammonia maser, which we've been analyzing, is a *two-level laser*. Unfortunately, it is not a very practical laser, and emits miniscule amounts of power in its microwave beam. This is all to do with the fact that it operates as a **two-state system**, where we have only one upper and one lower state.

In a two-state system, any atom (or ion or molecule) that decays from the upper state must necessarily increase the population of the lower state. But a laser operates by achieving a population inversion, where the upper state has a greater population than the lower state, so we need to constantly boost the atoms back into the upper state using our pump source. This *can* theoretically be done, but uses a lot of energy. A good analogy, which we borrow here from the textbook *Physical Chemistry* from Libretexts[22], is reversing the flow of water in a waterfall; while possible, it is incredibly energy-consuming and inefficient.

Any form of pumping can only establish a *thermal equilbrium* between the two states, where the population N_2 of the upper state and the population N_1 of the lower state are equal, that is, $N_1 = N_2$. Thus, $N_2/N_1 = 1$, which cannot create a population inversion, since a population inversion is created (by definition) when the population of a upper state N_j and the population of a lower state N_i satisfies $N_j/N_i > 1$. The ammonia maser gets around this issue by not using a pump source at all; it uses

[22] *Physical Chemistry*, Libretexts, Chapter 15.3

an electric field to separate ammonia molecules that happen to naturally be in the higher-energy state from those of the lower-energy state, using the fact that they have different angular momenta. Unfortunately, it also means that it outputs very little power. We can also get around this issue by running the laser in *pulses*, so that there is not enough time for thermal equilibrium to develop, and we can immediately pump more energy after each pulse to raise the population of the upper level. But it is **impossible** to make a pumped two-level laser (or maser) operating continuously - you either need to give up continuous operation and make the laser pulsed, or give up pumping and end up with a very weak laser/maser beam.

So it should come as no surprise that almost all lasers are either three-level lasers or four-level lasers. The most efficient lasers, particularly for continuous operation, are four-level lasers.

Case studies of lasers We have seen the general mechanism of all lasers, but let's now examine specific types of lasers to gain a deeper understanding of how *real* lasers work. The five laser/maser types we will examine are all very well-known and represent a variety of different laser designs:

Name	Gain medium	Emitted wavelength	Type of laser light	Pump source
Ammonia maser	Ammonia gas	1.26 cm	Microwave	None
Hydrogen maser	Hydrogen gas	21 cm	Microwave	None
Rubidium maser	Rubidium crystal	4.39 cm	Microwave	Rubidium lamp
Nd:YAG laser	Neodynium-doped (Nd^{3+}) yttrium aluminium garnet crystal (YAG)	1064 nm	Infrared	Flashlamp
HeNe laser	Helium and neon gas	633 nm	Visible (red) light	Electrical discharge
Ruby laser	Ruby crystal, more specifically chromium-doped (Cr^{3+}) aluminum (III) oxide ($Al2O3$)	694.3 nm	Visible (red) light	Electrical discharge

The ammonia maser The first of the two microwave laser (maser) types we will cover is the **ammonia maser**. The ammonia maser is one of the simplest types of lasers/masers; this is because it is a **two-level system**, composed of just one ground state and one excited state, which we have already analyzed in-depth previously.

The lasing action of the ammonia maser results from the ammonia molecule transitioning between the two states. This physically occurs as a result of the nitrogen atom in the ammonia molecule "flipping" to the opposite side of the molecule, which is often called a **nitrogen inversion** or **umbrella inversion transition**[23]. As these two states have slightly-different energies, there exists an *energy difference* between the two states of about 97.8 μeV. As you can see, this is an *extremely small* energy difference, hence why it leads to the emission of photons in the *microwave range* (which carry much less energy than visible or UV light).

> **Note:** Technically-speaking, the ammonia molecule has a lot more possible states, but only two of them are relevant in the ammonia maser, so we can just consider those two states. Additionally, the lower-energy state that we have been calling the "ground state" is not *technically* the ground state. We have only chosen to use this terminology for its much greater familiarity.

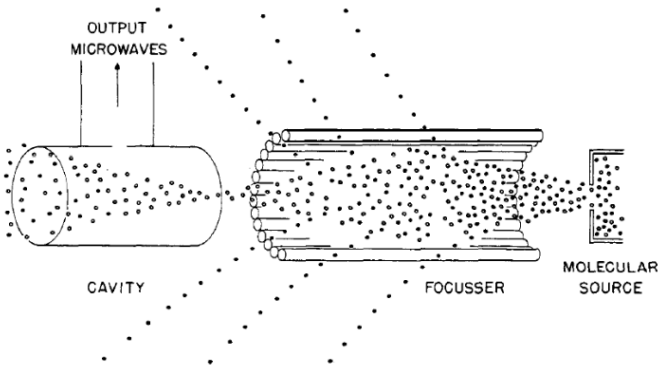
[23] https://chem.libretexts.org/Bookshelves/Physical_and_Theoretical_Chemistry_Textbook_Maps/Quantum_Tutorials_(Rioux)/

The umbrella inversion transition happens spontaneously as a result of quantum tunneling "through" the molecule. The configuration of the ammonia molecule can be modelled as a harmonic potential with two stable equilibria (representing the nitrogen atom on the left and on the right of the ammonia molecule), along with a Gaussian potential barrier in between. That is, we have:

$$V(x) = \frac{1}{2}kx^2 + ce^{-bx^2} \tag{1052}$$

Where k, c, b are some constants to fit to empirical data. Solving the time-independent Schrödinger equation $\hat{H}\psi = E\psi$ allows us find the allowed energy levels, from which we calculate the frequency of the umbrella inversion transition to be 24 GHz, or in terms of wavelength, 1.26 cm.

Remember that for stimulated emission to occur, we must first bring the gain medium (in this case, the ammonia molecules) to an excited state. We may accomplish this by utilizing the fact that at any one time, the ammonia molecule may be in either one of its two aforementioned states. So, on average, in a certain quantity of ammonia gas, there will be some ammonia molecules in the ground state, as well as some ammonia molecules in the excited state. Other than having different energies, the two states are also distinct in another respect: they possess a different **electric dipole moment**. Thus, if we apply an electric field, we can separate the ammonia molecules in the excited state from those in the ground state, without the need for any pump source! This is known as **state selection**, since the electric field "selects" the ammonia molecules in the excited state.



The general design of an ammonia maser (source). Note: the "focusser" (sic.) contains both the collimator and the electric field that selects out the atoms in the higher-energy state.

Thus, the ammonia maser is designed as follows. Ammonia gas, stored in a container, is slowly discharged and flows through a collimator that keeps it focused in a tight beam. An electric field is then applied to separate the ammonia molecules in the excited state from those in the ground state. This means that only the ammonia molecules that are in the excited state are fed into the optical cavity (technically, microwave cavity) of the maser. The cavity is a *resonant cavity*, meaning that is constructed to restrict the electromagnetic waves within the cavity to those that match the resonant frequency of the transition - that is, 24 GHz. Once in the resonant cavity, the excited ammonia molecules decay to their ground state by the process of stimulated emission, emitting photons in the microwave range as a result (in more familiar terminology, *microwaves*). Those emitted photons, bolstered by the resonant characteristics of the cavity, continue to reflect within the chamber and re-excite ammonia molecules to their excited state, ensuring that stimulated emission continues.[24]

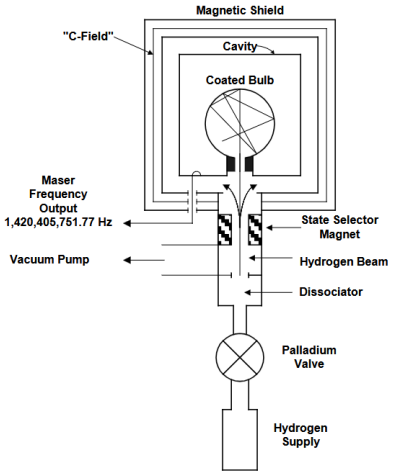
Note
We say "photon" and "electromagnetic wave" interchangeably here, because electromagnetic waves *are* (the classical description of) photons.

[24] https://bingweb.binghamton.edu/~suzuki/QuantumMechanicsFiles/7-3_Maser_physics.pdf

We have therefore achieved a **population inversion**, where most of the ammonia molecules in the resonant cavity are in their excited state, which allows stimulated emission to take place continually. The output coupler allows some of the microwaves to pass through, giving the ammonia maser its characteristic microwave beam. This beam, however, is extremely weak, in some cases on the order of nanowatts[25], so the ammonia maser is mostly relegated to scientific research and has very few other applications.

The hydrogen maser The hydrogen maser is very similar to the ammonia maser: it uses a gas as its gain medium, emits microwaves (though of a longer wavelength), it requires no pump source, and it produces a very weak beam. However, its gain medium uses atomic hydrogen rather than ammonia, and it operates on a different type of transition, known as the **hyperfine transition**.[26]

The hyperfine transition in the hydrogen atom results from a tiny energy level just above the ground state, caused by the spin of its electron. Even when it is in its ground state, the hydrogen atom possesses a slightly different energy depending on whether its electron is spin-up or spin-down. The hyperfine transition occurs when the electron "flips" its spin, meaning it changes from spin-up to spin-down (or spin-down to spin-up). The energy difference between the two states is about 5.87 μeV, so the transition wavelength is about 21 cm (corresponding to a frequency of 1.43 GHz). This is also known as the **hydrogen line** and is very important in astronomy, but we will focus only on its relevance to the hydrogen maser here.



The general design of a hydrogen maser (source).

In a hydrogen maser, hydrogen gas is slowly discharged from a container, where, just like in the ammonia maser, it is collimated. Then, an electrical discharge is passed through the gas, ionizing the molecular hydrogen ($H2$) into atomic hydrogen (individual atoms of hydrogen). Within the hydrogen gas, there will be some hydrogen atoms in the spin-up state, and some in the spin-down state. Crucially, hydrogen atoms in the spin-up and spin-down electron states are deflected differently in the presence of a magnetic field; thus, a magnetic field is used to separate hydrogen atoms in the excited state from those in the lower-energy state. Only the hydrogen atoms in the excited state are fed into a resonant cavity, which, similar to the ammonia maser's resonant cavity, is tuned to 1.43 GHz. Once again, stimulated emission takes place; the output coupler lets out a fraction of the microwaves, producing a very stable microwave beam - so stable, that hydrogen masers are frequently used as high-precision clocks.[27] The microwave beam produced by hydrogen masers is even weaker than that of ammonia masers, and is often around only 1 picowatt (or even lower!).[28] Thus, they are useful for only a few applications, such as (again) scientific research and timekeeping.

[25] https://physics.aps.org/story/v15/st4
[26] https://bingweb.binghamton.edu/~suzuki/QM_Graduate/Hyperfine_splitting.pdf
[27] Hydrogen maser
[28] https://ivscc.gsfc.nasa.gov/meetings/tow2013/Diegel.MW.pdf

Note
It is important to remember that while the ammonia maser and hydrogen maser do not require a pump source, they still need to be powered to maintain a population inversion. For instance, in the case of the ammonia maser, the collimator and electric field source used for state selection must be powered. Meanwhile, in the case of the hydrogen maser, the electrical discharge used to produce atomic hydrogen from molecular hydrogen and the state-selector magnets also need to be powered, among other supporting equipment. If masers could operate without being powered, we would break the laws of thermodynamics!

The HeNe laser The HeNe (helium-neon) laser is the first type of four-level laser we will encounter, and also the first type we have seen that uses optical pumping.

See: https://chem.libretexts.org/Courses/Grinnell_College/CHM_364%3A_Physical_Chemistry_2_(Grinne
Neon_Laser

0.2.9 Microwave engineering

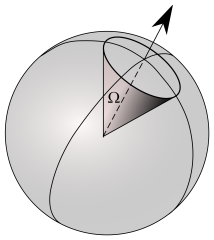
Antenna theory

At the beginning of our guide to our research, we went over the design of the ground-based power receivers, although our discussion was mostly conceptual. We will now introduce a *theoretical analysis* of the receiver stations, using the methods of antenna theory.

Antenna engineering To be able to perform detailed theoretical analysis on the ground-based receiver stations, we must first understand the general ideas of *antenna theory*, as the receiver stations are arrays of large parabolic antennas. We recall that all electromagnetic waves are solutions of *Maxwell's equations*. To solve for the electric and magnetic fields in the vecinity of an antenna, the full Maxwell equations - which can only be solved numerically in the vast majority of cases - are necessary for getting the most accurate results. However, there are some important analytical results and formulas we can derive before needing to reach for a computer, and these come from the field of electrical engineering.

> **Note**
> Before we start, a useful fact to keep in mind is that an antenna can generally be run both in *receiving* and *transmitting* mode (with a few modifications of its supporting circuitry/electronics, but not to the antenna itself). This means that a receiving antenna is just a transmitting antenna run in reverse. Thus, all results derived for receiving antennas also apply for transmitting antennas, and vice-versa.

To begin our study of antenna theory, let us introduce some terminology that is commonly-used in antenna theory. First, we introduce the **solid angle**. Think of a flashlight - note how its light spreads out in a cone. The spread of that cone can be measured in terms of solid angles, just like the spread between two lines (or vectors) can be measured in terms of regular angles. The unit of solid angle used in electrical engineering is the **steradian**, and just like radians, it is a dimensionless quantity. A full hemisphere is 2π steradians, and a sphere (the steradian equivalent of 360 degrees but in 3D) is 4π steradians. A visual example of the steradian is shown here:



A steradian describes a cone-shaped field of view, referred to as the *solid angle*.

We define the **radiation intensity** U of electromagnetic waves as the power carried in the wave passing through a given solid angle. We can calculate U from the magnitude of the Poynting vector S via:

$$U = Sr^2 \tag{1053}$$

Note that in general U depends on direction depends on direction, and using the the azimuthal and polar angles θ and ϕ (think longitude and latitude), we may write $U(\theta, \phi) = S(\theta, \phi)r^2$. It has units of W/sr (watts per steradian).

Now we focus on more antenna-specific terminology. An antenna's **radiation pattern** is a normalized function used to visually represent the intensity of the electromagnetic waves produced by an antenna in every direction. The radiation pattern function $\text{Rad}(\theta, \phi)$ is given by:

$$\text{Rad}(\theta, \phi) = \frac{U(\theta, \phi)}{U_{\max}} \tag{1054}$$

Since the radiation pattern depends on the radiation intensity which depends on the Poynting vector's magnitude, it requires finding an analytical (in rare cases) or numerical (most common) solution to Maxwell's equations - more on numerical methods soon.

Another important quantity of antennas is their **gain**. When we first examined two analytical wave solutions of Maxwell's equations, we looked at plane waves and more realistic waves that falloff $\propto 1/r$ - the proper term for them is **spherical waves**. Far away from the source of the wave, spherical waves are a good approximation, and even plane waves are often sufficient. But no perfect spherical waves exist in the universe, because of Birkhoff's theorem in electromagnetism. In addition, no antenna is perfectly efficient - all real antennas have some losses. So while real antennas can have *approximately* spherical wavefronts that move equally outward in all directions (we call the perfect case an *isotropic* radiator), spherical waves are an idealized approximation, a better one than plane waves, but still not physically possible.

Which brings us to the gain. The gain denoted G, is the radiated power of an antenna *relative* to an ideal, lossless, isotropic radiator (which again, bears repeating, is not physically possible and is an idealization). In general, it is also a function of direction, that is, $G = G(\theta, \phi)$. Given an antenna be fed with P_I watts of power (I for input power), its radiated power P_O over a certain area $A = \int dA$ is given by:

$$P_O = P_I G(\theta, \phi) \iint \frac{1}{4\pi r^2} dA \tag{1055}$$

Mathematically, gain can be written as a ratio between the power radiated by an antenna and the power radiated by an ideal isotropic radiator (or analogously, the ratio between their respective radiation intensities). That is to say:

$$G = \frac{P_O}{P_{\text{iso}}} = \frac{U(\theta, \phi)}{U_{\text{iso}}} = \frac{U(\theta, \phi)}{P_I/4\pi} = 4\pi \frac{U(\theta, \phi)}{P_I} \tag{1056}$$

Where in the previous equation we used the fact that an isotropic radiator has $U = P_I/4\pi$ as its radiation intensity, because remember, radiation intensity is **power per unit solid angle**, not power per unit area. All of these compound definitions means that we can write the gain in terms of the magnitude of the Poynting vector S via:

$$G(\theta, \phi) = 4\pi \frac{Sr^2}{P_I} \tag{1057}$$

And thus we can write the radiated power P_O as:

$$P_O = 4\pi S r^2 \iint \frac{1}{4\pi r^2} dA = Sr^2 \iint \frac{1}{r^2} dA \tag{1058}$$

Luckily, there is an analytical expression for the maximum gain of a *parabolic* antenna (remember this is the same whether the antenna is run as a receive or transmitter), and parabolic antennas are the main type the microwave-based power transfer approach uses. While parabolic antennas have complicated gain functions, the *maximum* of their gain function is given by a simple expression:

$$G_{\max} = \left(\frac{2\pi R}{\lambda}\right)^2 e_A \tag{1059}$$

Where e_A is a constant known as the *effective aperture*, typically marked on the antenna by the manufacturer, but reasonably well-approximated by a value between 0.5-0.7 (this is the typical range of aperture efficiencies), and can be found by doing controlled tests with the antenna. Also: note that the gain is often given in decibels, which are a dimensionless unit expression for logarithmic scales, where $G_{\text{dB}} = 10 \log_{10} G$. This is because raw gain measurements can quickly explode into the millions for large parabolic antennas and so to keep numbers within a reasonable range, it is better to work with decibels than the pure numerical value of the gain.

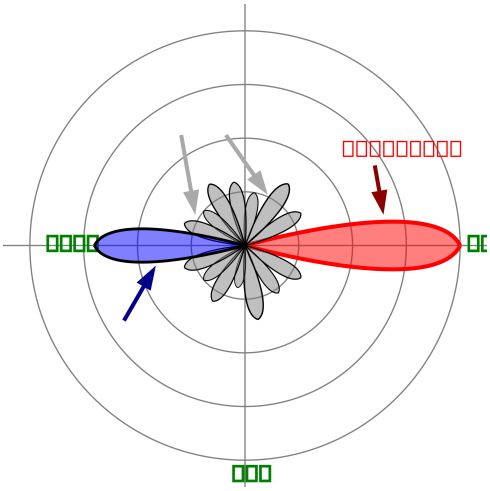
For a parabolic antenna, there exists an analytical solution for the radiation pattern (one may read more on this solution on Wikipedia), and it is given by:

$$\text{Rad}(\theta) = \frac{2\lambda}{\pi D} \frac{J_1(2\pi R/\lambda)\sin\theta}{\sin\theta} \tag{1060}$$

Where $J_1(\theta)$ is a Bessel function of the first kind, defined by:

$$J_1(\theta) = \frac{1}{\pi}\int_0^\pi \cos(n\tau - \theta\sin\tau)\ d\tau \tag{1061}$$

Here's a plot of the radiation pattern, courtesy of Comprod, a seller of antennas and other digital equipment, notice they use decibels otherwise the gain would be extreme and be almost entirely directed towards the direction the parabolic antenna is facing:



A parabolic antenna has a specific pattern in the intensity (electromagnetic power density) of its radiation. Source: Wikipedia

The crucial parameter to successful power transfer is to maximize the gain in the direction of reception for the ground-based receiver station antennas. This involves two things:

1. Reducing losses in general to increase efficiency, as greater inefficiencies means lower gain
2. Doing multiple iterations of the antenna design, computing the radiation pattern each time, and trying to tighten the radiation pattern around the transmission direction until it is highly focused without the "lobes" that protrude outwards and represent lost power.

Antenna simulations With an overview of antenna theory, we have gained a basic understanding that we can apply to analyze our power receiver stations. But all theoretical models are idealizations, and while they serve as good sanity-checks and provide analytical results, these often require heavy use of approximations or only apply in idealized scenarios. Further, some problems are completely unsolvable by using purely analytical means. This is why utilizing numerical methods to perform **computer-based simulations** are such a big part of our work at Project Elara.

We will go into the *general theory* of numerical methods for PDEs later, but we will begin with a direct application of a numerical method for designing and simulating antennas: using **finite element analysis** to solve the electromagnetic wave equation and Helmholtz equation.

Finite element analysis Finite element analysis (also called the *finite element method*) is the general name for a wide class of approaches that solve partial differential equations by approximating the solution with piecewise functions. The goal of finite element analysis is to find the correct coefficients for the piecewise function(s) that satisfies an integral form of the PDE. This particular integral form is called a *weak form* (or *variational form*), and relies on using methods from vector calculus to rewrite a PDE in a specific form that is easy to integrate by computer. Discretizing the integrals results in a system of linear equations, which can be solved using the methods of numerical linear algebra.

Warning
This subsection is very mathematically-heavy, so feel free to skip this section if you don't plan to work on simulations or other applications relying on numerical methods.

As a reminder, our PDEs to be solved for are two wave equations for the electric and magnetic fields:

$$\frac{\partial^2}{\partial t^2}\begin{pmatrix}\mathbf{E}\\ \mathbf{B}\end{pmatrix} = c^2\nabla^2\begin{pmatrix}\mathbf{E}\\ \mathbf{B}\end{pmatrix} \tag{1062}$$

As well as the Helmholtz equation for the electric and magnetic fields (note that these are for the *time-independent components* of each field, unlike the wave equation, which is *time-dependent*):

$$\nabla^2\mathbf{E}(\mathbf{r}) + k^2\mathbf{E}(\mathbf{r}) = 0 \tag{1063}$$
$$\nabla^2\mathbf{B}(\mathbf{r}) + k^2\mathbf{B}(\mathbf{r}) = 0 \tag{1064}$$

To perform finite element analysis, we must perform three general steps:

1. Rewrite the PDE + boundary conditions (which is called the **strong form**) into an equivalent integral equation (which is called the **weak form**)
2. Simplify the weak form as much as we can with vector calculus identities
3. Input the weak form into a finite element software of choice, plus the boundary conditions, then let the software solve for us

To get the weak form from the wave equation, we multiply both sides of each PDE by a test function Φ which is vector valued, then integrate over the 3D spatial domain Ω that includes everywhere within the boundaries. Note that we did not include time in the domain even though there are time derivatives - to do so would be computationally expensive. Rather, we can take out the time derivatives, and leave them there for now, we will replace the time derivatives with a numerical approximation later. We also have the helpful fact that the two PDEs are identical in form, so we will just work with the electric field wave equation, because the results for the magnetic field wave equation are identical other than replacing $\mathbf{E}$ with $\mathbf{B}$. Let's restate the wave equation for electric fields:

$$\frac{\partial^2\mathbf{E}}{\partial t^2} = c^2\nabla^2\mathbf{E} \tag{1065}$$

Remember that this is written in vector calculus notation and expands to:

$$\frac{\partial^2\mathbf{E}}{\partial t^2} = c^2\left(\frac{\partial^2\mathbf{E}}{\partial x^2} + \frac{\partial^2\mathbf{E}}{\partial y^2} + \frac{\partial^2\mathbf{E}}{\partial z^2}\right) \tag{1066}$$

Which is itself shorthand for a system of three PDEs:

$$\frac{\partial^2 E_x}{\partial t^2} = c^2\left(\frac{\partial^2 E_x}{\partial x^2} + \frac{\partial^2 E_x}{\partial y^2} + \frac{\partial^2 E_x}{\partial z^2}\right) \tag{1067}$$
$$\frac{\partial^2 E_y}{\partial t^2} = c^2\left(\frac{\partial^2 E_y}{\partial x^2} + \frac{\partial^2 E_y}{\partial y^2} + \frac{\partial^2 E_y}{\partial z^2}\right) \tag{1068}$$
$$\frac{\partial^2 E_z}{\partial t^2} = c^2\left(\frac{\partial^2 E_z}{\partial x^2} + \frac{\partial^2 E_z}{\partial y^2} + \frac{\partial^2 E_z}{\partial z^2}\right) \tag{1069}$$
$$\tag{1070}$$

When we're doing mathematical manipulations, it helps to not get too lost in notation and lose track of the underlying things we're operating on. In addition, when working with vector-valued PDEs or PDE systems, it is *much-preferred* to use tensors. Here, the Einstein summation convention is implied, and everything is assumed to be in a Euclidean space (so upper and lower indices are related by $A^j = A_i \delta^{ij}$ where $\delta^{ij} = \delta_{ij}$ is the Kronecker delta. Writing the electric field wave equation in tensor form, we have:

$$\partial_t^2 E^i = c^2 \partial^j \partial_j E^i \tag{1071}$$

Where $\partial_t^2 = \dfrac{\partial^2}{\partial t^2}$. Again, we will treat time derivatives separately via difference quotient approximations so we will just leave them there and not try to simplify the left-hand side. After all, the time dimension is a very regular one-dimensional domain and easy to use just a finite difference with, unlike the highly irregular spatial domain. We multiply (vector multiplication is dot product here) by vector test function Φ_i to everything, so we get:

$$\Phi_i \partial_t^2 E^i = c^2 \Phi_i \partial^j \partial_j E^i \tag{1072}$$

And then we integrate everything to get:

$$\int_\Omega \Phi_i \partial_t^2 E^i \; dV = c^2 \int_\Omega \Phi_i \partial^j \partial_j E^i dV \tag{1073}$$

Theoretically speaking, this is fine for an integral equation. But for solving, we would rather want a lower-order - ideally, first order - expression, and reduce the dimensionality of the integrals which improves computational performance. You could look up a table of vector calculus identities. Or, you can do it purely with tensors, by using integration by parts. Recall that the product rule is given by:

$$\frac{\partial}{\partial x^a}(uv) = v \frac{\partial u}{\partial x^a} + u \frac{\partial v}{\partial x^a} \tag{1074}$$

Be careful to note that all terms of the product rule must use the *same* index. You cannot have a ∂_a in one term and ∂_b in another term. Writing in compact tensor notation, we can rewrite as $\partial_a(uv) = v\,\partial_a u + u\,\partial_a v$, which we can rearrange to $u\,\partial_a v = \partial_a(uv) - v\,\partial_a u$. The expression we want to simplify is $\Phi_i \partial^j \partial_j E^i$, which is in the form $u\,\partial_a v$ where $u = \Phi_i$ and $\partial_a v = \partial^j \partial_j E^i$. So $v = \partial_j E^i$ and $\partial_a u = \partial^j \Phi_i$, and thus:

$$\Phi_i \partial^j \partial_j E^i = \partial^j (\Phi_i \partial_j E^i) - (\partial_j E^i)(\partial^j \Phi_i) \tag{1075}$$

We can drop the brackets on the second term, so long as we remember that it is a product:

$$\Phi_i \partial^j \partial_j E^i = \partial^j (\Phi_i \partial_j E^i) - \partial_j E^i \partial^j \Phi_i \tag{1076}$$

Now, substituting this back into the right-hand side term, we obtain:

$$\int_\Omega \Phi_i \partial_t^2 E^i \; dV = c^2 \int_\Omega \Phi_i \partial^j \partial_j E^i dV = c^2 \int_\Omega \partial^j (\Phi_i \partial_j E^i) - \partial_j E^i \partial^j \Phi_i \, dV \tag{1077}$$

Using the Kronecker delta on the right-hand side term, we can rewrite $E^i = \delta^i{}_j E^j$. So:

$$\int_\Omega \Phi_i \partial_t^2 E^i \; dV = c^2 \int_\Omega \partial^j (\Phi_i \partial_j E^i) - \partial_j E^i \partial^j \Phi_i \, dV \tag{1078}$$

We can split the right-hand side integral into two for convenience:

$$\int_\Omega \Phi_i \partial_t^2 E^i \; dV = c^2 \int_\Omega \partial^j (\Phi_i \partial_j E^i) \, dV - c^2 \int \partial_j E^i \partial^j \Phi_i \, dV \tag{1079}$$

Notice the term $\partial^j(\Phi_i \partial_j E^i)$. We can write it as $\partial^j B_j$ where $B_j = \Phi_i \partial_j E^i$. But by the Divergence Theorem:

$$\int_\Omega \partial^j B_j \, dV = \int_{\partial\Omega} B_j dA^j \tag{1080}$$

This may look more familiar when written in standard vector calculus notation:

$$\int_\Omega \nabla \cdot \mathbf{B}\ dV = \int_{\partial\Omega} \mathbf{B} \cdot d\mathbf{A} \tag{1081}$$

This is great because we can reduce the dimensionality of the integral from a volume integral to a surface integral! Therefore we arrive at:

$$\int_\Omega \Phi_i \partial_t^2 E^i\, dV = c^2 \int_{\partial\Omega} \Phi_i \partial_j E^i\, dA^j - c^2 \int_\Omega \partial_j E^i \partial^j \Phi_i\, dV \tag{1082}$$

Which we can write in more typical notation as:

$$\int_\Omega \Phi \cdot \frac{\partial^2 \mathbf{E}}{\partial t^2}\ dV = c^2 \left[\int_{\partial\Omega} \Phi \cdot \nabla_J \mathbf{E}\ d\mathbf{A} - \int_\Omega \nabla_J \mathbf{E} : \nabla_J \Phi\ dV \right] \tag{1083}$$

Where : is the double dot product (tensor product for matrices). If we move all the terms to one side, we have:

$$\int_\Omega \Phi \cdot \frac{\partial^2 \mathbf{E}}{\partial t^2}\ dV - c^2 \int_{\partial\Omega} \Phi \cdot \nabla_J \mathbf{E}\ d\mathbf{A} + c^2 \int_\Omega \nabla_J \mathbf{E} : \nabla_J \Phi\ dV = 0 \tag{1084}$$

Notice that we have a time derivative on the left. As we integrate only over space, we must *timestep* our weak form, meaning we have to discretize the derivative and compute the weak form for every discrete t. Using a table of derivative approximations, such as on this website, we may choose the following approximation for the second derivative:

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} = \frac{\mathbf{E}(t_{n+1}, \mathbf{x}) - 2\mathbf{E}(t_n, \mathbf{x}) + \mathbf{E}(t_{n-1}, \mathbf{x})}{\Delta t^2} \tag{1085}$$

We have to be careful because the $\mathbf{E}(t, \mathbf{x})$ that appears in the variational form is actually the unknown next time-step $\mathbf{E}^{n+1}$ that we want to calculate. So in order to make this explicit, the variational form is given by:

$$\int_\Omega \Phi \cdot \frac{\mathbf{E}^{n+1} - 2\mathbf{E}^n + \mathbf{E}^{n-1}}{\Delta t^2}\ dV - c^2 \int_{\partial\Omega} \Phi \cdot \nabla_J \mathbf{E}^{n+1}\ d\mathbf{A} + c^2 \int_\Omega \nabla_J \mathbf{E}^{n+1} : \nabla_J \Phi\ dV = 0 \tag{1086}$$

This is the *most general* simplified weak form of the wave equation. Beyond this point, we need to individually-substitute each of the boundary conditions into the variational form, which is dependent on the specific electromagnetics problem, and the rest is software-specific. For inputting these weak forms into software, it is important to know the four categorizations of weak-form terms:

- **Bilinear terms** are terms that involve the function to be solved for
- **Linear terms** are terms that *don't* involve the function to be solved for
- **Domain terms** are terms integrated over Ω (the entire domain)
- **Boundary terms** are terms integrated over $\partial\Omega$ (the boundary of the domain)

We must take special care when identifying the terms in the wave equation, because the terms are not as simple as they seem. As $\mathbf{E}^{n+1}$ is the function we're computing on every time-step based on the calculated result $\mathbf{E}^n$ from the previous time-step, all terms that involve $\mathbf{E}^{n+1}$ are bilinear (as we are solving for that), while all the terms that contain $\mathbf{E}^n$ and $\mathbf{E}^{n-1}$ are linear (as we are not solving for them). *(Yes, this is quite confusing).*

We should also note that by applying the same process to the Helmholtz equation $\nabla^2 \mathbf{E} + k^2 \mathbf{E} = 0$ we can also find its weak form, which we will not derive to avoid making this page overly long. The result is:

$$-\int_\Omega \nabla_J \mathbf{E} : \nabla_J \Phi\, dV + k^2 \int_\Omega \Phi \cdot \mathbf{E}\, dV + \int_{\partial\Omega} \Phi \cdot \nabla_J \mathbf{E}\, d\mathbf{A} = 0 \tag{1087}$$

After substituting the boundary conditions into the problem and setting the domain, the weak form of the PDE can be substituted into a variety of software packages designed to find numerical solutions by the finite element method. We use two major ones in Project Elara: FreeFEM, which is a

standalone software, and FEniCS, which is a Python library. Both software packages are open-source, feature-rich, and used extensively by scientists and engineers.

0.2.10 Astrodynamics

Space is truly the final frontier: an unforgiving environment for humans and machines alike. Spaceflight is dangerous, expensive, and technologically demanding, with no margin for error. To design successful satellites and manned spacecraft is naturally an incredibly daunting task. The following chapters will be a broad overview of concepts in astrodynamics, space vehicles, and satellite engineering.

Rocket physics

You know the famous phrase "it's not rocket science"? Rocket science is often thought of as an incredibly difficult field of engineering and physics, but this is not entirely true. It is possible to get a good basic understanding of rocket physics - certainly not enough to work on real, skyscraper-sized rockets, but enough to be well-versed in the general concepts. This is what we aim to provide in this chapter on rocket physics

The rocket equation

$$\Delta v = I_{sp} g_0 \ln\left(\frac{m_0}{m_f}\right) \tag{1088}$$

Gravity and orbital mechanics

Given that we plan to construct complex space-based megastructures, we must of course design our system *for* space, which comes with its unique challenges. Gravity is the primary force that determines the trajectories of objects in deep space, and so is something we have to give careful consideration to. Newtonian mechanics, luckily, gives a very good mathematical description of gravity that is sufficient for *preliminary* calculation purposes. The differential equation of motion for all solar orbits under Newtonian gravity is:

$$\frac{d^2\mathbf{r}}{dt^2} + \left(\frac{GM}{r^2} + \sum_i \frac{Gm_i}{|r_i - r|^2}\right)\hat{r} = 0 \tag{1089}$$

Where M is the mass of the Sun, and $m_1, m_2, \ldots, m_i, r_1, r_2, \ldots, r_i$ are the masses and positions of all the non-Earth gravitating bodies in the Solar system. The Sun and the 8 planets together account for 99.985% of the total mass of the Solar system, with the Sun itself being 99.85% of the total mass. Meanwhile, the differential equation of motion for a geosynchronous orbit around Earth, under Newtonian gravity, would be given by:

$$\frac{d^2\mathbf{r}}{dt^2} + \frac{GM_{\text{earth}}}{r^2}\hat{r}_{\text{earth}} + \frac{GM_{\text{sun}}}{{r_{\text{sun}}}^2}\hat{r}_{\text{sun}} + \frac{GM_{\text{moon}}}{{r_{\text{moon}}}^2}\hat{r}_{\text{moon}} = 0 \tag{1090}$$

Where r is the distance from Earth to the orbiting satellite, r_{moon} is the distance from the satellite to the moon, r_{sun} is the distance from the satellite to the Sun, and each of the basis vectors represents the unit vector of the displacement vector pointing between the Earth and that particular celestial body (for instance, $\hat{r}_{\text{sun}}$ is the displacement vector between the satellite and the Sun).

If use the approximation that Earth's gravity is dominant in this scenario, we may ignore the lunar and solar terms, and together with using the fact that orbits are planar, we have the system of equations:

$$\frac{d^2x}{dt^2} + \frac{GM_{\text{earth}}}{(x^2+y^2)^{3/2}} = 0 \tag{1091}$$

$$\frac{d^2y}{dt^2} + \frac{GM_{\text{earth}}}{(x^2+y^2)^{3/2}} = 0 \tag{1092}$$

$$\langle x_0, y_0\rangle = r_0\hat{\theta}, \langle v_{0x}, v_{0y}\rangle = v_0\hat{\theta} \tag{1093}$$

These aren't easy to solve but there *is* an analytical solution if you convert to polar coordinates, do a substitution of variables, and then use some other math tricks. The solution is given by:

$$r = \frac{L^2}{m^2}\frac{1}{GM(1 + e\cos\theta)} \tag{1094}$$

Where $h = L/m$, L is the angular momentum, m is the satellite's mass, and M is the Earth's mass, and we can get typical Cartesian coordinates with the typical $x = r\cos\theta$ and $y = r\sin\theta$. e is a parameter known as the *eccentricity*, given by:

$$e = \sqrt{1 + \frac{2EL^2}{G^2M^5}} \tag{1095}$$

Using $L = I\omega$, $I = mr^2$ for the satellite, the fact that that the tangential velocity obeys $v_{\text{tang.}} = r\omega$, and the conservation of angular momentum, we find that:

$$L = mr^2\omega = mrv_{\text{tang.}} \tag{1096}$$

In our case, $v_{\text{tang.}} = v = \|\mathbf{v}\|$ where $\mathbf{v} = \langle v_x, v_y\rangle$ (this is because the tangent vector always points in the direction of motion and the direction of motion happens to be rotational). Conservation of angular momentum means we can equivalently write:

$$L = mr_0 v_0 \tag{1097}$$

By a similar approach using the conservation of energy, we have:

$$E = \frac{1}{2} m {v_0}^2 - \frac{GMm}{r_0} \tag{1098}$$

Which means that the complete solution to the Newtonian differential equations for Earth orbits is given by:

$$r(\theta) = \frac{{r_0}^2 {v_0}^2}{GM} \left(\sqrt{\frac{G^2 M^5 - r_0 \ m^3 \ v_0^2 \left(2GM - r_0 v_0^2\right)}{G^2 M^5}} \ \cos(\theta) + 1 \right)^{-1} \tag{1099}$$

Hence the reason why real-world problems almost never appear in an introductory treatment of differential equations. Further, this is not something to be computed by hand - we highly recommend sympad, an awesome open-source computer algebra system interface, which speeds up calculations tremendously.

However, it should be noted that this is a *simplified* gravitational model that works for limited purposes. First, it ignores the Moon completely, which is the biggest source of gravity other than the Sun and Earth. Second, it doesn't account for the fact that Earth's gravity is not technically uniform, with slight surface gravity variations and gravitational anomalies. Third, it does not incorporate any relativistic effects or the effects of radiation pressure or various other emissions from the spacecraft, which all have small - but tangible - effects on the spacecraft's orientation and trajectory through time, requiring orbital corrections. Finally, it does not take into account all the *other* gravitational influences caused by the uneven distribution of mass in the Solar System, and due to the fact that the n-body problem (that is, systems involving more than two gravitationally-interacting objects) is known to be chaotic, any orbit in the Solar System will decay over very long timespans. This is due to the tiny gravitational effects from other gravitational influences (e.g. comets, asteroids, dwarf planets) that we don't typically include in our calculations out of sheer complexity.

For this reason, while preliminary theoretical analysis is important, it is advisable to use professional orbital calculations software such as NASA's Copernicus and standard gravitational models such as the Earth Gravitational Models (EGM), which is part of the World Geodetic System for the final high-precision research calculations.

0.2.11 Fundamentals of research

With all the prerequisites covered, we're now ready to cover the actual research of Project Elara. Our research ties together many fields of physics and engineering, which is why we will explain it step-by-step. The following sections serve as both a learner's guide and reference guide to our research, and it is our honor to share it with the world.

Introduction to our research

Project Elara is not a single research project; it is a collection of multiple research projects managed under the umbrella of the same organization, which also leads its research alliances and conducts its partnerships. However, the priority research project conducted by Project Elara is its **universal power initiative**. This research focuses on the technologies crucial to building a system of power collection and distribution designed to endure for centuries and provide free (or nearly free), universal, and abundant energy to all corners of the globe.

The central portion of this technology is the development of **space power swarms**. We are trying to bring the energy of the Sun to Earth. The technical term is space-based solar power, but it might as well be called stellar energy harvesting at the scale we're planning to do it at. The end goal is a source of energy from space so powerful that we would have more abundant energy than we would ever need.

There are two key numbers that are central to this entire mission. The Sun is a star, which means it is a nuclear furnace. A statistic we will quote frequently is the following: the Sun emits 3.846×10^{26} watts of power from nuclear fusion, while Earth's 2023 global power consumption was, on average, 2.049×10^{13} watts. We emphasize: to collect *even a fraction* of the Sun's total power output would not just be sufficient for the energy needs of humanity, but orders of magnitude beyond.

The question then becomes, how do we collect this power? More pragmatic people have already made solutions - solar panels and concentrated solar thermal power. The first doesn't need any explanation; the second works rather like a heat ray that is used to run a heat engine. But these solutions have bottlenecks that inherently limit them from collecting any substantial fraction of solar power.

First, **solar availability**. The Earth is not right next to the Sun; rather, it is an astronomical unit away, or around 150 million kilometers. Recall that due to energy conservation, $P \propto 1/r^2$, that is, power decays by the inverse square of distance. This would not be as much of an issue, if not for the fact that most regions do not have constant sunlight at all times. Even more so, solar power stations on Earth are typically limited in size: among the largest non-fossil fuel power plants, we see nuclear power stations and hydro-power stations, but not really solar power.

But neither is necessarily a dealbreaker. The *biggest* issue inherent to all terrestrial solar power plants is that the Sun is not available at night, during storms, or on cloudy days - and for tropical regions, heavy rainfall and typhoons/monsoons/hurricanes make solar power incredibly vulnerable to the weather. The Sun may shine bright, but it can only shine so bright when the atmospheric conditions stand in the way.

A space-based solar system would be purpose-built for one specific purpose: collect as much power as possible from the sun, and transmit it as efficiently as possible to Earth. Size is a blessing in the context of solar power, because $P \propto A$, that is, power is proportional to the receiving surface area. And space-based power solutions - especially when satellite constellations are used - can transmit solar energy to the ground 24 hours a day, with longer wavelengths of light that can pass through the atmosphere with nearly no loss of power, even in heavy rain or storms.

Second, **scale**. Earth-based structures have inherent limitations in where they can be built. Space-based structures, however, are theoretically unlimited in scale, and can be made to have minimal structural support, as they are in free-fall and (roughly speaking) do not need to withstand gravity. Their solar collection surfaces can be extremely large and thin, allowing them to gather as much sunlight as possible, which can then be focused into a tight beam for transmission.

And third, **equity**. This is not as much of a current concern, but a reliance on Earth's natural resources for energy has invited endless conflict and seizures of energy resources have left many countries behind. Placing energy sources in space does not change human nature, but it does allow for essentially unlimited distribution of energy to anywhere on earth. The ubiquity of GPS and other satellite communication systems is not without reason.

However, space comes with its own set of issues. The first is complexity: the intersection of space physics, material physics, aeronautical engineering, quantum optics, and electromagnetic theory is certainly not simple. The second is cost. Launching anything into orbit (at present) is *immensely* expensive, so every kilogram launched has to be worth its thousand-dollar price tag. However, cost is

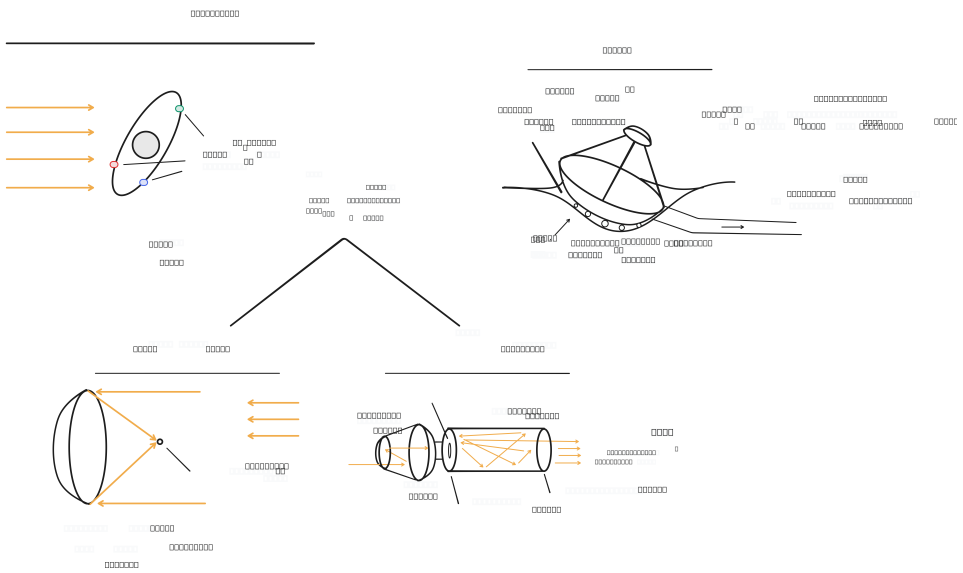
not as much of an issue *if the technology can be made worth that cost.* The third, of course, is that this has never been done before. It may be inspiring and adventurous, but that also makes it very risky.

This is why Project Elara was founded to not operate over a few years, or even one person's lifetime, but over many (possibly dozens) of generations, utilizing the best practices in data preservation to ensure that future generations can build on an ever-growing body of knowledge, and administratively designed to ensure continuity of leadership even through catastrophic events. This allows us to tackle the sorts of projects no one else could dare to do, the projects that truly could change the world.

Conceptual design

With all the mathematics and physics prerequisites covered, we can now introduce our proposed design for the solar power satellites. This section involves minimal mathematics, and is primarily a conceptual overview of the whole system that does not go into too much detail on the physics or engineering.

The system captures solar energy with with a massive solar collector array. We borrow an idea dating back to the time of Archimedes, but upgraded to the space age: using lots of curve mirrors that together form a giant reflector that focuses sunlight to a single point, something we can already do on Earth with impressive results. We then convert sunlight to microwaves, using the sunlight to optically pump powerful microwave lasers that transmit the power down to Earth. We use star-tracking and ground-tracking technology and a constellation of satellites to guarantee 100% coverage of the sky and 24-hour microwave transmission in all weather conditions to our terrestrial receiver stations. These stations use giant sunken parabolic antennas to receive the power, which can then be converted straightforwardly to electricity. We thought it might be easier to explain the system with a diagram, so here it is:



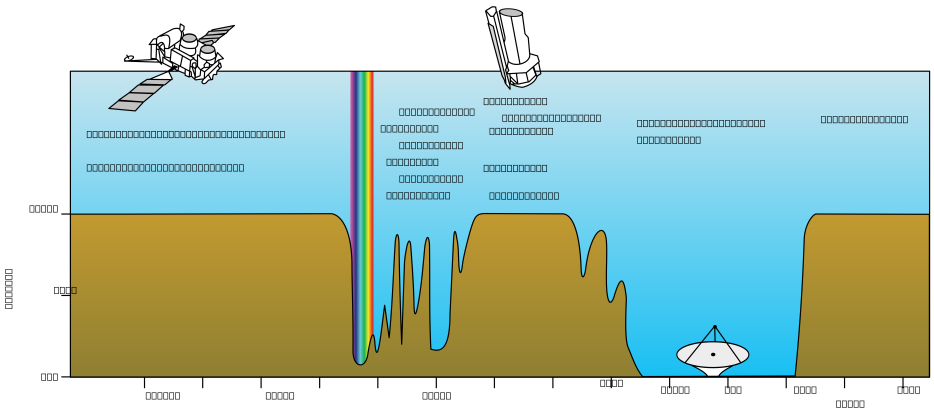
Our concept system's major components: the space mirror, power satellites, and power receiving stations

We will dive deep into each aspect of the system, as described on the diagram, in the following sections.

Choice of wavelength Our first design consideration is the wavelength of the light we use to transmit power we collect from space down to Earth. While it may seem obvious that visible light is a terrible choice, let us re-review the reason why.

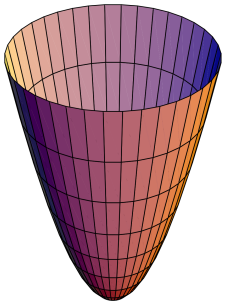
The primary issue that makes visible light highly unsuitable is *atmospheric attenuation*. Earth's atmosphere does not allow all wavelengths of light to pass through equally. The below plot of atmospheric opacity - that is, how much of each wavelength of light passes through the atmosphere - showcases this issue very clearly (source: Wikipedia):

Visible light is partially absorbed by the atmosphere in even the best of conditions; when considering stormy weather, clouds, rain, and other atmospheric obstructions, visible light is very ineffective for energy transfer. Infrared does not fare much better (in fact, it is actually much worse!), and neither do the shorter wavelengths of light like ultraviolet and X-rays, which we would not use anyway as they are ionizing radiation and dangerous. That leaves us with just microwaves and short to medium-wavelength radio waves, but the longer wavelengths of radio waves as compared to microwaves means that they need larger receiver antennas (we will cover why when we discuss antenna theory). Having eliminated all the other options, it becomes apparent that microwaves are our best bet.



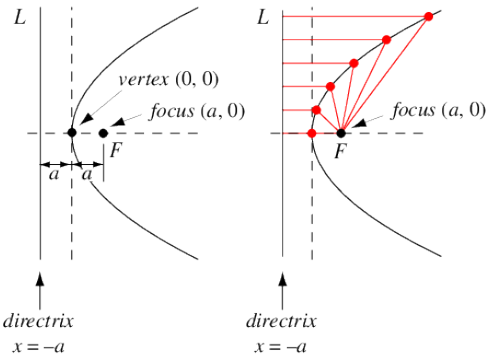
A plot of atmospheric attenutation across the electromagnetic spectrum. The atmosphere is near-transparent to long-wavelength microwaves. (Source: Wikimedia Commons)

Concentrating sunlight To concentrate solar energy - that is, sunlight, which is primarily in the visible range, with some ultraviolet and near-infrared - we use lots of space-based mirrors that together form a **paraboloid**. Recall that a parabola is a curve in the form $y = kx^2$. A paraboloid is simply a surface of revolution formed by rotating a parabola about the x-axis, or alternatively, the 3D surface $z = a(x^2 + y^2)$, where a is a constant:



A plot of a paraboloid as a 3D surface.

A parabola has the special property that all points along it are equidistant from a special point known as the **focus** and a special vertical line known as the **directrix**, see the below diagram:



The key components of a parabola. The horizontal distance from the directrix to every point along the parabola is equal to the distance from the parabola to that point.

This means that parallel lines converge on the focus, which is useful for directing plane waves of light to the focus, allowing a reflector shaped as a parabola to act as a solar energy concentrator. For a parabola that opens up along the y-axis, given by the function $y = x^2/4a$, where a is a constant,

then $(0, a)$ is the location of the focus, and $y = -a$ is the equation of the directrix. We call a the **focal length**, and it is the point at which all parallel rays directed at the parabola converge. Due to this property, all light incident on the parabolic reflector would be concentrated at the focus, producing a region of extremely concentrated energy. In fact, this is enough concentrated energy that portable parabolic reflectors are used in some regions of the world as a heat source for cooking, and need only a sunny day to reach temperatures of up to 400 degrees Celsius (source: Wikipedia).

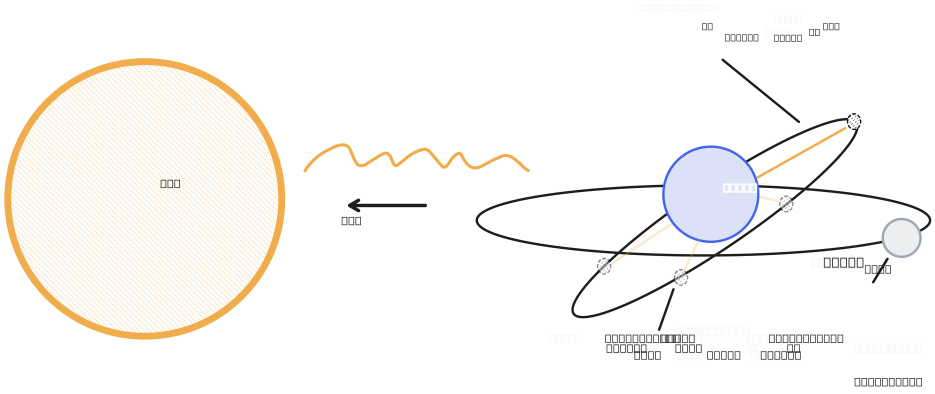
Geometrically, our design shares the same basic properties, only we would be doing this in space, with heliostats (smaller individual mirrors) that together form a space-based composite parabolic reflector to focus sunlight. The combined surfaces of the smaller mirrors acts as one very large parabolic reflector with a very large total collection area.

A concept render of the solar mirrors. See source files in the Elara art repository

Theoretically speaking, the more heliostats, the better, but due to practical considerations regarding the cost of launches, the first prototypes might have to make do with only a few heliostats, or alternatively, many but small heliostats. That is no issue, however; with time, we can always add more, after all, over many launches.

Orbital placement With the basic design of the solar mirror (i.e. composite parabolic collector) settled, we must now choose a suitable placement for its location, along with the power satellites that house the lasers to transmit the power back to Earth. Our proposed orbit is called a *geosynchronous orbit* (GSO). To understand geosynchronous orbits, we must first understand *geostationary orbits*. A geostationary orbit is a circular orbit around the Earth where the satellite orbits at just the right radius to complete one full orbit in the same time as one revolution of Earth (that is, one day). This means that any satellites placed in geostationary orbit can each track one point on Earth continuously, making them highly useful for telecommunications and broadcasting.

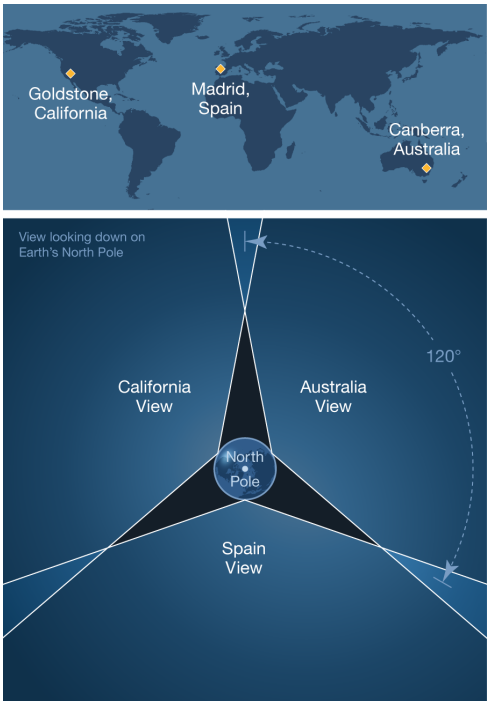
A geosynchronous orbit is similar to a geostationary orbit in that it matches Earth's rotation, but different in that instead of placing transmitting satellites at the same place above the equator, we place the collector and transmitting satellite together in an eccentric (angled) and elliptical (ellipse-shaped rather than circular) orbit, as shown in the figure below:



This configuration allows the collector and transmitting satellite to always be facing in full view of the Sun, unlike a geostationary orbit where the Earth blocks out the Sun for a significant duration of the orbit, while requiring minimal tracking. Due to its high eccentricity (i.e. going above and

below Earth's orbital plane), it also mostly avoids the issue of the Moon blocking out light. The only exception would be when the satellites pass the ecliptic (Earth orbital plane) when the Moon is at a point in its orbit where it is exactly between the satellites and the Sun, but this is easily countered by simply having a pair (or even constellation) of collectors and transmitters at opposite ends of the same orbit. Finally, this orbit is very well-understood and a stable orbit, so much so in fact that there is an analytical expression for it.

The one difficulty with a geosynchronous orbit, other than its sheer distance from the Earth, is that each laser on the power satellite must be attached to a gimbal mount so as to track different ground stations receiving power as the Earth rotates, and an automated computer system must switch between different power beams as one station passes out of view and another comes into view. This is because while the satellite always stays at the same *longitude* with respect to the Earth, its *latitude* continuously varies, so we must implement a station tracking system. As a reference for a similar existing system, this is a diagram showing the coverage of the three respective stations used for NASA's Deep Space Network (credit: NASA):



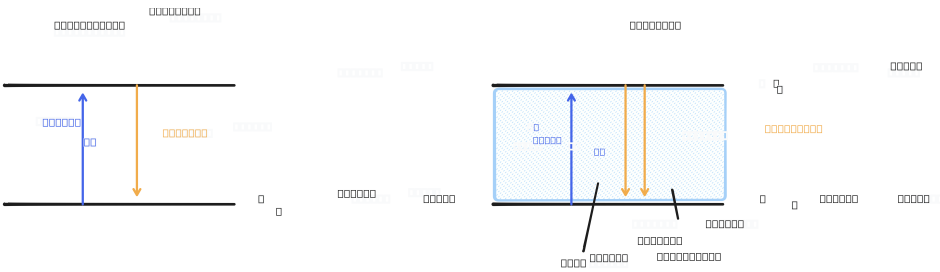
Each of the three installations of NASA Deep Space Network has a field of view of approximately 120 degrees.

In our case, each of the ground stations is designed to be able to track the power beams across the sky, ensuring complete coverage. The transmitter satellites must coordinate with the ground stations in keeping the beam accurately focused to higher than GPS precision as the Earth rotates. This complicates the already highly-complicated task of keeping a beam almost perfectly collimated (that is, making sure the beam travels in parallel rays) over immense distances, as any vibrations or misalignment of the mount could mean overshooting or undershooting the ground stations by hundreds of meters or even kilometers.

Transmitting power to Earth After collecting solar energy in deep space, we have to get the energy to earth in some way. This consists of two parts: the power satellite in space, and the receiver stations on Earth. In space, a power satellite is placed at the focus of the composite solar mirror, where the sunlight collected by the mirror is concentrated. The power satellite has a set of two mirrors facing the concentrated sunlight, a primary and a secondary mirror, to redirect the light inside the power satellite.

Here, the light enters into a chamber - the optical cavity of a laser, and it is used to optically pump a (or several) microwave lasers (masers) - lasers but with microwaves instead of visible light. This

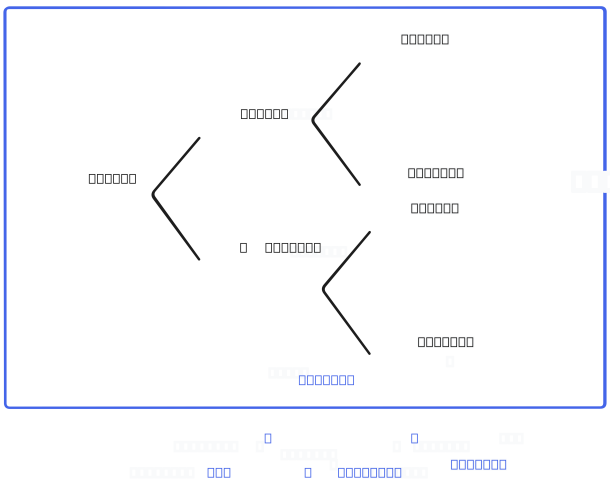
optical cavity is where the energy of the concentrated sunlight is converted into a powerful microwave beam.



A comparison between an atomic transition from the excited state to the grouind state in spontaneous emission and stimulated emission.

A maser - as with any laser - works by the principle of **stimulated emission**, which we will briefly review. Recall that in quantum mechanics, an atom is allowed to only take certain *states*. Each state corresponds to different momenta, different probabilities of occupying different locations in space, and crucially, different **energies**. Thus, we speak of *energy levels* in an atom, each energy level being a particular energy of a given state. If this is unfamiliar, feel free to re-read the subchapter on quantum mechanics, which we went through earlier in this chapter.

Consider an atom with two primary energy levels, which are each associated with a given state. These two states are the *ground state*, which is the lowest-energy state, and an *excited state*, which has a higher energy. Let us call the energy of the ground state E_1, and the energy of the excited state E_2. When an atom *absorbs* a photon, the energy of the photon is transferred into the atom, causing it to "jump" from the ground state to the excited state. **Stimulated emission** is one of two modes that allow atoms to change state (which we call an *atomic transition*), the other being **spontaneous emission**. Conventionally, atoms in an excited state transition back to the ground state through spontaneous emission, where the atom randomly emits a single photon of energy $\Delta E = E_2 - E_1$ without any external input. Thus, we say that it is a *spontaneous* (i.e. random) process. **Stimulated emission**, by contrast, requires that a photon pass towards an excited atom, triggering the transition and releasing a photon *identical to the received photon*. The incident (passing) photon is not absorbed by the atom, so we have a combined total of *two* photons released. Each of those two photons can stimulate other atoms, leading to a rapid emission of *identical* photons that gives lasers their monochromatic (i.e. single-frequency light) and beam-like character. We illustrate this in the diagram below:



In stimulated emission, 2 photons are emitted per transition, which start a chain reaction doubling the number of photons emitted every atom they strike.

Note that stimulated emission occurs predominantly when atoms are *raised to a higher-energy state* by some external energy source. In our case, the mirror-focused sunlight is our energy source, which is used to optically-pump (i.e. force energy into) atoms within the laser cavity. This powers our maser beam, which transfers the energy of the concentrated sunlight into microwaves that can be readily converted to electricity on Earth.

Receiving power As well as solar space mirrors to capture solar energy and power satellites to transmit it down to Earth, the system also needs ground-based receiver stations to receive the power from space. Each receiver station is composed of an array of large parabolic antennas housed in rotatable semi-sunken platforms. This allows them to each capture some portion of the power beam, then combine their beams together through waveguides that channel microwaves into one central hub. This allows the entire array to effectively act as one antenna, without requiring a single unrealistically-massive antenna.

Each parabolic antenna is designed to be a scaled-up version of traditional radio and telecommunication antennas, only with far more advanced tracking technology to stay in perfect alignment with the power satellites orbiting overhead, and strengthened design to be able to accomodate much-higher power microwaves. This means that the parabolic antennas, while complex in engineering terms, are not too different from traditional parabolic antennas in the sense that they share the same physics. We will cover the details of how to do antenna analysis and apply antenna theory later within this chapter.

More details This introduction is meant to only be an entry-level description of our system. Each component of the system is described by complex physics and mathematics, and the following sections will go through the details and delve deeper into our research.

Optical analysis

Having covered the background knowledge necessary, we will continue to elaborate on our design by focusing on the next essential aspect - light. The analysis of light is the domain of **optics**. Using an optics-based approach, we will now present a general sketch of how light is to be transported through the system.

The Gaussian Beam The standard model for the wave propagation of light with lasers is the **Gaussian beam model**. The Gaussian beam is a solution to the Maxwell wave equations and describes a propagating beam of electromagnetic waves - microwaves, infrared, visible light, etc. - that is focused along one direction, called the **axis of propagation**.

A key feature of a Gaussian beam is that with increasing distance, the beam grows wider and wider, similar to a flashlight's cone (although the spread is much more gradual). The angle θ at which the beam diverges from the axis of propagation is given by:

$$\theta = \frac{\lambda}{n\pi w_0} \tag{1100}$$

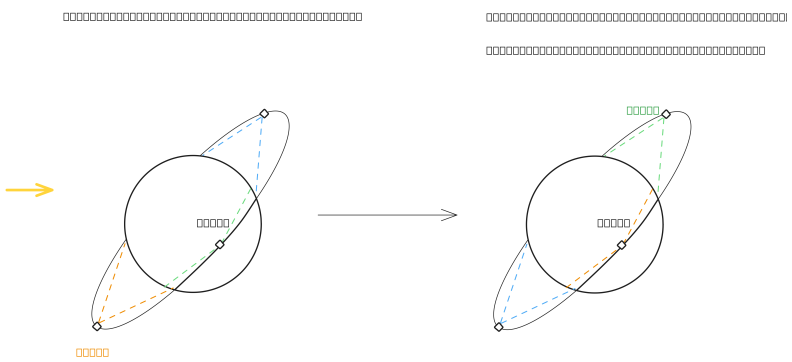
Where w_0 is the beam's radius at the light source, that is, at $x = 0$, n is the refractive index (for all intents and purposes we can use $n = 1$ as the laser is in space) and λ is the wavelength. In the case of a laser, the light (or microwaves) propagate out of an **aperture**. The aperture width A, which is the diameter of the beam source, is given by $A = 2w_0$.

The Gaussian beam model predicts that smaller apertures lead to faster divergence, meaning that the narrower you make the laser's beam at the aperture, the more quickly it will spread out. This means there is an inherent trade-off:

- You could build a *very small* aperture laser, which would have very focused light at short distances from the aperture (and thus a very narrow beam), but with rapid beam spreading and complete loss of focus at far distances
- Or you could build a *very large* aperture laser, which would have less-focused light at short distances from the aperture (and thus a very wide beam), but the beam spreads more slowly and remains focused at long distances
- Or you could build something in between, something that maximizes the benefits and minimizes the drawbacks of very-small- and very-large-aperture lasers; this is the *optimal* laser aperture

> **Note**
> The word **collimated** is the more technical word for *focused* and is the one that will be predominantly used going forwards.

Discussion on atmospheric attenuation The key reason for space solar power capture is that the rotation of the Earth hinders solar power during the night-time, conditions such as rain and clouds often obscure visible light, and the atmosphere absorbs - in more technical terms, *attenuates* - visible light passing through the atmosphere. The microwave portion of the visible spectrum avoids both of the latter issues. Microwave frequencies from 1 GHz-10GHz (equivalently, wavelengths between 3.75-30 cm) have no problem penetrating through the atmosphere, with under 10% loss even in rain and cloud conditions. As for the first issue, that of the Earth's rotation, a spacebound power satellite can continuously track a ground station as it orbits around the Earth. By using a satellite constellation, three power satellites are sufficient to cover nearly the entire Earth, and by continuously shuttling between the three power satellites, each receiver station on Earth will always have a power satellite overhead and is able to receive uninterrupted power. This last point requires some elaboration as it may be rather difficult to grasp from a textual description alone. Therefore, here is a diagram to explain it instead:



The satellite constellation ensures that there is always at least one satellite above every power station.

> **Note**
> In addition, each satellite is planned to be equipped with multiple gimbal-mounted laser apertures that can be opened and closed on command, meaning that each satellite can transmit and track **multiple** ground stations within its field of view at once, which can eventually be up to dozens or perhaps even hundreds of ground stations located within proximity of every country on Earth.

Let us now return to the issue of atmospheric attenuation. We have set out a feasible wavelength range of 3.75-30 cm as the suitable frequencies at which space-to-ground power transmission can realistically occur. This range can be further subdivided into four named microwave bands, from longest to shortest wavelength:

Band name	Frequencies	Associated wavelengths
L-band	1-2 GHz	15-30 cm
S-band	2-4 GHz	7.5-15 cm
C-band	4-8 GHz	3.75-7.5 cm
K-band	8-12 GHz	2.5-3.5 cm

> **Note**
> We set a cutoff at the K-band of 10 GHz (3 cm) as anything below would exceed 10% power loss due to atmospheric effects.

In theory, using the higher end of the feasible wavelength range (that is, 15-30 cm) would lead to the minimal losses - in fact, essentially no losses. However, this runs into the unfortunate issue that as the divergence angle $\theta \propto \lambda$, these longer wavelengths cause the beam to diverge faster and become less focused, ultimately negating the advantages of their low atmospheric loss. Rather, it is a better choice to use the lower end of the feasible wavelength range (3.75 cm-7.5 cm). In fact, we will use the lowest of the wavelength range - that is, $\lambda = 3.75\text{cm}$ - to minimize the beam spread.

Addenum: The ITU-R model The ITU-R atmospheric attenuation model, from the International Telecommunications Union (Radiocommunication Sector), forms the basis of the previous statement that 1-10 GHz (3.75-30 cm) microwaves can pass through the atmosphere with less than 10% loss. In the next subchapter, we will perform the relevant calculations using the ITU-R attenuation model to *justify* this statement. But we will take this as fact for the moment.

> **Note**
> For the remainder of this section, we will always give calculations in terms of the *range* of values

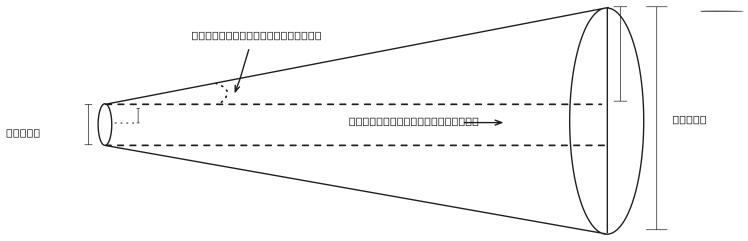
within our target microwave range of 1-12 GHz, or provide an average value (in which we will state explicitly that it is an average).

Determination of ideal aperture For Project Elara, determining this ideal aperture is *crucial* for designing a functional laser for power transmission from orbit. The main goal we want to attain is to minimize the beam size at the power receivers (in our case, the receiver stations) on Earth, which is a formidable task considering that the beam must traverse 37,000 km from geosynchronous orbit. We may derive an expression for the beam size at the power receivers through some algebra and trigonometry.

Let the angle of divergence $\theta = \dfrac{\lambda}{n\pi w_0}$ be the angle by which the beam diverges (spreads). Recalling that $A = 2w_0$, we may write this in equivalent form as:

$$\theta = \frac{2\lambda}{n\pi A} \tag{1101}$$

Over a straight-line distance D along the axis of propagation, the deviation (spread) of the beam from the axis of projection would then be given by $D\tan\theta$. As this is the spread from just one side of the beam, adding up the spread of both sides as well as the original beam width gives the total beam size at the receiver of $B = A + 2D\tan\theta$. Here is a graphical diagram of the derivation we have just completed:



The divergence of the Gaussian beam can be approximately-modelled as circular spot that grows in radius with distance.

Note
To be completely accurate, D should be written $D(t)$, as the distance between the power satellites and the receiver stations changes continuously as the satellites pass overhead. However, as D is unaffected by any of the variables we consider, we can treat it as effectively constant. In addition, the deviation of D from a straight line - Gaussian beams do not diverge at a constant rate - can be temporarily ignored due to the distances being so long that we may assume $\tan\theta \approx \theta$.

Given that we want a beam with minimal divergence - in fact, this would almost certainly be the most precise laser ever built - we can use the small-angle approximation $\tan\theta \approx \theta$ as θ *must* be very small. Therefore, we have:

$$B = A + 2D\tan\theta \tag{1102}$$
$$\approx A + 2D\theta \tag{1103}$$
$$= A + \frac{4D\lambda}{n\pi A} \Rightarrow \tag{1104}$$
$$B(A) = A + \frac{4D\lambda}{n\pi A} \tag{1105}$$

We want to find the value of A such that the beam size at the receiver is minimized. This is a calculus-based calculation - we need to find the points at which $\frac{dB}{dA} = 0$. Taking the derivative with respect to A, we have:

$$\frac{dB}{dA} = \frac{d}{dA}\left[A + \frac{4D\lambda}{n\pi A}\right] \tag{1106}$$

$$= 1 - \frac{4D\lambda}{n\pi A^2} \tag{1107}$$

$$\tag{1108}$$

And setting this equal to zero, we have:

$$1 - \frac{4D\lambda}{n\pi A^2} = 0 \tag{1109}$$

$$1 = \frac{4D\lambda}{n\pi A^2} \tag{1110}$$

$$A^2 = \frac{4D\lambda}{n\pi} \tag{1111}$$

$$A = \sqrt{\frac{4D\lambda}{n\pi}} \tag{1112}$$

We previously discussed that we intended to use a wavelength of $\lambda = 3.75$cm for optimal space-to-ground transmission, and here, D is the distance to geosynchronous orbit, which is roughly 37,000 km. Thus, when evaluated, this corresponds with an *ideal* aperture size of approximately 1.3 km. The corresponding beam size on the ground, calculated by $A + \frac{4D\lambda}{n\pi A}$, is close to 2.7 km (this is the *minimal beam size* within our range of microwave frequencies). If we were to use the slightly shorter (and thus slightly more lossy) wavelength of $\lambda = 3$cm, at the very edge of our wavelength range, the numbers come out to be 1.2 km for the aperture and 2.4 km for the beam size, respectively.

Now, apertures of greater than a kilometer are not as ridiculous as it would first seem, because through the technique of *beam combining*, smaller-aperture laser beams can be combined together into one laser beam of a much higher *effective* aperture. However, even with beam-combining, the substantial engineering hurdles of constructing such a laser make it non-ideal for the initial versions (though this is certainly not out of scope once we design later versions of the system).

Improved design considerations For a more realistic design, we would want a laser with a smaller aperture, with the trade-off of some increased beam divergence. This is not simply because the optimal aperture (calculated previously), even with beam-combining, is very difficult to achieve. There is also another major issue to consider. Recall that the beam size on the ground in the optimal case is 2.4-2.7 km. As we optimized *precisely* for the laser to stay as collimated as possible, this also means that the 1.2-1.3 km aperture laser would need to have a 2.4-2.7 km receiver. Again, this is *not* impossible, and in any case the eventual design will probably be on similar scales, if not the same design. However, in this design, the receiver cannot be made any smaller, as this is already the minimal spread aperture (even if that is hard to believe!) and the laser beam is *completely confined* to this region, meaning that the power density within the region will be extreme. That is to say, we optimized to make an extremely efficient laser that produces a highly-focused beam; but those are exactly the properties that make the laser extremely dangerous, meaning that we must cover the entire 2.4-2.7 km diameter area on the ground with a giant receiver that can safely receive the very powerful laser beam. Understandably, this is not ideal for the initial experimental stages of the solar power satellites.

Rather, it is more feasible to construct a laser design that is purpose-built for *safety*. Following official regulations on microwave safety, we can design a laser that would have a relatively spread-out beam that would ensure relatively low power densities on the ground. This means that a ground receiver need not be kilometers in size; it can cover just a small area, with the portions of the beam

outside of the receiver's area passing safely into the surrounding environment. The FDA requires that household microwave ovens have a maximum power density of $5mW/cm^2$ at 5 cm or less from the surface (21 CFR 1030.10), although this is far below a harmful dose. From some reading it appears that restaurants use up to twice that amount, at up to $10mW/cm^2$, which, again, is perfectly safe. If, out of an abundance of caution, we design for a power density limit of $I_0 = 1\text{mW/cm}^2$ - one-tenth of the restaurant amount - then we can calculate the aperture we would need to ensure the beam was sufficiently spread-out to have this power density on the ground.

Recall that the beam size on the ground, B, is the *diameter* of the beam by the time it reaches the Earth's surface. Thus, we can find the *cross-sectional area* of the beam, which we will denote A_{cs} (to avoid confusion with the aperture width A), by using the typical formula for the area of a circle:

$$A_{\text{cs}} = \pi \left(\frac{B}{2}\right)^2 \tag{1113}$$

Power density (intensity) is power over area, and thus we can relate the power density limit I_0 to A_{cs} and the total power of the laser P_T as follows:

$$I_0 = \frac{P_T}{A_{\text{cs}}} \tag{1114}$$

Assuming the laser is ideal, the total power of the laser would be equivalent to the total power of the captured sunlight (otherwise it is less than that). If the composite solar collector is composed of N hexagonal component mirrors each of side length s that combine to form one paraboloid., then the total power would be given by:

$$P_T = \frac{3\sqrt{3}}{2} s^2 N I_\odot \tag{1115}$$

> **Note**
> Hexagonal mirrors have the advantage that adding more naturally preserves the shape of the parabolic reflector, unlike circular mirrors, which means that the solar collector can be expanded essentially indefinitely as more mirrors are launched into orbit (at least in theory).

Where $I_\odot = 1361\text{W/m}^2$ is the mean solar irradiance (the average intensity of sunlight at Earth orbit). To solve for the aperture width A that would satisfy our power density limit, we can substitute in the expression for B into the expression for A_{cs}, resulting in a quadratic equation:

$$A_{\text{cs}} = \frac{P_T}{I_0} = \pi \left(\frac{B}{2}\right)^2 \tag{1116}$$

$$B = 2\sqrt{\frac{P_T}{I_0 \pi}} \tag{1117}$$

$$A + \frac{4D\lambda}{n\pi A} = 2\sqrt{\frac{P_T}{I_0 \pi}} \tag{1118}$$

$$A^2 + \frac{4D\lambda}{n\pi} = 2A\sqrt{\frac{P_T}{I_0 \pi}} \tag{1119}$$

$$A^2 - 2A\sqrt{\frac{P_T}{I_0 \pi}} + \frac{4D\lambda}{n\pi} = 0 \tag{1120}$$

$$\tag{1121}$$

We may solve the quadratic equation by applying the quadratic formula, and thus we have (after some simplifying):

$$A \leq A_0, A_0 = \sqrt{\frac{P_T}{I_0\pi}} - \sqrt{\frac{P_T}{I_0\pi} - \frac{4D\lambda}{n\pi}} \tag{1122}$$

Note
Here we use the smaller of the two roots because the larger root is nonsensical (it approaches impossibly large numbers for large P_T)

This, however, requires that $\frac{P_T}{I_0\pi} \geq \frac{4D\lambda}{n\pi}$ to get a sensible answer for A_0. Thus the *minimum* total power required for a real solution is given by:

$$P_T = \frac{4D\lambda I_0}{n} \tag{1123}$$

Which calculates to $P_T \approx 55.5$MW, approximately equivalent to the total power collected by a 115m radius *flat* (circular) mirror. This is also around equal to a composite mirror composed of 500 hexagonal segments each with 5.5m sides (although this is also an approximation because this also presumes a flat, not parabolic, shape). This means that for all $A \leq A_0$ the mean power density on the ground is within the safe limit (which, again, is a *very* conservative limit). The larger we make the composite mirror in space (by adding more hexagonal segments), the smaller the aperture can be to guarantee the same safe power density while also increasing the total power received on the ground, so it is advantageous to make the composite mirror as big as possible. The following table (calculation spreadsheet available at this link) showcases values for the number of hexagonal mirrors needed and aperture width required:

Effective aperture width A	s = 10m hexagonal mirrors required	s = 5.5m hexagonal mirrors required	s = 3m hexagonal mirrors required
1.329 km	157	519	1744
313.77 m	785	2594	8720
215.69 m	1570	5190	17440
174.55 m	2355	7784	26160
122.36 m	4709	15567	52320
94.46 m	7848	25944	87199
66.62 m	15696	51887	174398

These aperture values are *effective* aperture values (i.e. aperture after beam combining), so the *actual* apertures of the individual lasers do not have to be this large. As a general approximation, an effective aperture of $A_{\text{eff.}}$ can be achieved with n individual lasers of aperture A_{single} by $A_{\text{eff.}} = \sqrt{n}A_{\text{single}}$. By rearrangement, we can solve for n with:

$$n = \left(\frac{A_{\text{eff.}}}{A_{\text{single}}}\right)^2 \tag{1124}$$

The below table lists the number of individual lasers required to attain the effective aperture for individual lasers of different apertures:

Effective aperture width A	Number of 40m aperture lasers required	Number of 25m aperture lasers required	Number of 10m aperture lasers required	Number of 5m aperture lasers required	Number of 3m aperture lasers required
1.329 km	873	2826	17663	70650	196249
313.77 m	49	158	985	3939	10940
215.69 m	23	75	466	1861	5170
174.55 m	16	49	305	1219	3836
122.36 m	8	24	150	599	1664
94.46 m	5	15	90	357	992
66.62 m	3	8	45	178	494

Note
For comparison, the largest current laser is the **National Ignition Facility** (NIF) laser for inertial confinement fusion reseach, which uses 192 lasers each of 40 cm aperture.

The numbers, certainly, are formidable, but they are not out of reach. The hexagonal mirrors can be launched in batches, slowly, over the course of decades, with the solar power satellite providing more and more power over time, until it reaches full power. While the mirror numbers may seem staggering, remember that these mirrors are in space and therefore need no active support, and they can be made extremely thin and lightweight; their only requirement is to be reflective. Additionally, they can be folded to a very small size, then expand once they reach orbit, meaning that with some creative folding they can be made to take up relatively little room aboard a rocket. The much bigger engineering challenge, at least on paper, are the individual lasers, considering their very large apertures, even just individually.

A constructible prototype design It is important to note that so far, we have been discussing *only* designs on the very large-scale. However, our first space launch is likely to be a far, far smaller design - most likely a satellite on the order of a few thousand kilograms, deployed into orbit over three launches. Some rough reference parameters are listed below:

Power satellite mass	Power satellite orbit	Satellites in constellation	Mirror mass	Total payload size
~5,500 kg	Geosynchrnous	3	~500 kg	~6000 kg

The composite solar mirror would "ride along" with the power satellites and be launched alongside, in a folded compact configuration, until both reach orbit, at which point the mirror segments unfurl and lock together. Each segment would be hexagonal and very thin. Again, some reference parameters are listed below:

Composite mirror segments	Segment side length	Fully-unfurned diameter	Total captured power
5-55 segments	3-5m	20-40 m	428 kW - 1710 kW

Each power satellite is anticipated to have a large number of small lasers, using beam-combining to achieve a greater effective aperture. We must unfortunately use shorter wavelengths as the much smaller effective apertures as compared to the designs we previously examined mean that the lasers diverge *far* more quickly, even though we know that longer wavelengths are better at passing through the atmosphere without being affected by weather conditions. Below are some rough values for the projected power beam laser[29]:

[29]The received power values are assuming a 10-meter (diameter) ideal receiver antenna. In addition, the ground power density values are *not accounting for atmospheric attenuation*, which, at such short wavelengths, can mean losses up to 50% or more in heavy rain or stormy weather, even if losses are within 10% during clear skies.

Feed laser aperture	Wavelength	Number of beam-combined feed lasers	Effective aperture	Ground beam width	Ground power density	Received power
At least 25 cm	1-2 cm	At least 200	>350 cm	133-266 km	$7.67\mu W/m^2$ to $122.62\mu W/m^2$	0.6 - 9.6 mW

On Earth, the receiver would also be rather small compared with our previous designs - we envision a 10-meter (diameter) receiver parabolic antenna, which is used to calculate the received power in the last column. In addition, the receiver antenna will not be a custom one; it will likely involve a temporary (or permanent) lease of a pre-existing telecommunications antenna. Of course, the larger the receiver, the better; however, the received power is likely to be rather weak, about 8 times as powerful as a household AA battery in theory, but perhaps only half as powerful in reality. Even in the best possible case (such as, for instance, if we hypothetically used NASA's 70m Deep Space Network antenna as the receiver), the received power would still be under half a watt, or at most, a few watts.

So it should be emphasized that this is a *research testbed* to test out the system design in a real-world environment, which must be undertaken before building any large-scale solar power satellites for practical energy generation. This means that we don't expect this testbed to actually produce any useful amount of energy[30], and while the mirrors will be re-used (recall that the hexagonal mirrors can be continuously extended with more segments), and the receiver antenna returned to its operator, the power satellite will be placed in a graveyard orbit after its useful lifetime. In some sense, it would be like a Chicago Pile for Project Elara's solar power satellites - essentially, a proof-of-concept.

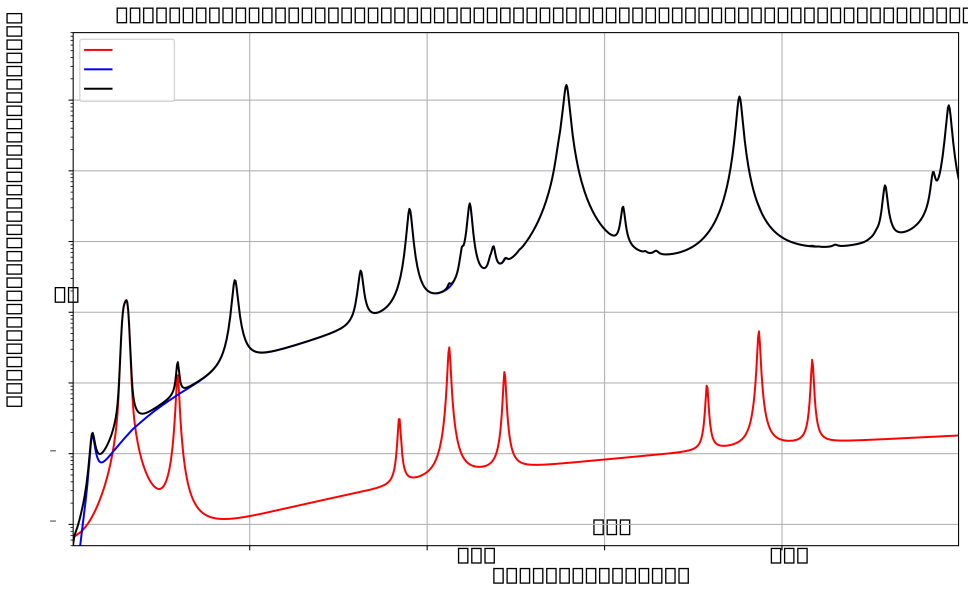
[30]Do note, however, that commercial satellites have even weaker signals, so weak that even the milliwatt power levels in the projected testbed are *orders of magnitude* more powerful than nearly all satellites in similar orbits.

Atmospheric attenuation

In the previous section, we stated our desired frequency range without proof. In this section, we will prove it. To do so, we calculate the atmospheric attenuation (that is, the effect of the atmosphere on space-to-ground power transmission) specific cases as well as the general case. We will be using these values to determine a suitable microwave range for the Elara solar power satellites.

```
% pip install itur
import itur
import astropy.units as u
from astropy.table import Table
import numpy as np
import matplotlib.pyplot as plt

# Vector graphics
%config InlineBackend.figure_format = 'svg'
# Serif text
plt.rcParams["font.family"] = "serif"
plt.rcParams['mathtext.fontset'] = 'stix'
```

We use the Recommendation ITU-R P.618 model created by the International Telecommunications Union, as implemented by the ITU-Rpy library, to perform our calculations. To give a sense of the model, the following is a graph of atmospheric attenuation for an older version of the model (P.676):



Specific attentuation (attenuation per unit distance) across microwave frequencies from 1-1000 GHz.

Gaseous attenuation The first case of attenuation is the attenuation due to atmospheric gases in the atmosphere. We will do the calculations here, using the Recommendation ITU-R P.618 model (which we will henceforth refer to as *"the ITU-R reference model"* or as simply *"the ITU-R model"*), as we discussed.

```
# reference conditions
rho = 7.5 * u.g / u.m**3 # water vapour density
P = 1013 * u.hPa # atmospheric pressure
T = 25 * u.deg_C # temperature
```

The ITU-R model can be chosen to yield an approximate or exact solution, depending on the level of accuracy required. We will use the approximate mode for now, though once the precision is necessary, we will need to use the exact mode.

```
MODE = "approx"
```

In general, the attenuation is a function of both atmospheric conditions (water vapour, density, rain, dust, and cloud conditions) as well as the angle of elevation of the power satellites relative to a receiver station, which will change continuously as the power satellites orbit. We can see the effect of elevation angle on attenuation quite distinctly. For instance, with otherwise identical parameters, an elevation angle of 90 degrees (i.e. satellite directly above the receiver station) has *drastically* lower attenuation as compared to an elevation angle of 5 degrees (i.e. satellite at the edge of the horizon relative to the receiver station:

```
Att5 = itur.gaseous_attenuation_slant_path(1, 5, rho, P, T) # for elevation angle of 5 deg
Att5
```

```
Att90 = itur.gaseous_attenuation_slant_path(1, 90, rho, P, T) # for elevation angle of 5 deg
Att90
```

```
print("For 90 deg: " + str(round((1 - 10**( -(Att90/u.dB).value/10)) * 100, 2)) + "%")
```

```
print("For 5 deg: " + str(round((1 - 10**( -(Att5/u.dB).value/10)) * 100, 2)) + "%")
```

We will enter in the conditions that result in maximum attenuation (power loss due to the atmosphere) as our **baseline value** (i.e. with the minimum elevation angle supported by the ITU-R model, which is 5 degrees). This means that this is an estimate of the worst-case power losses, which is important to design for to ensure that the system is robust.

```
el = 5 * u.deg # lowest elevation angle (nearly completely horizontal = longest distance = max
f = np.linspace(1, 100, 1000) * u.GHz
```

```
Att = itur.gaseous_attenuation_slant_path(f, el, rho, P, T, mode=MODE)
```

```
plt.semilogy(f, Att)
plt.xlabel('Frequency [GHz]')
plt.ylabel('Gaseous Attenuation [dB]')
plt.grid(which='both', linestyle=':')
```

Note that these attenuation values are in *decibels*, which are a logarithmic unit:

```
Att.unit
```

To extract the raw values we make it dimensionless by dividing by `u.dB` to cancel out the units, then apply the typical log formula $\ell = 10^{-\mathrm{dB}/10}$ where ℓ is the percent loss in linear (instead of logarithmic) units.

```
Att_dimensionless = Att/u.dB
```

```
loss_linear = 1 - 10**( -Att_dimensionless/10)
```

We now pick the frequencies whose attenuation values that corresponds to <10% loss (which is >90% transmittance):

```
percent = u.def_unit("%", u.dimensionless_unscaled) #define new percent unit
```

```
# pick out the frequencies corresponding to
# attenuations with less than 10% loss
f_range = f[loss_linear <= 0.1]
# and pick out the respective attenuations
loss_percents = loss_linear[loss_linear <= 0.1] * 100 * percent

c = 299792458 * u.m / u.s

c

wavelengths = (c/f_range).to("cm")

freq_table = Table([
    np.round(f_range, 2),
    np.round(wavelengths, 2),
    np.round(loss_percents, 3)],
    names=["Frequency", "Wavelength", "Gaseous attenuation losses (worst -case)"],
    descriptions=None, dtype=None, meta=None)

freq_table
```

Based on the gaseous attenuation, the suitable frequencies are between 1 and 6.75 GHz, corresponding approximately to 4.4 - 30 cm. We will now examine the other sources of attenuation. This includes the contribution due to rain, clouds, and scintillation (rapid changes of refractive index as electromagnetic waves pass through the atmosphere due to turbulence and other effects). While the specific modelling of each is rather complex, the full model is thankfully implemented in code so we can compute it readily.

For this, we will use the `atmospheric_attenuation_slant_path` method, which also takes into account the Earthbound receiver diameter and the latitude and longitude of the receiver. We will use two test locations for calculating the attenuation. The first is the Bering sea, located at 54.393203ř N, 172.369927ř W, representing (*in theory*) close to the worst-case attenuation due to its remoteness and high lattitude (which both increase atmospheric losses). The second is Hawaii, located at 19.8987ř N, 155.6659ř W, representing (*in theory*) close to the most favorable conditions and therefore the best-case attenuation. (See this GPS coordinates viewer for an interactive way to view these coordinates). From this we can calculate a range of values. Note that *we are not planning to put a power receiver station at either location.* Their general location on the planet is simply a reference point for planning.

```
# Calculate other sources of attenuation (dust, rain, clouds) and also the total attenuation
# args: lat, lon, f, el, p, D
# D: earth -station antenna diameter (m)
# we will use 10m as a conservative estimate
# as attenuation is highest for small -diameter receivers
D = 10

# for lat/long. we will calculate using two values:
# 1) that of approx. the Bering sea (because further north there are basically no human settle
# 2) close to the equator (Hawaii)
# Note that we must set the longitude with a negative sign as
# itur expects a longitude value with respect to the east (not the west)
lat1, long1 = (54.393203, -172.369927)
lat2, long2 = (19.8987, -155.6659)

# Just like before we use 5 degree elevation anlge as it is the
# lowest elevation = highest attenuation
# and we design for the highest -attenuation conditions as our baseline value
```

```
# This parameter controls the percentage of time the rain
# attenuation is exceeded - we want our system to work even
# in the most extreme storm conditions
R_percent = 0.5

# Bering sea location
Att_location1 = itur.atmospheric_attenuation_slant_path(lat1, long1, f, el, R_percent, D, mode
# Hawaii location
Att_location2 = itur.atmospheric_attenuation_slant_path(lat2, long2, f, el, R_percent, D, mode

plt.semilogy(f, Att_location1, label="Bering Sea location")
plt.semilogy(f, Att_location2, label="Hawaii location")
plt.xlabel('Frequency [GHz]')
plt.ylabel('Total Attenuation [dB]')
plt.grid(which='both', linestyle=':')
plt.legend()
plt.show()
```

Note

It does appear that the tropical ocean may contribute to the attenuation in a way that was not entirely expected, making it so that the Hawaii test location has a higher attenuation than the Bering Sea test location at most frequencies.

We will now do the same analysis as previously, just for both the test locations:

```
def attenuation_losses(att):
    # same formula as before in modified form
    loss_linear = 1 - 10**( -Att/(10 * u.dB))
    # pick out the frequencies corresponding to
    # attenuations with less than 10% loss
    f_range = f[loss_linear <= 0.1]
    # convert to wavelengths
    wavelengths = (c/f_range).to("cm")
    # and pick out the respective attenuations
    loss_percents = loss_linear[loss_linear <= 0.1] * 100 * percent
    return f_range, wavelengths, loss_percents

def generate_att_table(att):
    f_range, wavelengths, loss_percents = attenuation_losses(att)
    return Table([
        np.round(f_range, 2),
        np.round(wavelengths, 2),
        np.round(loss_percents, 3)],
    names=["Frequency", "Wavelength", "Gaseous attenuation losses (worst -case)"],
    descriptions=None, dtype=None, meta=None)

Att_table_1 = generate_att_table(Att_location1)
Att_table_2 = generate_att_table(Att_location2)

Att_table_1

Att_table_2
```

Numerical Simulations, Part 1

In the previous chapter, we discussed the ideas that led up to choosing a particular frequency for power transmission. In addition, in our section on antenna theory, we discussed the *general ideas* behind finite element analysis, and derived a weak form for the wave equation and its closely-related time-independent cousin, the **Helmholtz equation**. However, we did not discuss how to actually translate these ideas into code so that they can be solved by finite-element software. This is what we will cover in this section.

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import norm
from findiff import FinDiff
from findiff.diff import Coef, Id
from findiff.pde import BoundaryConditions, PDE
import scipy.sparse as sparse
import scipy.optimize as optimize
from numpy.linalg import norm
from random import randint

%matplotlib inline
# settings for professional output
plt.rcParams["font.family"] = "serif"
plt.rcParams["axes.grid"] = True
plt.rcParams["mathtext.fontset"] = "stix" # use STIX fonts (install from https://www.stixfonts
# optional - always output SVG (crisper graphics but often slow)
# %config InlineBackend.figure_format = 'svg'
```

Finite element analysis Our first numerical simulation will be to understand how microwaves reflect off a parabolic reflector, which describes both solar mirrors and the receiving antennas we plan to receive energy from space. This simulation involves using the FreeFEM++ software (we'll just call it "FreeFEM" for short) to be able to run a finite-element simulation of the Helmholtz equation.

We'll start from where we left off earlier. Recall that the weak form of the Helmholtz equation we found previously (in our section on antenna analysis) was given by:

$$-\int_\Omega \nabla_J \mathbf{E} : \nabla_J \Phi \, dV + k^2 \int_\Omega \Phi \cdot \mathbf{E} \, dV + \int_{\partial\Omega} \Phi \cdot \nabla_J \mathbf{E} \, d\mathbf{A} = 0 \tag{1125}$$

Translating this into FreeFEM's custom finite-element simulation language, this weak form becomes:

```
// jacobian macro for vector function F = [Fx, Fy]
macro J(F) [grad(F#x), grad(F#y)] //

// ... previous code defining ‘Efield‘ and ‘TestPhi‘ functions ...

problem ElectricField(Efield, TestPhi)
    = int2d(Th)(
        k^2 * TestPhi * Efield
    )
    - int2d(Th)(
        grad(Efield)' * grad(TestPhi)
    )
    + on(SOME_FARAWAY_BOUNDARY, Efield=SOME_SMALL_NUMBER);

// ... rest of code ...
```

However, just having the weak form isn't enough! We also need the *boundary conditions*. This comes in several parts. First, since we can't simulate an infinite domain, we'll have to place our parabolic reflector in a mathematical box of side lengths L. Second, we'll have to *parametrize* the parabolic reflector to model it mathematically. These additional complications can make us wonder if our solver is correct at all! So rather, let us start with a much *simpler* example: solving the Helmholtz equation on a unit square with some basic boundary conditions:

$$E(x,0) = 3, \quad E(x,1) = 10$$
$$E(0,y) = 6, \quad E(1,y) = 1$$

The weak form then simplifies to:

$$\int_\Omega \nabla E \cdot \nabla \Phi \, dV - k^2 \int_\Omega \Phi E \, dV - \int_{\partial\Omega} \Phi \cdot \nabla E \, d\mathbf{A} = 0 \tag{1126}$$

Where we multiplied both sides by -1 for getting rid of the -1 on the first term of the weak form. We can now start writing the finite-element code for FreeFEM. First, we define a few constants to label our boundaries and specify the number of samples we want:

```
int top = 1;
int right = 2;
int bottom = 3;
int left = 4;
int squaren = 30;
real k = 3.0;
```

Then, we define our boundaries for our unit square via parametric curves:

```
border boxtop(t=1, 0){ x=t; y=1; label=top; } // top
border boxright(t=0, 1){ x=1; y=t; label=right; } // right
border boxbottom(t=0, 1){ x=t; y=0; label=bottom; } // bottom
border boxleft(t=1, 0){ x=0; y=t; label=left; } // left

mesh Th = buildmesh(boxtop(squaren) + boxright(squaren) + boxbottom(squaren) + boxleft(squaren
plot(Th, value=1, wait=1);
```

Next, we discretize our geometry to add our test function Φ and our electric field magnitude $E(x,y)$ as variables to be solved for:

```
fespace Vh(Th, P1);
Vh TestPhi;
Vh Emag;
```

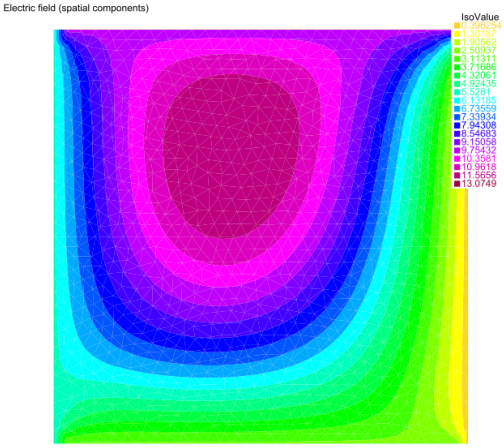
We can then input the weak form of the Helmholtz equation:

```
// Note: Helmholz equation = all bilinear terms
problem ElectricField(Emag, TestPhi)
    = int2d(Th)(
        k^2 * Emag * TestPhi
    )
    - int2d(Th)(
        grad(Emag)' * grad(TestPhi)
    )
    // boundary conditions
    + on(bottom, Emag=3)
    + on(top, Emag=10)
    + on(left, Emag=6)
    + on(right, Emag=1);
```

We're now ready to solve! We can then show a plot of the solution, as follows:

```
cout << "Solving PDE (validation test)..." << endl;
ElectricField;
plot(Emag, value=true, fill=true, cmm="Electric field (spatial components)", wait=1, eps="plot
```

We show a visualization of this solution below:



Scalar-valued validation test Up to this point, our results *look* correct. But do we know if they are? This is a tricky answer because most numerical problems - including the one we are analyzing - do not have analytical solutions. However, we can *validate* our finite-element results by comparing against another numerical solver. Specifically, we will compare our results to a solve using a different numerical technique - the **finite difference method (FDM)**.

To simplify our analysis, we begin by working with the scalar variant of the Helmholtz equation, which is given by:

$$(\nabla^2 + k^2)E = 0 \tag{1127}$$

This is the exact same as the finite element formulation which we will keep for consistency. We wish to solve it via the finite difference method with a discrete Laplacian. To do this, we first set up our grid:

```
k = 3.0
# this has to be set to a low number or else the generated mesh
# might take up too much memory and crashing Python
squaren = 100
shape = (squaren, squaren)

x = np.linspace(0, 1, squaren)
y = np.linspace(0, 1, squaren)
X, Y = np.meshgrid(x, y, indexing='ij')

dx = 1 / squaren
dy = 1 / squaren

bc = BoundaryConditions(shape)
bc
```

```
<findiff.pde.BoundaryConditions at 0x18259683e60>
```

In our domain, we set a simple problem to solve, using the Dirichlet boundary conditions $E|_{\partial\Omega} =$ const.:

```
bc[:, 0] = 3 # E(x, 0)
bc[:, -1] = 10 # E(x, 1)
bc[0, :] = 6 # E(0, y)
bc[ -1, :] = 1 # E(1, y)
```

We have two FDM solvers - a custom solver `solve_Helmholtz_equation()` and a wrapper for FinDiff's own solver `findiff_solve_Helmholtz_equation()`, which are below.

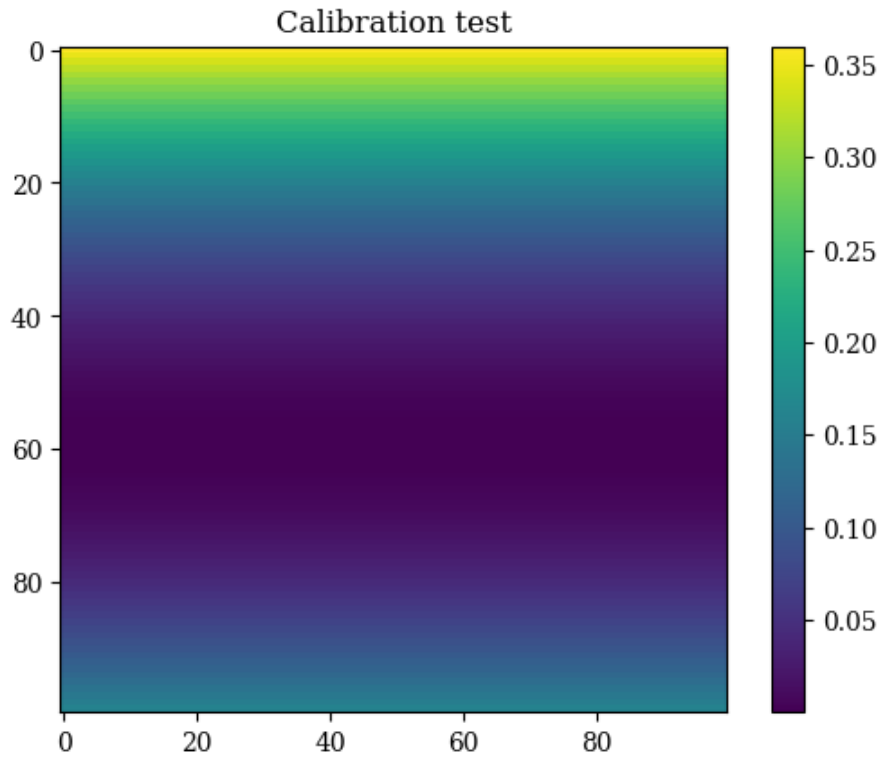
For Project Elara researchers
Previously in testing there was a findiff "issue" with the construction of the operator $\nabla^2 + k^2$. It would be preferable to just convert the stuff to a matrix to solve because it is still unknown whether `Coef(k**2)*Id()` is the correct way to construct it. But seems like now that it yielded the correct solution all along and the solution was just *thought* to be wrong.

```
def findiff_solve(bc=bc, grid_shape=shape, k=k, dx=dx, dy=dy):
    Helmholtz = FinDiff(0, dx, 2) + FinDiff(1, dy, 2) + Coef(k**2)*Id()
    rhs = np.zeros(grid_shape)
    pde = PDE(Helmholtz, rhs, bc)
    return pde.solve()

def create_Helmholtz_operator(dx, dy, grid_shape, k):
    n, m = grid_shape
    laplacian = FinDiff(0, dx, 2) + FinDiff(1, dy, 2)
    ksquared = k**2 * sparse.eye(np.prod(grid_shape))
    # reshape() automatically selects to whichever shape necessary
    Helmholtz_mat = laplacian.matrix(grid_shape) + ksquared
    return Helmholtz_mat

def solve_Helmholtz_equation(dx=dx, dy=dy, bc=bc, grid_shape=shape, k=k):
    Helmholtz = create_Helmholtz_operator(dx, dy, grid_shape, k)
    rhs = np.zeros(shape)
    f = rhs.reshape( -1, 1)
    # set boundary conditions
    # this code is copied over from
    # findiff's source code in findiff.pde.PDE.solve()
    nz = list(bc.row_inds())
    Helmholtz[nz, :] = bc.lhs[nz, :]
    f[nz] = np.array(bc.rhs[nz].toarray()).reshape( -1, 1)
    print("Solving (this may take some time)...")
    solution = sparse.linalg.spsolve(Helmholtz, f).reshape(grid_shape)
    print("Solving complete.")
    return solution

def plot_E(E, X=X, Y=Y, label="Surface plot of solution data", rot=30):
    fig = plt.figure()
    ax = fig.add_subplot(projection='3d')
    ax.set_xlabel("$x$")
    ax.set_ylabel("$y$")
    ax.set_zlim(np.min(E), np.max(E))
    ax.view_init(30, rot)
    surf = ax.plot_surface(X, Y, E, cmap="coolwarm")
    fig.colorbar(surf, shrink=0.6)
    if not label:
```

```
        plt.title("Surface plot of solution data (v2)")
    else:
        plt.title(label)
    plt.show()

E = solve_Helmholtz_equation()
E_findiff = findiff_solve()

Solving (this may take some time)...
Solving complete.
```

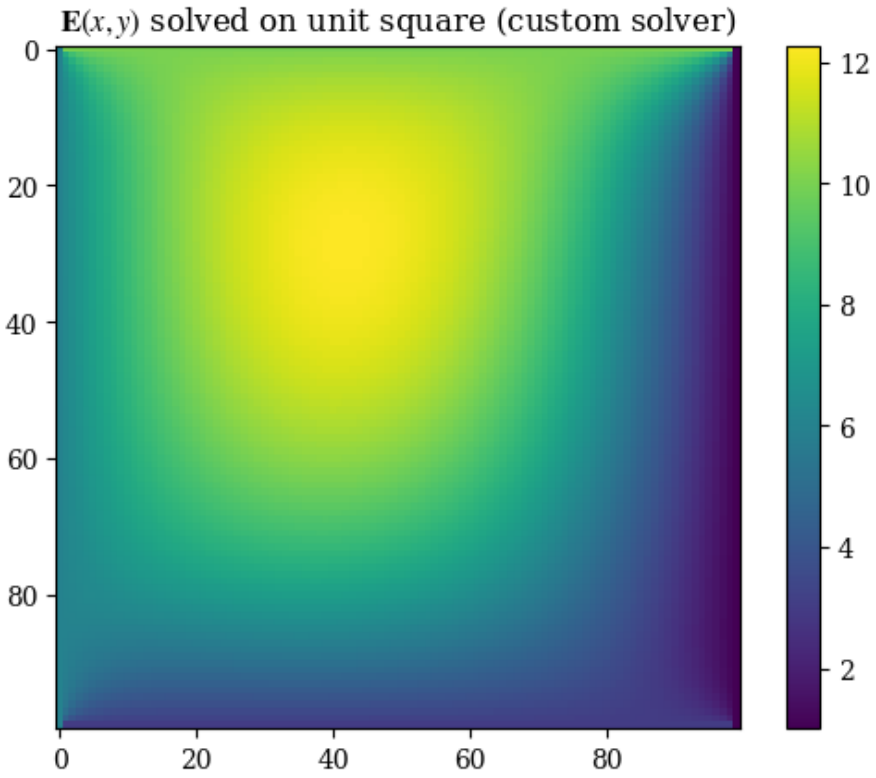
However, the solution is different in form to the typical mathematical (Cartesian) representation. This is because arrays are stored in (row, column) order, that is, (y, x), as is standard for computers, and in addition to this their origin is located at the top-left, rather than the bottom-left as is used in Cartesian coordinates. So we must convert to the standard mathematical representation before displaying. This consists of transposing, then flipping the array along the columns axis.

```
def correct_axes(mat2d):
    return np.flip(mat2d.T, axis=0)

def calibrate(x=X, y=Y):
    f = (y - 0.4)**2 # the asymmetrical test function for calibration
    plt.imshow(correct_axes(f), interpolation="none")
    plt.title(r"Calibration test")
    plt.grid(False)
    plt.colorbar()
    plt.show()

calibrate()
```

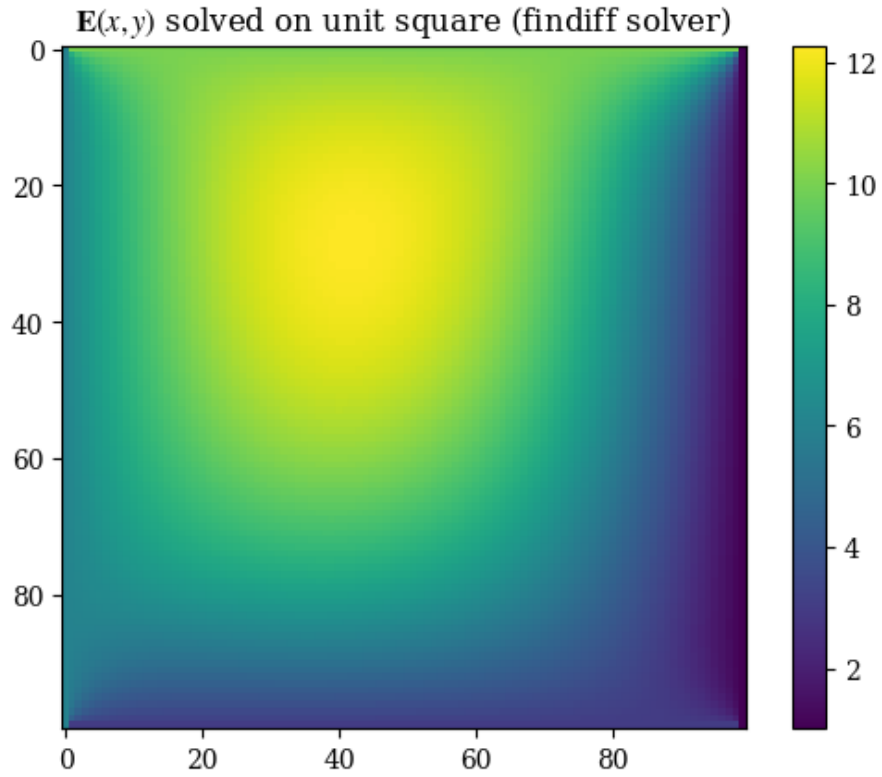


Performing the proper transformation to yield the correct representation for images, we can see the results below:

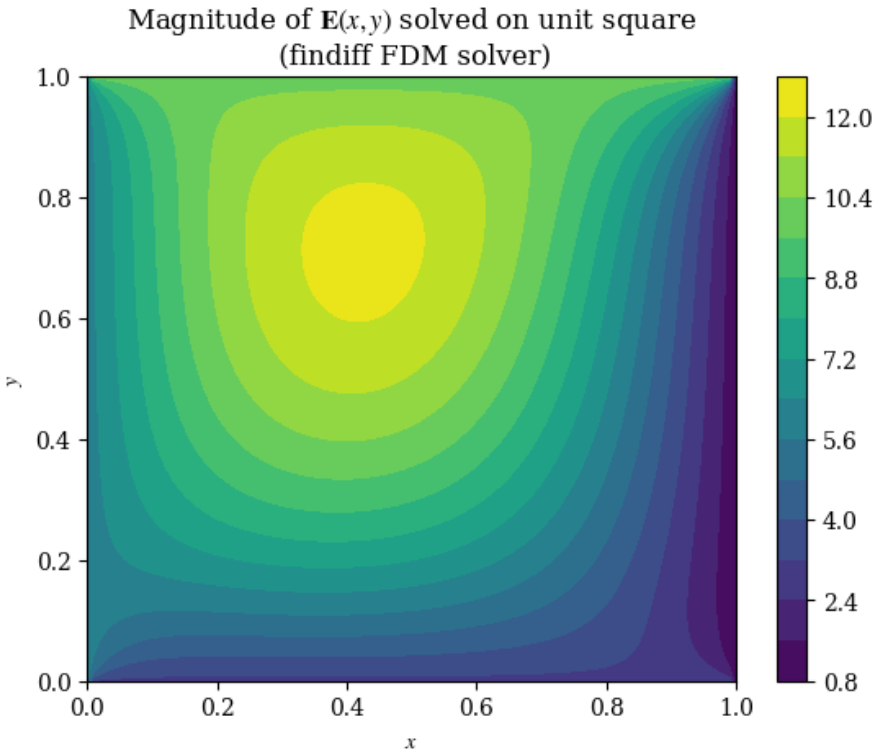
```
plt.imshow(correct_axes(E), interpolation="none")
plt.title(r"$\mathbf{E}(x, y)$ solved on unit square (custom solver)")
plt.colorbar()
plt.grid(False)
plt.show()
```



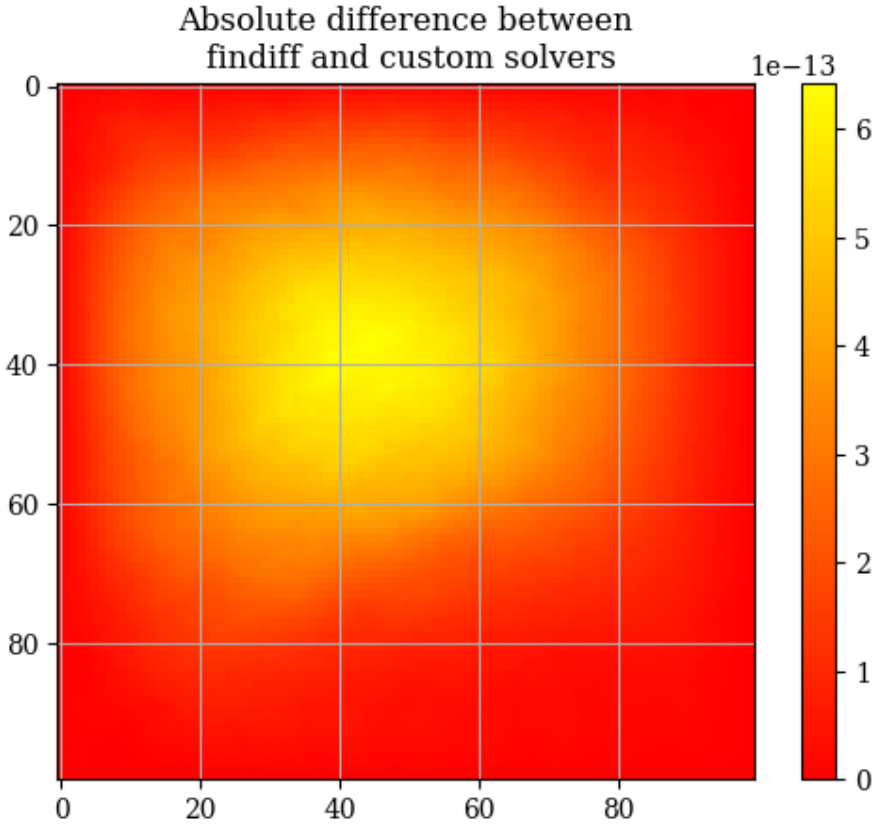

```
plt.imshow(correct_axes(E_findiff), interpolation="none")
plt.title(r"$\mathbf{E}(x, y)$ solved on unit square (findiff solver)")
plt.colorbar()
plt.grid(False)
plt.show()
```




```
imgX, imgY = np.meshgrid(np.arange(squaren), np.arange(squaren))

#plt.imshow(correct_axes(E_findiff), interpolation="none")
plt.contourf(X, Y, E_findiff, levels=15)
plt.title(r"Magnitude of $\mathbf{E}(x, y)$ solved on unit square" + "\n (findiff FDM solver)"
plt.xlabel(r"$x$")
plt.ylabel(r"$y$")
plt.colorbar()
plt.grid(False)
plt.show()
#plt.savefig("fdm -validation.eps", dpi=300)
```




```
abs_difference = np.abs(E - E_findiff)

plt.imshow(correct_axes(abs_difference), interpolation="none", cmap="autumn")
plt.title("Absolute difference between\n findiff and custom solvers")
plt.colorbar()
plt.show()
```



```
abs_difference.max()
```

```
np.float64(6.430411758628907e -13)
```

The general differences between the findiff solver and the custom solver are very very miniscule on the order of $\mathcal{E} \sim 10 \times e^{-13}$. Therefore we will use the two solutions interchangeably and consider them equivalent. If we inspect the solution, we can clearly see that the boundaries follow the boundary conditions:

```
E[0, :] # E(0, y) = 6
```

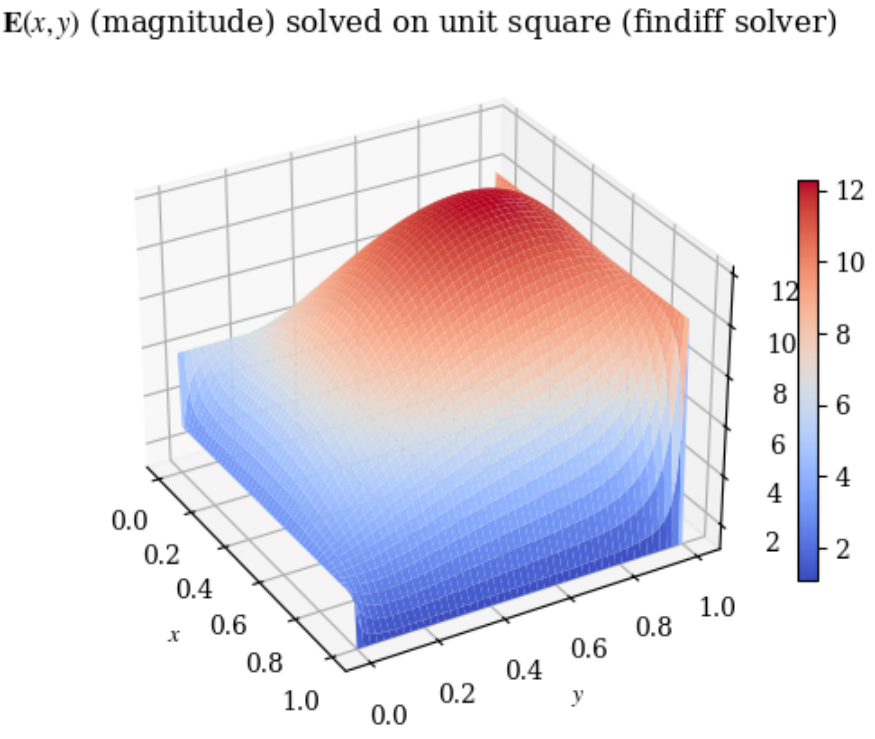
```
array([6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6.,
       6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6.,
       6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6.,
       6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6.,
       6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6.,
       6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6., 6.])
```

```
E[ -1,:] # E(1, y) = 1
```

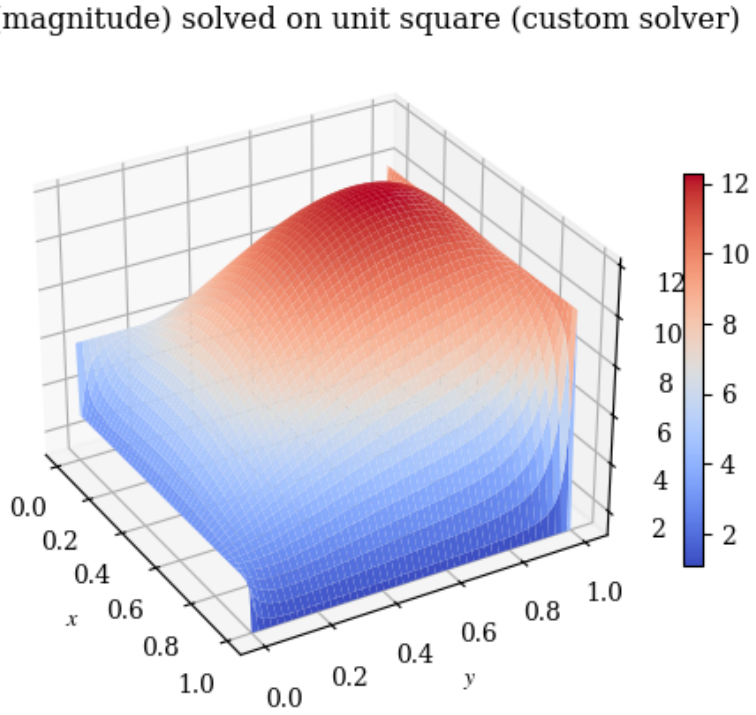
```
array([1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.,
       1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1., 1.])
```

We can also verify with a 3D plot (where we now don't need to do the transpose trick because we're displaying 3D points and matplotlib is smart enough to figure that out):

```
plot_E(E_findiff, label=r"$\mathbf{E}(x, y)$ (magnitude) solved on unit square (findiff solver
```




```
plot_E(E, label="E(x, y) (magnitude) solved on unit square (custom solver)", rot= -30)
```



The second test is the vector-valued version that uses the same coordinate transformations as the full version. However, do note that the Helmholtz equation's vector components are **independent** of each other, meaning that the components can just as well be simulated separately (as two scalar PDEs) as together (as one vector PDE).

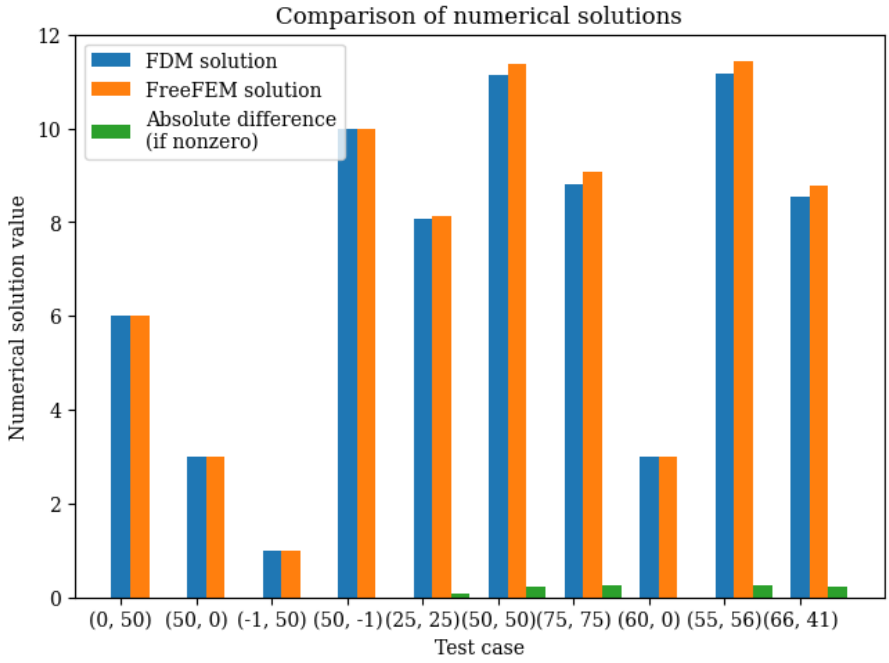
```
def gen():
    step = randint(0, squaren)
    step2 = randint(0, squaren)
    print(f"Generated test point ({step*dx:.3f}, {step2*dx:.3f}) corresponding to index [{step

gen()

Generated test point (0.980, 0.420) corresponding to index [98, 42]
```

We test the validation points on the domain to compare against the FreeFEM version - note that this is because the finite element output is not on a regular mesh so cannot be compared value-by-value against the finite-difference result:

```
# convert real -space (x, y)
# coordinates to index (m, n)
# of the solution array
def convert_coord_index(x, y):
    x_idx = round(x*squaren)
    y_idx = round(y*squaren)
    # prevent out of bound
    # access errors
    if x_idx == squaren:
        x_idx = -1
    if y_idx == squaren:
        y_idx = -1
    return (x_idx, y_idx)

# unlike the freefem version, for this one
# we use the *index* locations of each point
```

```
# e.g. (1.0, 1.0) is equal to index [ -1, -1]
# because it is the endpoint on both x and y
#
# for the ones where this method is a bit wonky, the
# convert_coord_index function is used
# which rounds the evaluation location to
# the closest set of [idx, idx] values
cases = [
    # boundary points
    (0, 0.5),
    (0.5, 0),
    (1., 0.5),
    (0.5, 1),
    # center points
    (0.25, 0.25),
    (0.5, 0.5),
    (0.75, 0.75),
    # the three nonstandard points
    # given by the gen() function above
    (0.6, 0.),
    (0.55, 0.562),
    (0.663, 0.413)
]

testpoints = [convert_coord_index(*c) for c in cases]

def validate():
    results = [0 for i in range(len(testpoints))]
    for p_idx, p in enumerate(testpoints):
        i = p[0]
        j = p[1]
        # uncomment to show the corresponding index
        # print("Index:", f"({i}, {j})")
        if i == 0:
            x = 0
        else:
            x = i/squaren if i!= -1 else 1
        if j == 0:
            y = 0
        else:
            y = j/squaren if j!= -1 else 1
        res = E[i][j]
        print(f"On test point ({x:.3f}, {y:.3f}), result value {res}")
        results[p_idx] = res
    return results

res = validate()

On test point (0.000, 0.500), result value 6.0
On test point (0.500, 0.000), result value 3.0
On test point (1.000, 0.500), result value 1.0
On test point (0.500, 1.000), result value 10.0
On test point (0.250, 0.250), result value 8.06733660036659
On test point (0.500, 0.500), result value 11.143921337204521
```

```
On test point (0.750, 0.750), result value 8.816212248786162
On test point (0.600, 0.000), result value 3.0
On test point (0.550, 0.560), result value 11.175781231851033
On test point (0.660, 0.410), result value 8.548252807003232

# from wave -parabolic -6 -validation_4.edp
fem_results = [
    6.0,
    3.0,
    1.0,
    10.0,
    8.14275,
    11.3661,
    9.07413,
    3.0,
    11.4339,
    8.78092
]

res_array = np.array(res)
fem_res_array = np.array(fem_results)

bar_x = np.arange(10)

points_labels = [str(p) for p in testpoints]

def plot_barchart(width=0.25, save=False):
    fig, ax = plt.subplots(layout='constrained')
    ax.set_title("Comparison of numerical solutions")
    ax.set_xlabel("Test case")
    ax.set_ylabel("Numerical solution value")
    p1 = ax.bar(bar_x, res_array, width=width, label="FDM solution")
    #ax.bar_label(p1, label_type="edge", fmt=lambda x: '{x:.2f}')
    p2 = ax.bar(bar_x + 0.25, fem_res_array, width=width, label="FreeFEM solution")
    #ax.bar_label(p2, label_type="edge", fmt=lambda x: '{x:.2f}')
    p3 = ax.bar(bar_x + 0.5, np.abs(fem_res_array - res_array), width=width, label="Absolute d
    #ax.bar_label(p2, label_type="edge", fmt=lambda x: '{x:.2f}')
    ax.set_xticks(bar_x, points_labels)
    ax.legend()
    plt.grid(False)
    if save:
        plt.savefig("validation -bar -chart.eps", dpi=600)
    else:
        plt.show()

plot_barchart()
```



Vector-valued validation test We also want to do a simulation with the same boundary conditions but using a transformed coordinates system and vector-valued. As findiff can only solve scalar PDEs, in practice this means that we are solving 2 separate PDEs with different boundary conditions, and then combining them to get a vector-valued function. We are still using the same value of k and the same domain (of a unit square) as before. For the vector PDE, the coordinates in (x, y) space are:

$$E_x(x,0) = 2, E_x(x,1) = 7, E_x(0,y) = 0.5, E_x(1,y) = \pi \tag{1128}$$
$$E_y(x,0) = 2\pi, E_y(x,1) = 3, E_y(0,y) = 12, E_y(1,y) = 1 \tag{1129}$$

What we want is to convert to (u, v) coordinates where $u = e^x$ and $v = e^y$. To preserve the same physical results, we must ensure that $E(u, v) = E(x, y)$. which means re-expressing each of those boundary conditions in terms of functions $E_u(u, v)$ and $E_v(u, v)$. To do this we use the forward transforms $E(u, v) = E(e^x, e^y)$. This means the function stays identical, and is simply expressed in different coordinates.

As an example, consider the first boundary condition $E_x(x, 0) = 2$. The constant-valued outputs of the functions remains the same under the coordinate transformation, and the only thing that changes are the coordinates. That is to say, $E_u(u, v) = E_x(u(x), v(y)$. For this, we substitute $u(x)$ and $v(y)$ into $E_x(x, 0)$ as follows:

$$E_u(u,v) = E_x(u(x), v(y))\big|_{(x,y)=(x,0)} = E_x(u(x), v(0)) = E_x(e^x, 1) = E_u(u, 1) \tag{1130}$$

Doing so for each of the expressions, we obtain:

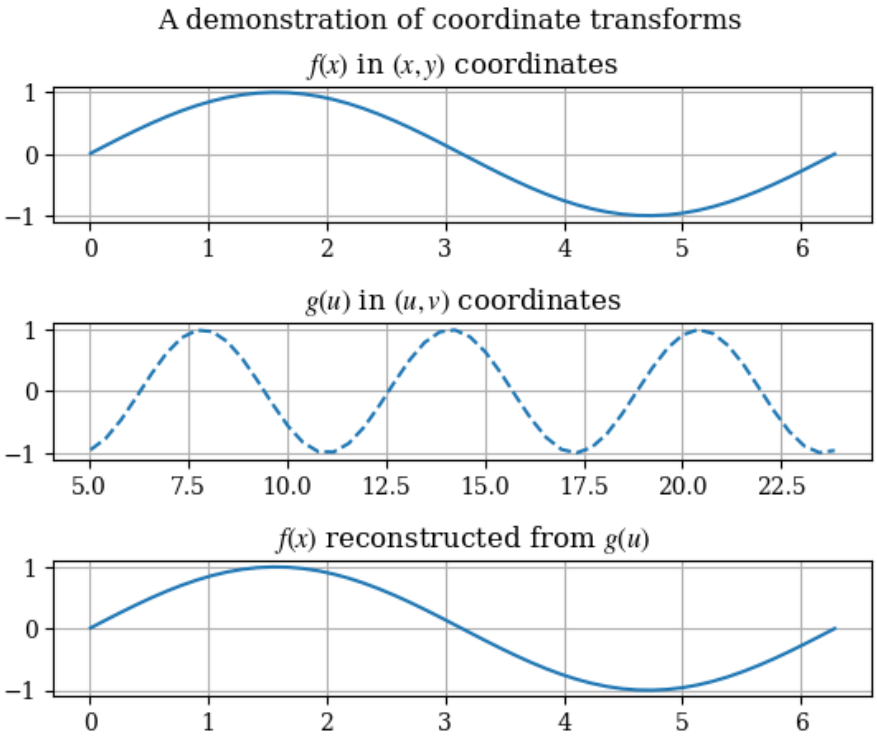
$$E_u(u,1) = 2, E_u(u,e) = 7, E_u(1,v) = 0.5, E_u(e,v) = \pi E_v(u,1) = 2\pi, E_v(u,e) = 3, E_v(1,v) = 12, E_v(e,v) = 1 \tag{1131}$$

Where the domain $(x, y) \in [0, 1] \times [0, 1]$ becomes rescaled to $(u, v) \in [1, e] \times [1, e]$, as can be seen below:

```
u_example = np.exp(x)
print(f"New range: {u_example[0]:.3f}...{u_example[ -1]:.3f}")

New range: 1.000...2.718
```

As a demonstration of the equivalence of solutions after a coordinate transform, consider the following change of variables $x \to u = 3x + 5$ on a given function:

```
def test_coord_transform():
    # not from zero because we don't want 1/0
    x = np.linspace(0.01, 2*np.pi, 50)
    # u = u(x) the forwards transform
    u_of_x = lambda x: 3*x + 5 # u in terms of x
    # x = x(u) the backwards transform
    x_of_u = lambda u: (u - 5)/3 # x in terms of u
    fig = plt.figure()
    fig.suptitle("A demonstration of coordinate transforms")
    ax1 = fig.add_subplot(3, 1, 1)
    ax1.set_title("$f(x)$ in $(x, y)$ coordinates")
    f = lambda x: np.sin(x)
    # plot in x -space
    ax1.plot(x, f(x))
    # plot in u -space
    ax2 = fig.add_subplot(3, 1, 2)
    ax2.set_title("$g(u)$ in $(u, v)$ coordinates")
    u = u_of_x(x)
    g = lambda u: np.sin(u)
    ax2.plot(u, g(u), linestyle=" - -")
    ax3 = fig.add_subplot(3, 1, 3)
    ax3.set_title("$f(x)$ reconstructed from $g(u)$")
    ax3.plot(x, g(x_of_u(u)))
    plt.subplots_adjust(hspace=0.75)
    plt.show()

test_coord_transform()
```



To solve the PDE in the new coordinates, the PDE itself must be converted to the new coordinates. By the chain rule, it can be shown that the new form the vector-valued PDE takes is given by:

$$\left(u\frac{\partial}{\partial u}\left(u\frac{\partial}{\partial u}\right)+v\frac{\partial}{\partial v}\left(v\frac{\partial}{\partial v}\right)+k^2\right)\tilde{E}_u(u,v)=0 \tag{1132}$$

$$\left(u\frac{\partial}{\partial u}\left(u\frac{\partial}{\partial u}\right)+v\frac{\partial}{\partial v}\left(v\frac{\partial}{\partial v}\right)+k^2\right)\tilde{E}_v(u,v)=0 \tag{1133}$$

Note that in the numerical programming we must expand out the partial derivatives. The expanded version of the Helmholtz operator is:

$$u\left(\frac{\partial}{\partial u}+u\frac{\partial^2}{\partial u^2}\right)+u\left(\frac{\partial}{\partial u}+u\frac{\partial^2}{\partial u^2}\right)+k^2 \tag{1134}$$

In addition, we will convert the boundary conditions respectively to:

$$\tilde{E}_u(u,1)=2,\tilde{E}_u(u,e)=7,\tilde{E}_u(1,v)=0.5,\tilde{E}_u(e,v)=\pi \tag{1135}$$

$$\tilde{E}_v(u,1)=2\pi,\tilde{E}_v(u,e)=3,\tilde{E}_v(1,v)=12,\tilde{E}_v(e,v)=1 \tag{1136}$$

```
# forward transformations
u_of_x = lambda x: np.exp(x)
v_of_y = lambda y: np.exp(y)

U = u_of_x(X)
V = v_of_y(Y)
```

We can see that the domain $[0,1]\times[0,1]$ in (x,y) maps to $[1,e]\times[1,e]$ in (u,v) coordinates:

```
U.min(), U.max()
```

```
(np.float64(1.0), np.float64(2.718281828459045))
```

Consider the first boundary condition $E_x(x,0)=2$. We can show that it takes the form $E_u(u,1)=2$ below:

```
example_Ex_func = lambda x, y: 2

example_Eu_func = example_Ex_func(u_of_x(x), v_of_y(y))

example_Eu_func
```

```
2
```

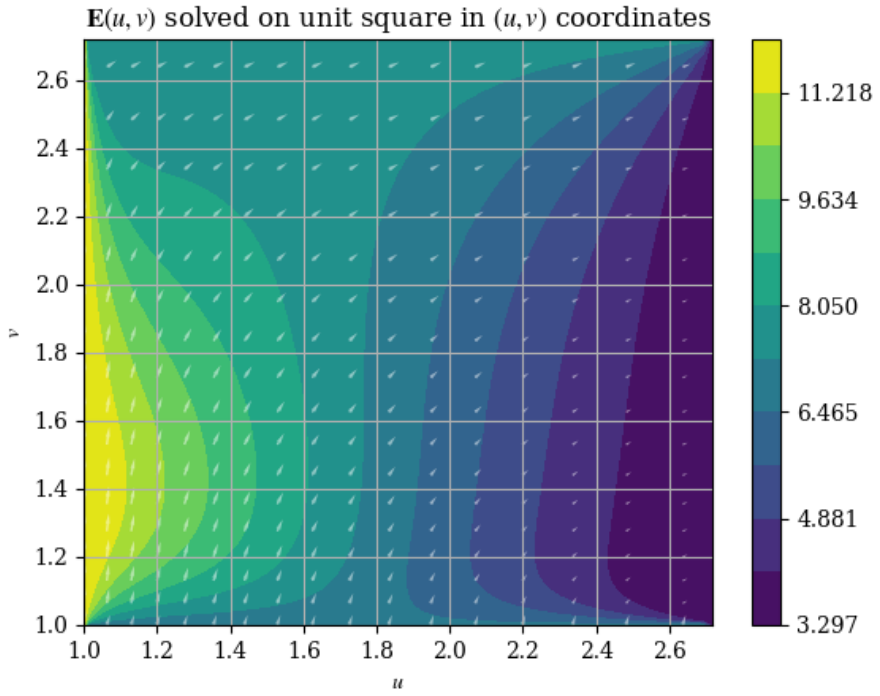
```
# again incapsulate in function to prevent variable
# overwrite in global scope
# if refactoring is best to make a class for everything
def solve_transformed_helmholtz(x=x, y=y, k=k, shape=shape):
    u = u_of_x(x)
    v = v_of_y(y)
    du = u[1] - u[0]
    dv = v[1] - v[0]
    # not sure if we need to convert to meshgrid for this
    U, V = np.meshgrid(u, v, indexing='ij')
    d_du = FinDiff(0, du)
    d_ddu = FinDiff(0, du, 2)
    d_dv = FinDiff(1, dv)
    d_ddv = FinDiff(1, dv, 2)
```

```
    # helmholtz operator
    # Helmholtz = Coef(U)*d_du + Coef(U**2)*d_ddu + Coef(V)*d_dv + Coef(V**2)*d_ddv + Coef(k**
    Helmholtz = Coef(U)*(d_du + Coef(U)*d_ddu) + Coef(V)*(d_dv + Coef(V)*d_ddv) + Coef(k**2)*I
    rhs = np.zeros(shape)
    # we need separate boundary conditions for E_u and E_v components
    # of the vector -valued PDE
    bc_u = BoundaryConditions(shape)
    bc_v = BoundaryConditions(shape)
    # here note that we set based on the boundaries
    # by index not by value
    # the domain is [1, e] x [1, e]
    bc_u[:, 0] = 2.0 # E_u(u, 1)
    bc_u[:, -1] = 7.0 # E_u(u, e)
    bc_u[0, :] = 0.5 # etc.
    bc_u[ -1, :] = np.pi

    bc_v[:, 0] = 2*np.pi
    bc_v[:, -1] = 3
    bc_v[0, :] = 12
    bc_v[ -1, :] = 1

    # PDE for u component of electric field
    # the Helmholtz operator is identical for both PDEs
    pde_u = PDE(Helmholtz, rhs, bc_u)
    pde_v = PDE(Helmholtz, rhs, bc_v)
    solution_u = pde_u.solve()
    solution_v = pde_v.solve()
    return solution_u, solution_v

transform_u, transform_v = solve_transformed_helmholtz()
```

After solving for the components of $\mathbf{E}$, we can visualize $\mathbf{E}$ as a vector field as follows:

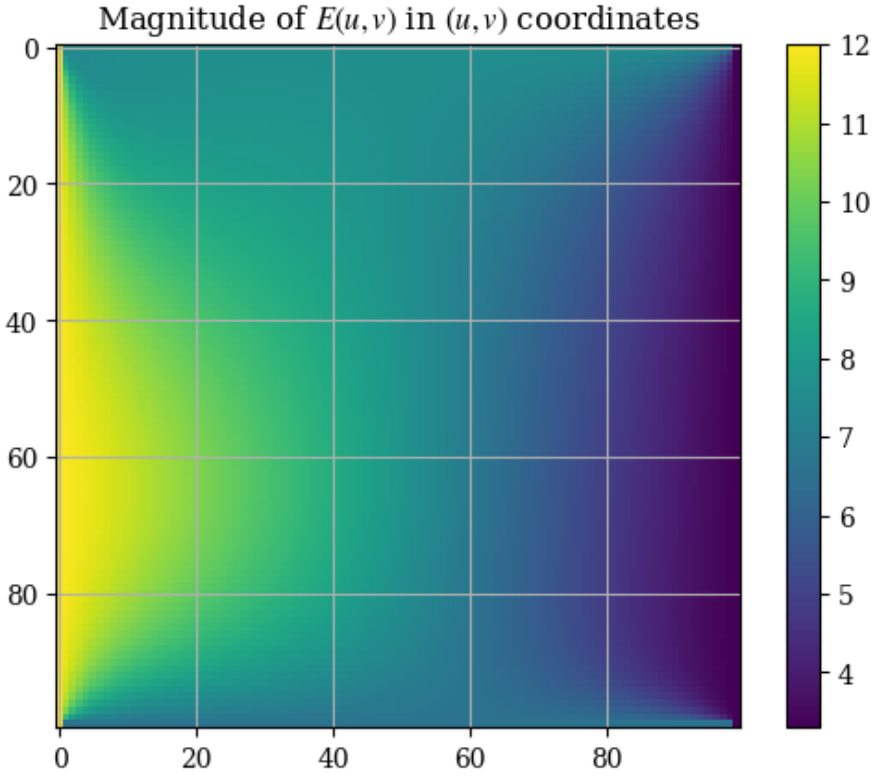
```
def magnitude(E1, E2):
    squared_norm = E1**2 + E2**2
    return np.sqrt(squared_norm)

def plotE_uvspace(u=U, v=V, Usol=transform_u, Vsol=transform_v, desc=None, opacity=0.5):
    vec_density = 6 # plot one vector for every 10 points
    contour_levels = 12 # number of contours (filled isocurves) to plot
    transform_mag = magnitude(transform_u, transform_v)
    levels = np.linspace(transform_mag.min(), transform_mag.max(), contour_levels)
    # plot the filled isocurves
    plt.contourf(u, v, transform_mag, levels=levels)
    plt.colorbar()
    # plot the vectors
    plt.quiver(u[::vec_density, ::vec_density], v[::vec_density, ::vec_density], Usol[::vec_de
    plt.xlabel("$u$")
    plt.ylabel("$v$")
    if not desc:
        plt.title(r"$\mathbf{E}(u, v)$ solved on unit square in $(u, v)$ coordinates")
    else:
        plt.title(desc)
    plt.show()
```

```
plotE_uvspace()
```



And the magnitude is respectively given by $E = \|\mathbf{E}\| = \|E_u\hat{\mathbf{u}} + E_v\hat{\mathbf{v}}\| = \sqrt{{E_u}^2 + {E_v}^2}$, which can be plotted as shown:

```
plt.imshow(correct_axes(magnitude(transform_u, transform_v)), interpolation="none")
plt.title("Magnitude of $E(u, v)$ in $(u, v)$ coordinates")
plt.colorbar()
plt.show()
```



Note that the axes tickmarks can be ignored (they are not accurate) , as `imshow()` treats the input as if it were an image (which it obviously is not). We can validate the solution with the analytical

expressions for the magnitude calculated from the boundary conditions for E_u and E_v:

```
E_prime_bottom = magnitude(2, 2*np.pi)
E_prime_top = magnitude(7, 3)
E_prime_left = magnitude(0.5, 12)
E_prime_right = magnitude(np.pi, 1)
print(f"Analytical values - top: {E_prime_top:.3f}, right: {E_prime_right:.3f}, bottom: {E_pri
```

```
Analytical values - top: 7.616, right: 3.297, bottom: 6.594, left: 12.010
```

Which agree reasonably with the values shown in the plot. We will now do the comparison with the finite element solution for the validation.

```
# convert transformed -space (u, v)
# coordinates to index (m, n)
# of the solution array
def convert_coord_index_uv(u, v):
    u_idx = round((u -1)/(np.e -1)*squaren)
    v_idx = round((v -1)/(np.e -1)*squaren)
    # prevent out of bound
    # access errors
    if u_idx == squaren:
        u_idx = -1
    if v_idx == squaren:
        v_idx = -1
    return (u_idx, v_idx)

def validate_vector_uv():
    # here (u, v) is the domain [1, e] x [1, e]
    points = [
        [np.e/2, 1], # bottom
        [1, np.e/2], # left
        # the remainder are random points
        [1.3, 2.2],
        [1.7, 1.65],
        [2.4, 2.5]
    ]

    for p in points:
        idx_u, idx_v = convert_coord_index_uv(*p)
        transform_mag = magnitude(transform_u, transform_v)
        res = transform_mag[idx_u, idx_v]
        print(f"On test point ({p[0]:.3f}, {p[1]:.3f}), magnitude {res:.4f}")

validate_vector_uv()
```

```
On test point (1.359, 1.000), magnitude 6.5938
On test point (1.000, 1.359), magnitude 12.0104
On test point (1.300, 2.200), magnitude 9.0540
On test point (1.700, 1.650), magnitude 8.5958
On test point (2.400, 2.500), magnitude 6.3063
```

To evaluate the solution (x, y) coordinates, we remap each of the points from (u, v) space to (x, y) space, that is, applying the inverse transforms $x(u) = \ln u$ and $y(v) = \ln v$ (again the prime here denotes transformation, it is not a derivative symbol. As the solution is numerical (and therefore

discrete), we must interpolate it to find $\tilde{\mathbf{E}}(u, v)$ so that we can calculate the correct values according to the formula $\mathbf{E}(x, y) = \tilde{\mathbf{E}}(x(u), y(v))$. For this we use `scipy.optimize.curve_fit` with a cubic polynomial in the form $f(x, y) = ax^3 + by^3 + cx^2y^2 + dx^2 + gy^2 + hxy + mx + nx + r$ on both components of $\mathbf{E}'$:

$$E_u(u, 1) = 2, E_u(u, e) = 7, E_u(1, v) = 0.5, E_u(e, v) = \pi \tag{1137}$$
$$E_v(u, 1) = 2\pi, E_v(u, e) = 3, E_v(1, v) = 12, E_v(e, v) = 1 \tag{1138}$$

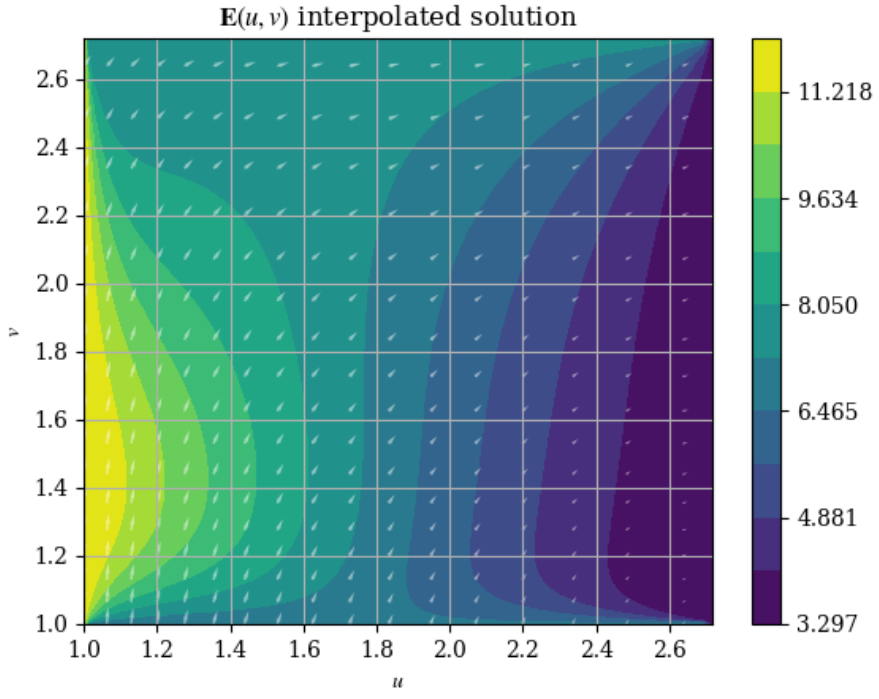
```
# have to flatten arrays to make this work
def interpolation_func(X, a=1, b=1, c=1, d=1, g=1, h=1, m=1, n=1, r=1):
    u_raw, v_raw = X
    squaren = n # change based on the value of squaren globally
    x = u_raw
    y = v_raw
    out = a*x**3 + b*y**3 + c*x**2*y**2 + d*x**2 + g*y**2 + h*x*y + m*x + n*x + r
    return out.flatten()

sol_u, cov_u = optimize.curve_fit(interpolation_func, (U.flatten(), V.flatten()), transform_u.

sol_v, cov_v = optimize.curve_fit(interpolation_func, (U.flatten(), V.flatten()), transform_v.
```

We can then use the interpolated function as usual functions, `E_u(u, v)` and `E_v(u, v)` that can be given arguments, which represents $\mathbf{E}'(u, v)$:

```
# these take in vector -valued inputs i.e. you need to use np.meshgrid for them
E_u = lambda u, v: interpolation_func((u.flatten(), v.flatten()), *sol_u).reshape(squaren, squ
E_v = lambda u, v: interpolation_func((u.flatten(), v.flatten()), *sol_v).reshape(squaren, squ
Emag_uv = lambda u, v: magnitude(E_u(u, v), E_v(u, v))
```

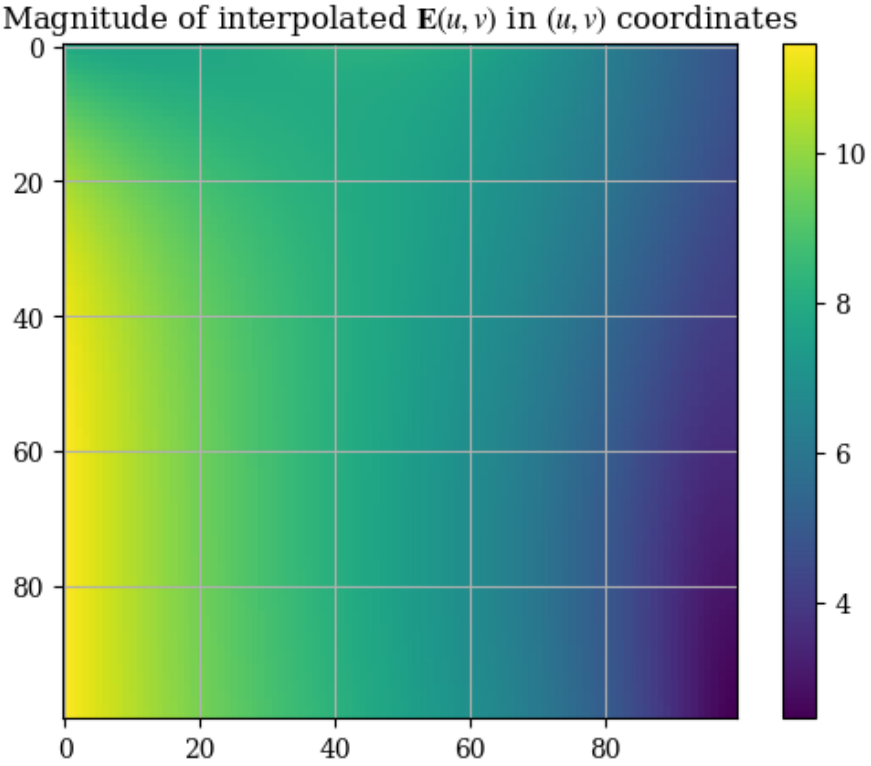
And we can plot it just like the original numerical solution:

```
plotE_uvspace(Usol=E_u(U, V), Vsol=E_v(U, V), desc=r"$\mathbf{E}(u, v)$ interpolated solution"
```



```
plt.imshow(correct_axes(Emag_uv(U, V)), interpolation="none")
plt.title("Magnitude of interpolated $\mathbf{E}(u, v)$ in $(u, v)$ coordinates")
plt.colorbar()
plt.show()
```

```
<>:2: SyntaxWarning: invalid escape sequence '\m'
<>:2: SyntaxWarning: invalid escape sequence '\m'
C:\Users\Jacky\AppData\Local\Temp\ipykernel_92892\1675798493.py:2: SyntaxWarning: invalid esca
  plt.title("Magnitude of interpolated $\mathbf{E}(u, v)$ in $(u, v)$ coordinates")
```

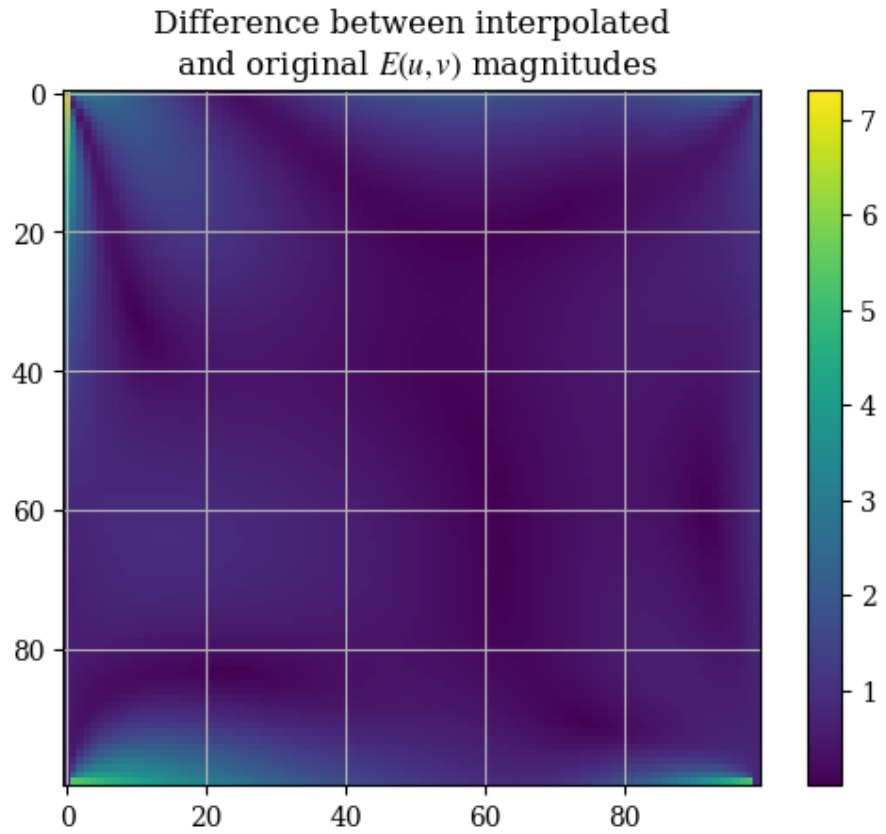


We can compare the interpolation functions' accuracy with the original numerical solution:

```
interpolation_err_u = np.abs(E_u(U, V) - transform_u)
interpolation_err_v = np.abs(E_v(U, V) - transform_v)
```

For instance, we can find difference of the interpolation as compared to the numerical solution:

```
interpolation_err = magnitude(interpolation_err_u, interpolation_err_v)

plt.imshow(correct_axes(interpolation_err), interpolation="none")
plt.title("Difference between interpolated\n and original $E(u, v)$ magnitudes")
plt.colorbar()
plt.show()
```



We can see that the interpolation is quite good *except* for the boundaries. This can be seen from the median and mean of the interpolation error, which is (relatively) small:

```
np.median(interpolation_err)
```

```
np.float64(0.4978049155751837)
```

```
np.mean(interpolation_err)
```

```
np.float64(0.6408549856200115)
```

We can also evaluate the numerial vs interpolated solution next to each other on a common set of points:

```
# we use pointwise variants of the interpolation function as opposed
# to the one that takes vectorized input
E_u_pointwise = lambda u, v: interpolation_func((u, v), *sol_u)
E_v_pointwise = lambda u, v: interpolation_func((u, v), *sol_v)
E_uv_pointwise = lambda u, v: float(np.sqrt(E_u_pointwise(u, v)**2 + E_v_pointwise(u, v)**2))

def validate_interpolate_vs_num_uv():
    # here (u, v) is the domain [1, e] x [1, e]
    points = [
        [(np.e -1)/2, 1], # bottom
        [1, (np.e -1)/2], # left
        # the remainder are random points
        [1.3, 2.2],
        [1.7, 1.65],
        [2.4, 2.5]
    ]
```

```
    print("Comparison of magnitudes (numerical vs interpolated)")
    for p in points:
        idx_u, idx_v = convert_coord_index_uv(*p)
        transform_mag = magnitude(transform_u, transform_v)
        res = transform_mag[idx_u, idx_v]
        interp_res = E_uv_pointwise(*p)
        print(f"On test point ({p[0]:.3f}, {p[1]:.3f}), numerical solution {res:.4f}, interpol

validate_interpolate_vs_num_uv()
```

```
Comparison of magnitudes (numerical vs interpolated)
On test point (0.859, 1.000), numerical solution 6.5938, interpolated solution 13.2166
On test point (1.000, 0.859), numerical solution 12.0104, interpolated solution 11.5047
On test point (1.300, 2.200), numerical solution 9.0540, interpolated solution 8.5078
On test point (1.700, 1.650), numerical solution 8.5958, interpolated solution 7.4088
On test point (2.400, 2.500), numerical solution 6.3063, interpolated solution 5.6210

C:\Users\Jacky\AppData\Local\Temp\ipykernel_92892\1914026686.py:5: DeprecationWarning: Convers
  E_uv_pointwise = lambda u, v: float(np.sqrt(E_u_pointwise(u, v)**2 + E_v_pointwise(u, v)**2)
```

We can also check for the boundary conditions against their analytical values:

```
print(f"Analytical values - top: {E_prime_top:.3f}, right: {E_prime_right:.3f}, bottom: {E_pri
```

```
Analytical values - top: 7.616, right: 3.297, bottom: 6.594, left: 12.010
```

```
half_u = (np.e - 1)/2
half_v = (np.e - 1)/2
E_interp_uv_top = E_uv_pointwise(half_u, np.e)
E_interp_uv_right = E_uv_pointwise(np.e, half_v)
E_interp_uv_bottom = E_uv_pointwise(half_u, 1)
E_interp_uv_left = E_uv_pointwise(1, half_v)
```

```
C:\Users\Jacky\AppData\Local\Temp\ipykernel_92892\1914026686.py:5: DeprecationWarning: Convers
  E_uv_pointwise = lambda u, v: float(np.sqrt(E_u_pointwise(u, v)**2 + E_v_pointwise(u, v)**2)
```

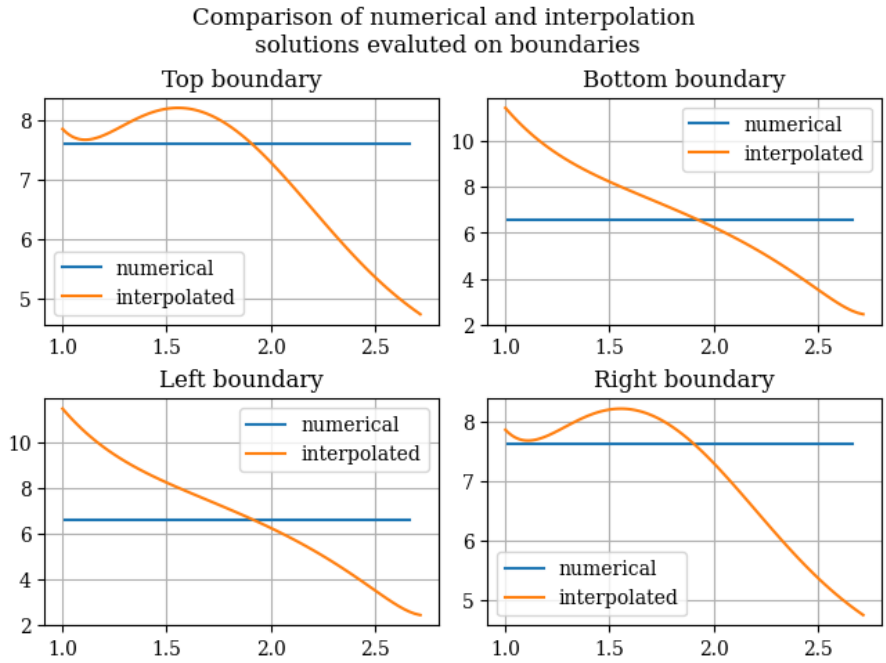
```
print(f"Interpolated values - top: {E_interp_uv_top:.3f}, right: {E_interp_uv_right:.3f}, bott
```

```
Interpolated values - top: 8.898, right: 2.265, bottom: 13.217, left: 11.505
```

Thus, we see that our interpolation is primarily poor at the boundaries (most significantly, the bottom boundary), even though it is not bad at the center of the domain. To examine the issue further, we can plot the boundaries obtained from the original numerical solution as well as the transformation, which is shown below:

```
def compare_uv_bcs(Emag_numerical = magnitude(transform_u, transform_v), Emag_interp = Emag_uv
    fig = plt.figure(layout='constrained')
    if not desc:
        fig.suptitle("Comparison of numerical and interpolation\n solutions evaluted on bounda
    else:
        fig.suptitle(desc)
    spec = fig.add_gridspec(ncols=2, nrows=2)
    ax1 = fig.add_subplot(spec[0, 0])
    ax1.set_title("Top boundary")
    # we use [1: -2] because we don't want the endpoints which are connected
    # to the nodes of other boundaries
```

```
    ax1.plot(u[1: -2], Emag_numerical[:, -1][1: -2], label="numerical")
    ax1.plot(u, Emag_interp[:, -1], label="interpolated")
    ax1.legend()
    ax2 = fig.add_subplot(spec[0, 1])
    ax2.set_title("Bottom boundary")
    ax2.plot(u[1: -2], Emag_numerical[:, 0][1: -2], label="numerical")
    ax2.plot(u, Emag_interp[:, 0], label="interpolated")
    ax2.legend()
    ax3 = fig.add_subplot(spec[1, 0])
    ax3.set_title("Left boundary")
    ax3.plot(v[1: -2], Emag_numerical[:, 0][1: -2], label="numerical")
    ax3.plot(v, Emag_interp[:, 0], label="interpolated")
    ax3.legend()
    ax4 = fig.add_subplot(spec[1, 1])
    ax4.set_title("Right boundary")
    ax4.plot(v[1: -2], Emag_numerical[:, -1][1: -2], label="numerical")
    ax4.plot(v, Emag_interp[:, -1], label="interpolated")
    ax4.legend()
    plt.show()

compare_uv_bcs()
```



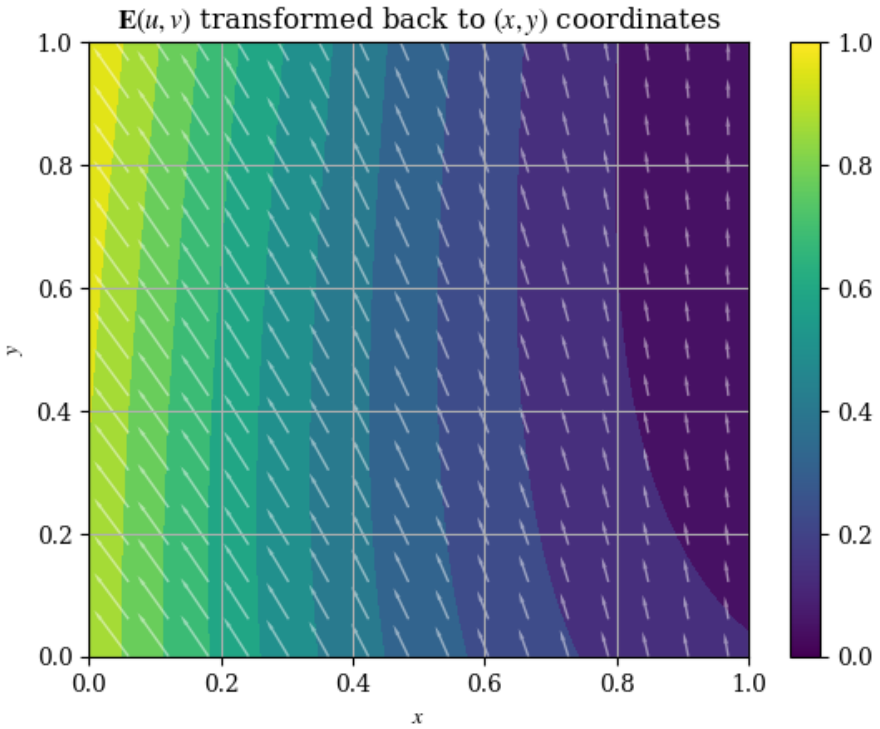
This confirms our theory that the boundaries are where the issues lie. This means we need to do more testing to find a fitting method that respects the boundary conditions.

But this is not enough. We then need to apply the backwards transforms, which, like the forward transforms, preserves the features of the function, including having constant boundary conditions just like the original boundary conditions.

Finally, we evaluate the original function $\mathbf{E}(x, y)$ through the interpolated version of $\mathbf{E}'(u, v)$, via $\mathbf{E}(x, y) = \mathbf{E}'(x(u), y(v))$. We have $x(u) = \ln u, y(v) = \ln v$. Evaluating these on the arrays of u and v recovers $\mathbf{E}(x, y)$.

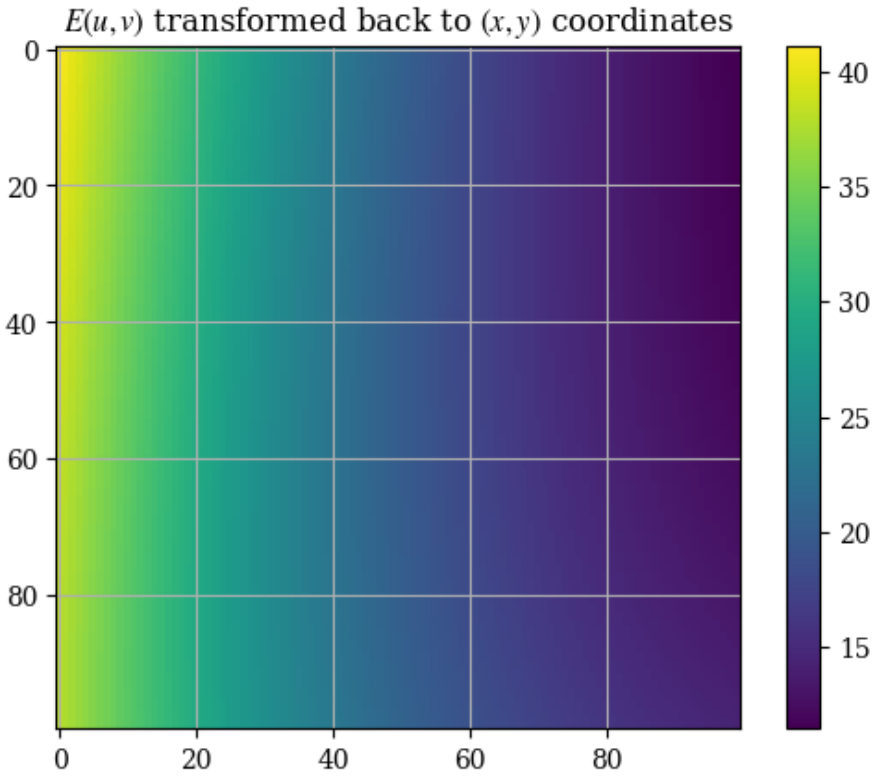
```
# backwards transforms
x_of_u = lambda u: np.log(u)
y_of_v = lambda v: np.log(v)
```

```
E_x = E_u(x_of_u(U), y_of_v(V))
E_y = E_v(x_of_u(U), y_of_v(V))

def plotE_xyspace(x=X, y=Y, E_x=E_x, E_y=E_y, desc=None, opacity=0.5):
    vec_density = 6 # plot one vector for every 10 points
    contour_levels = 12 # number of contours (filled isocurves) to plot
    E_mag = magnitude(E_x, E_y)
    levels = np.linspace(E_mag.min(), E_mag.max(), contour_levels)
    # plot the filled isocurves
    plt.contourf(x, y, E_mag, levels=levels)
    # plot the vectors
    plt.quiver(x[::vec_density, ::vec_density], y[::vec_density, ::vec_density], E_x[::vec_den
    plt.xlabel("$x$")
    plt.ylabel("$y$")
    if not desc:
        plt.title(r"$\mathbf{E}(u, v)$ transformed back to $(x, y)$ coordinates")
    else:
        plt.title(desc)
    plt.colorbar()
    plt.show()

plotE_xyspace()
```



We may also plot the magnitude of our transformed solution:

```
E_xy_mag = correct_axes(magnitude(E_x, E_y))
plt.title(r"$E(u, v)$ transformed back to $(x, y)$ coordinates")
plt.imshow(E_xy_mag)
plt.colorbar()
plt.show()
```


$E(u,v)$ transformed back to (x,y) coordinates
0
20
40
60
80
0
20
40
60
80
40
35
30
25
20
15

0.3 The expert's guide

0.3.1 Foreword to the Expert's Guide

Whether you've come from the last two chapters looking for more, or simply want a challenge, this section is a comprehensive guide to prepare the theoretical foundations for, and then introduce, Project Elara's most advanced research.

0.3.2 Computational physics

Physics has long been a key adopter of numerical methods. To name a few, particle physics, gravitational-wave interferometry, astrophysics, and molecular dynamics all make heavy use of numerical methods for problems that would otherwise be near impossible to solve by hand. In the following chapters, we will explain exactly *how* physicists (and engineers) take real-world problems and solve them using a computer.

Unlike earlier sections, which focused on *application*, these chapters are focused on developing the *theory* behind numerical computing and its applications to simulating physical systems.

Introduction to computational physics

```
import numpy as np
import scipy as sp
# import astropy.units as u
```

Numerical methods

Topics in numerical methods Numerical methods are focused on ways to solve problems that, in general, cannot be solved *exactly*.

> **Note**
> In this chapter, we will focus on writing codes that solve numerical problems that have reasonably good accuracy and performance, which are sufficient for most applications. We will *not* be trying to write super-optimized codes designed for high-performance and massively parallel computing.

We will describe both how to write these solvers from scratch, as well as how to write. For each, we will provide code examples using vanilla Python (with NumPy) and other libraries as needed.

ODEs and initial-value problems In our discussion on differential equations, we mentioned that ordinary differential equations take the general form:

$$\frac{dy}{dt} = f(t, y) \tag{1139}$$

(note that y can be a vector, and reduction of order may be necessary to cast a particular ODE in this form). An **initial-value problem** is a description of a physical scenario in which an ODE is provided along with the *initial conditions*, that is, $y(0)$, the value of y at $t = 0$. Often, we cannot solve initial-value problems exactly, and some initial-value problems are so complicated that even approximate solutions cannot be found. We therefore turn to *numerical methods*.

The general basis of solving an initial-value problem numerically is to discretize $y(x)$ into a set of line segments. Each segment connects the nth-point (t_n, y_n) to its next point (t_{n+1}, y_{n+1}). By starting at the known initial condition $(0, y(0))$, we may then use the fundamental of calculus to calculate the next point:

$$y_{n+1} = \int_{t_n}^{t_{n+1}} f(t, y_n) dt \tag{1140}$$

This process is known as **numerical integration** and is the principle behind solving ODEs. Different ways of computing the integral numerically result in a diverse class of specialized numerical integration schemes, each of which has its own advantages and disadvantages.

forwards Euler, backwards Euler, midpoint, leapfrog, runge-kutta 2 & 4, also some bit about symeptic integrators

PDEs and boundary-value problems The most common methods of solving boundary-value problems numerically are as follows:

- Finite difference methods, which approximate derivatives (which are limits of the difference quotient) with finite difference quotients, so a PDE can be reformulated as a system of equations and solved with some linear algebra. Subcategories of this include the method of lines. This category of numerical methods usually only work on a regular and discrete grid, so it is out of the question for our numerical work, which involves very nontrivial geometries for parabolic reflectors with openings and secondary mirrors
- Finite element methods as well as the related spectral methods first turn a PDE into an integral equation. The solution is approximated as a linear combination of basis functions (like simple polynomials that can be easily integrated) and their coefficients. The domain is then discretized so the integral can then be evaluated, resulting (again) in a system of equations and solved with some linear algebra.
- Finite volume methods involve cleverly rewriting the PDE in the form of a continuity equation, which can be converted into an equivalent integral equation that is then discretized and solved. This has the advantage of preserving conservation laws in the system, and in the future we will probably use these a lot more, but I haven't tried them yet.

The finite difference method More info here: https://labs.elaraproject.org/theoretical/Guide-to-partial-differential-equations.html#numerically-solving-pdes

The finite element method a) Assume the solution takes the form $u(\mathbf{x}) = \sum_i \varphi_i \phi_i(\mathbf{x})$ where $\phi_i(x)$ is a set of functions that form a complete basis. A popular choice is the Chebyshev polynomials, you can also use Legendre or Lagrange polynomials, but any basis will do. The more terms in the series expansion/approximation, the more accurate your solution will be. b) By autodiff (or just hard-coded derivative rules), substitute $u(x) = \sum_i \varphi_i \phi_i(\mathbf{x})$ into the ordinary/partial differential equation in question where the ordinary/partial differential equation is written in the form $F(u, \nabla u, \nabla^2 u) = 0$ and F is some combination of linear and nonlinear operators. This substitution will result in a system of (possibly nonlinear) equations in the form $\mathbf{G}(\mathbf{x}) = 0$. c) Solve this system of homogenous equations either with a linear algebra solver or Newton's method for nonlinear

Machine learning for differential equations

See https://codeberg.org/elaraproject/elara-pdes-tutorial

0.3.3 Developer guide

Design Philosophy

Project Elara's software should be robust, easily maintainable, and simple - requirements essential to building software that lasts for generation upon generation. Therefore, we insist on the following principles for software development:

1. Each library (or software) is intended to do one thing well. Instead of overloading existing software with more features, make new software instead.
2. Keep the (dependency-excluded) codebase of software under 10K lines of code, and preferably under 5K lines of code.
3. Use only the project's own libraries as dependencies when possible, and minimize use of dependencies
4. Keep everything licensed under public domain, as with the rest of Project Elara

The reason for the first principle is that software that has too many features easily becomes unmaintainable. More features means more complexity, which means a more difficult-to-understand codebase, which restricts work on a codebase to those who are already familiar with the software. We want everyone to be able to contribute, so we want to avoid that as possible. The reasoning is similar withe second rule. We want software that can be modified for generations, so we need the codebase to be manageably small.

The reason for the third principle is also similar - dependencies lead to more complexity, and to more code that could lead to bugs. In general, fewer code leads to fewer bugs, which leads to more maintainable software. The use of the project's own libraries is intended to protect the independence of the project from dependency interference. Finally, the public domain licensing is needed to keep the software always free and available to everyone.

0.3.4 Theoretical physics

Project Elara's main work (for now, at least) is to develop peaceful space-based solar energy technologies. However, there are many technologies that, while not directly developed by Project Elara, are enabled by (or adjacent to) our technology. We believe it fitting to at least include a discussion of them in the Handbook.

Some of these technologies are ones that are already in development or are conceptually well-established (e.g. solar sails). Others are far more speculative and are presently outside the realm of possibility (e.g. interstellar spaceflight). And still others are at the frontiers of physics itself. Regardless, we will touch on all of them, both for educational purposes (as a good way to gain a better understanding of physics and engineering) and in the spirit of sharing and passing our knowledge to others.

In addition, we will give an overview on advanced mathematics and physics, which form part of the necessary theoretical background to understand advanced topics. This includes special relativity, general relativity and relativistic quantum field theory. while Project Elara's research (mostly) does not involve advanced theoretical physics, these chapters are meant for sharing our knowledge and in case the Project will have applications that require an understanding of these topics in the future. We hope we can explain them in a way that is enriching and meaningful.

> "My methods are really methods of working and thinking; this is why they have crept in everywhere anonymously."
>
> **Emmy Noether**

Special Relativity

> "Henceforth space by itself, and time by itself, are doomed to fade away into mere shadows, and only a kind of union of the two will preserve an independent reality..."
> **Hermann Minkowski (1908)**

For over 200 years, the basic premises of Newtonian mechanics remained absolute. Space and time were separate entities, and the laws of physics followed Newton's postulates. That is, until 1905, when Einstein broke both previously-assumed absolute truths, and set out to create a new theory. This groundbreaking theory was the theory of **Special Relativity**, a description of the motion of objects that supercedes Newtonian mechanics and completely revolutionized our understanding of the world.

```
import matplotlib.pyplot as plt
import numpy as np
```

Events An event is anything that happens that can be measured. This can be a rocket flying through your window, a book that falls on your head, a crater that opens in your room, or all three at once! We can describe events with coordinates - perhaps your rocket-falling-book-crater event happened at position 3 meters to your right, 2 meters to your front, and 0 meters above your head, at time 2:55pm. Physicists obsess over events, because everything in physics is composed of a sequence of events, and so using the laws of physics, physicists can predict what events will happen.

Galilean Relativity Imagine you were seated aboard a train, and the train was moving with constant velocity. Are you moving and the earth underneath you stationary? Or is the train stationary and the earth is moving under you? In physics, both of these interpretations can be true - your understanding of your motion must be considered **relative** to some other object. For instance, you can pick the stationary object to be the earth, in which case you would be considered moving, or you could pick the stationary object to be yourself on the train, in which case the Earth would be moving. In either case, the laws of physics remain the same.

Reference frames A reference frame is a coordinate system with an origin centered at a chosen location. For example, you can choose your origin to be a house on the surface of the Earth, a moving train, a rocket, even a random point in interstellar space. You would then have a reference frame at that origin.

Typically speaking, however, when we refer to the reference frame of an observer, the origin is located at wherever the observer is located. So the reference frame of an astronaut in a rocket would have an origin centered at that rocket. In the astronaut's reference frame, they themselves are located at the point $(0, 0, 0)$, and everything else (such as the motion of the Earth) is measured relative to them.

We use reference frames to measure everything around us, everything from the position and velocities of objects to forces between objects. In fact, without reference frames, we wouldn't be able to measure any motion at all.

Transformations In physics, it is sometimes easier to do calculations in one reference frame than another - no one would like to compute the trajectory of a baseball on Earth from the reference frame of another galaxy! So, we want to be able to convert measurements from one reference frame to another.

The reference frame of an observer is known as the **unprimed frame**. The observer is at rest with respect to the unprimed frame, and everything is moving relative to them. Any measurement in the unprimed frame is described by coordinates (t, x, y, z).

Everything moving relative to the observer has a reference frame of its own. For example, our astronaut could be observing Earth, the Sun, the Moon, a space station - and each of those objects has its own reference frame. The observer measures the velocity v at which each reference frame moves

with respect to the observer. These other moving reference frames are known as **primed frames** because they are denoted with the prime ($'$) symbol. Any measurement in a primed frame is described by the coordinates (t', x', y', z').

These sets of different measurements are related as follows:

$$x' = x - vt \tag{1141}$$
$$y' = y \tag{1142}$$
$$z' = z \tag{1143}$$
$$t' = t \tag{1144}$$

> **Note**
> The primes here are not derivative symbols, they're used in this context to denote "different" or "alternate"

One thing we specifically notice is that here, time is **absolute** - the same in every reference frame. As we will see, this will no longer hold true in special relativity.

The constant speed of light In the late 19th-century, physicists finally came up with one unified theory of electromagnetism using Maxwell's equations, which we saw earlier. Recall that the equations are given by:

$$\nabla \cdot \vec{E} = \frac{\rho}{\epsilon_0} \tag{1145}$$
$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{1146}$$
$$\nabla \cdot \vec{B} = 0 \tag{1147}$$
$$\nabla \times \vec{B} = \mu_0 \vec{J} + \mu_0 \epsilon_0 \frac{\partial \vec{E}}{\partial t} \tag{1148}$$

Let's take Faraday's law:

$$\nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t} \tag{1149}$$

Suppose we take the curl of both sides:

$$\nabla \times (\nabla \times \vec{E}) = \nabla \times -\frac{\partial \vec{B}}{\partial t} \tag{1150}$$

The right hand side can be rearranged to be:

$$\nabla \times (\nabla \times \vec{E}) = -\frac{\partial}{\partial t}(\nabla \times \vec{B}) \tag{1151}$$

Which simplifies to:

$$\nabla \times (\nabla \times \vec{E}) = -\mu_0 \epsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2} \tag{1152}$$

We can use the vector identity $\nabla \times (\nabla \times \vec{E}) = -\nabla^2 E$ to simplify further to:

$$\nabla^2 E = \mu_0 \epsilon_0 \frac{\partial^2 \vec{E}}{\partial t^2} \tag{1153}$$

Using the same technique for Ampére's law and then Faraday's law yields the same result for magnetic fields:

$$\nabla^2 B = \mu_0 \epsilon_0 \frac{\partial^2 \vec{B}}{\partial t^2} \tag{1154}$$

Note that this looks very much like the wave equation:

$$\nabla^2 f = \frac{1}{v^2} \frac{\partial^2 f}{\partial t^2} \tag{1155}$$

Which means that oscillating electric and magnetic fields produce electromagnetic waves that move through space at a speed v. We can solve for v by noting that:

$$\frac{1}{v^2} = \mu_0 \epsilon_0 \tag{1156}$$

This yields:

$$v = \frac{1}{\sqrt{\mu_0 \epsilon_0}} = 299792458 \text{ m/s} = c \tag{1157}$$

Now notice something special. The velocity c of electromagnetic waves - light waves - is a constant, because it is composed of the reciprocal of the square root of two other constants. This means that regardless of the velocity of the reference frame, it must be the same speed.

However, remember that in Galilean relativity, we defined that velocities add by $\vec{v} + \vec{u}$. So we'd expect that an observer in a moving reference frame would measure a higher speed of light, while observers in a stationary reference frame would measure a lower speed of light. Through numerous experiments, this was proven not to be the case - we are certain that the speed of light is constant, regardless of the reference frame.

Therefore, the Galilean transformations must be wrong, and a new set of transformations - the Lorentz transformations - must supercede them.

The Lorentz Transformations The Lorentz transformations are Einstein's revision to Galilean relavity, derived from two postulates:

- The laws of physics hold true in every reference frame
- The speed of light c is constant in every reference frame

To derive the Lorentz transformations, let's start with the Galilean transformations for $x \to x'$ and $x' \to x$:

$$x' = x - vt \tag{1158}$$
$$x = x' + vt' \tag{1159}$$

To correct Galilean coordinate transformations, we intuitively need to multiply the Galilean transformations by a factor γ, which we can think of as the "correcting factor" to make sure that the Galilean transforms preserve the speed of light in every reference frame:

$$x' = \gamma(x - vt) \tag{1160}$$
$$x = \gamma(x' + vt') \tag{1161}$$

Now, we can multiply the left and right hand sides of the equation together, to combine the two equations into one equation:

$$x'x = \gamma(x - vt)\gamma(x' + vt') \tag{1162}$$
$$x'x = \gamma^2(xx' + xvt - x'vt - v^2tt') \tag{1163}$$

Remember the second postulate of special relativity is that the speed of light is an invariant in every reference frame - that is $c = \frac{x}{t} = \frac{x'}{t'}$. Rearranging, we can say that $x = ct$ and $x' = ct'$. Substituting that in, we have:

$$c^2tt' = \gamma^2(c^2tt' + ctvt - ct'vt - v^2tt') \tag{1164}$$

The two middle terms cancel each other out, so we have:

$$c^2tt' = \gamma^2(c^2tt' - v^2tt') \tag{1165}$$

We isolate γ^2 by dividing the right-hand side of the equation by the left, to obtain:

$$\gamma^2 = \frac{c^2tt'}{c^2tt' - v^2tt'} \tag{1166}$$

We can factor out the common factor of tt', to get:

$$\gamma^2 = \frac{c^2tt'}{tt'(c^2 - v^2)} \tag{1167}$$

$$= \frac{c^2}{(c^2 - v^2)} \tag{1168}$$

We can then simplify by dividing both the numerator and denominator by c^2, which gives us:

$$\gamma^2 = \frac{1}{1 - \frac{v^2}{c^2}} \tag{1169}$$

$$= \frac{1}{1 - \left(\frac{v}{c}\right)^2} \tag{1170}$$

Finally, taking the square root, we have:

$$\gamma = \frac{1}{\sqrt{1 - \left(\frac{v}{c}\right)^2}} \tag{1171}$$

We can use this to derive the Lorentz transformations for x, y, and z, but we need a little more algebra to figure out the Lorentz transform for t. To do this, we first write out the Lorentz transformations from $x \to x'$ and $x' \to x$:

$$x' = \gamma(x - vt) \tag{1172}$$

$$x = \gamma(x' + vt') \tag{1173}$$

We take the second equation to solve for t':

$$t' = \frac{x}{\gamma v} - \frac{x'}{v} \tag{1174}$$

And we can now plug in the first Lorentz transformation equation into x':

$$t' = \frac{x}{\gamma v} - \frac{\gamma(x - vt)}{v} \tag{1175}$$

We can now distribute to find:

$$t' = \gamma\left(\frac{x}{\gamma^2 v} - \frac{x}{v} + t\right) \tag{1176}$$

We can further factor out the first two terms as:

$$t' = \gamma \left(x \left[\frac{1}{\gamma^2 v} - \frac{1}{v} \right] + t \right) \tag{1177}$$

This simplifies to:

$$t' = \gamma \left(x \left[\frac{1 - \gamma^2}{\gamma^2 v} \right] + t \right) \tag{1178}$$

Which then simplifies to:

$$t' = \gamma \left(-\frac{xv}{c^2} + t \right) \tag{1179}$$

Or:

$$t' = \gamma \left(t - \frac{vx}{c^2} \right) \tag{1180}$$

So, we now have the complete set of **Lorentz transformations**, which obey the laws of special relativity:

$$t' = \gamma \left(t - \frac{vx}{c^2} \right) \tag{1181}$$

$$x' = \gamma(x - vt) \tag{1182}$$

$$y' = y \tag{1183}$$

$$z' = z \tag{1184}$$

Where the Lorentz factor γ is approximately 1 at speeds of everyday life, but rises to infinity as you approach c:

```
c = 299792458
v = np.linspace(0, 0.999 * c, 1000)
gamma = 1 / np.sqrt(1 - (v / c) ** 2)
one = np.ones(len(v))

plt.plot(v, gamma, label="Gamma")
plt.plot(v, one, label="y = 1")
plt.legend()
plt.title("Gamma factor as a function of speed (in m/s)")
plt.show()
```

The idea of spacetime In classical physics, time had always been thought of as a steady feature in the background of the universe, something that was universal, and crucially, experienced the same way by everyone. But now, with the Lorentz transforms, it was clear that time was a dimension, like any other, and it couldn't be separated from the dimensions of space. Hence, the new idea of **spacetime** was born - a 4-dimensional space that contained space *and* time.

But how would this new spacetime be described? One way to describe spacetime is by defining a **metric**, which we saw back in tensor calculus. A metric can be used to define how distances are measured in space. For instance, in 2D Euclidean space, we can measure distances with:

$$ds^2 = dx^2 + dy^2 \tag{1185}$$

which is just the Pythagorean theorem. Hermann Minkowski, Einstein's former professor, recognized that this metric would not work in special relativity when applied to spacetime. He instead proposed a different metric, which we today know as the **Minkowski metric**.

We start from the Euclidean metric in three dimensions:

$$ds^2 = dx^2 + dy^2 + dz^2 \tag{1186}$$

Now, if we add a time dimension, and still consider Euclidean space, we have:

$$ds^2 = dt^2 + dx^2 + dy^2 + dz^2 \tag{1187}$$

We want to use the same units for time as well as space (units of meters). Otherwise, we would have incompatible units in our metric. Thus, we add a factor of c^2 to get:

$$ds^2 = dt^2 + dx^2 + dy^2 + dz^2 \tag{1188}$$

Now, note the observation that in special relativity, as an object moves faster, it experiences *less* time, instead of more time, as in Euclidean space. Therefore, the time component of the metric must be negative:

$$ds^2 = -c^2dt^2 + dx^2 + dy^2 + dz^2 \tag{1189}$$

We have arrived at the Minkowski metric.

Consequences of special relativity

Relativity of simultaneity Let's go back to the Lorentz transform for time coordinates:

$$t' = \gamma\left(t - \frac{vx}{c^2}\right) \tag{1190}$$

We can rewrite this equation equivalently in terms of *changes* in time:

$$\Delta t' = \gamma\left(\Delta t - \frac{v\Delta x}{c^2}\right) \tag{1191}$$

From this equation, it is clear that two events with $\Delta t = 0$ - happening simultaneously in one frame - do not necessarily imply that $\Delta t' = 0$ - that is, they are happening simultaneously in the other frame. In fact, the actual case is that:

$$\Delta t = 0 \Rightarrow \Delta t' = -\frac{\gamma v \Delta x}{c^2} \tag{1192}$$

Which implies that events that are simultaneous in frame S are separated by a time of $-\frac{\gamma v \Delta x}{c^2}$ in frame S'. This is the **relativity of simultaneity**.

Time dilation Again, we return to the equation:

$$\Delta t' = \gamma\left(\Delta t - \frac{v\Delta x}{c^2}\right) \tag{1193}$$

In this case, suppose we have two clocks, one at rest aboard a moving spaceship (frame S'), and one at rest on Earth (frame S). Because both clocks are not moving, we can say that $\Delta x = 0$. But this leads us to find something strange:

$$\Delta t' = \gamma \Delta t \tag{1194}$$

That is, more time passes on Earth than aboard the spaceship during the same measured time interval. Or put it another way, clocks tick more slowly when placed in a moving reference frame. For instance, a 3-second time interval for a spaceship going at 80% light speed would be measured as 7-seconds by the earthbound clock. In practice, this effect doesn't show up until a reference frame is moving at more than 50% of the speed of light, but in interstellar spaceflight, the effects can be dramatic - at 99.999% of the speed of light, one year aboard a spacecraft would be 223 years on Earth! In fact, this is one mode of time travel into the future - passengers aboard a very fast craft would experience little time, while a lot more time passes for stationary observers, allowing passengers to seemingly magically travel into the future.

Length contraction We first take the x-coordinate Lorentz transform equation, and we write it in terms of the change in x:

$$\Delta x' = \gamma(\Delta x - v\Delta t) \tag{1195}$$

We set $\Delta t = 0$ as we want a snapshot of one moment in time - therefore, we have:

$$\Delta x' = \gamma \Delta x \tag{1196}$$

We can solve for Δx with:

$$\Delta x = \frac{\Delta x'}{\gamma} \tag{1197}$$

This means in the stationary frame, a moving object would be *contracted* along the direction of motion. This means in the paradoxical case that a meterstick travelling at 90% or more than the speed of light would be able to fit into a barn house less than a meter long.

Relativistic addition of velocities

$$u' = \frac{dx'}{dt'} = \frac{\gamma(dx - vdt)}{\gamma\left(dt - \frac{vdx}{c^2}\right)} = \frac{\frac{dx}{dt} - v}{1 - \left(\frac{v}{c^2}\right)\left(\frac{dx}{dt}\right)} = \frac{u - v}{1 - uv/c^2} \tag{1198}$$

Proper length and proper time: Proper time τ is the time measured by an observer's own clock as the observer moves through spacetime. It is related to the coordinate time t, which is the clock of an external observer, by:

$$\Delta t = \gamma \Delta \tau \tag{1199}$$

Proper length ℓ is the length measured by an observer's own ruler as the observer moves through spacetime. It is related to the coordinate length L, which is the ruler of an external observer, by:

$$\Delta L = \gamma \Delta \ell \tag{1200}$$

Generalizing Newtonian mechanics to special relativity Consider a particle moving along a path through spacetime $x^\mu(\tau)$. The four-velocity of that particle is given by:

$$U^\mu = \frac{dx^\mu}{d\tau} = \gamma \frac{dx^\mu}{dt} = \left(c\frac{dt}{d\tau}, \frac{dx}{d\tau}, \frac{dy}{d\tau}, \frac{dz}{d\tau}\right) \tag{1201}$$

It can also be written as:

$$U^\mu = (c\gamma, \gamma v) \tag{1202}$$

Relativistic four-momentum is given by:

$$P^\mu = mU^\mu = m\gamma v = (mc\gamma, m\gamma v) \tag{1203}$$

Relativistic four-force is given by:

$$F^\mu = \frac{dP^\mu}{d\tau} \tag{1204}$$

Relativitic kinetic energy is given by:

$$K = (\gamma - 1)mc^2 = \gamma mc^2 - mc^2 \tag{1205}$$

Total relativistic energy is given by:

$$E = \gamma mc^2 = K + mc^2 \tag{1206}$$

When the object is stationary, $\gamma = 1$, so the equation simplifies to:

$$E = mc^2 \tag{1207}$$

This provides another way to write relativistic momentum:

$$P^\mu = (E/c, mU^\mu) \tag{1208}$$

General Relativity, Part 1

General relativity is a theory of how gravity works. In General Relativity, gravity is not a force, but rather an effect caused by curved spacetime. This conclusion is based on two fundamental principles:

- The **Equivalence Principle**, which says that gravity is indistinguishable from the effect of an accelerating reference frame
- The **Principle of Covariance**, which says that all laws of physics should be in the same form in all reference frames

The culminating breakthrough of General Relativity is summarized succinctly by the Einstein Field Equations:

$$G_{\alpha\beta} = \frac{8\pi G}{c^4} T_{\alpha\beta} \tag{1209}$$

However, to truly understand what the equation means, we need to go slowly and build our understanding of relativity first. And it often helps to start with the physical intuition underlying relativity - the **equivalence principle**.

The Equivalence Principle Consider an observer inside a closed room. This room is accelerating upwards at a constant rate of 9.81 m/s^2. The observer holds a 1-kilogram ball. What would happen if the observer would drop a ball?

Well, we know that the room is under constant upwards acceleration, so when the observer releases the ball, the floor of the room will travel upwards towards it at 9.81 m/s^2. However, to the observer, who is moving upwards *along with* the floor, it would look like everything is stationary, and the *ball* is the object that is falling down.

If we use Newton's second law of motion, we find that the force experienced by the ball would be given by:

$$\vec{F_b} = m\vec{a} \tag{1210}$$

Thus, the force would be:

$$\vec{F_b} = -9.81N \tag{1211}$$

Now, consider another observer, inside another closed room. This room is placed on the surface of the Earth. The observer inside this second room drops another 1-kilogram ball. What would happen next?

Well, the ball will experience the force of Earth's gravity, causing it to fall downwards as well. If we use Newton's law of universal gravitation, we find that the force experienced by the ball would be given by:

$$\vec{F_b} = -G\frac{M_1 M_2}{r^2} \tag{1212}$$

We can rearrange this equation to the form:

$$\vec{F_b} = \left(\frac{-GM_\oplus}{(r_\oplus)^2}\right) M_2 \tag{1213}$$

Using our closest measurements of the mass of the Earth and its radius, we arrive at the result:

$$\vec{F_b} = -9.81N \tag{1214}$$

Notice that this is the **same result** as our closed room moving upwards through space at 9.81 m/s^2. The effect of gravity and of an accelerated reference frame is the same. But this is just a coincidence, right? Or is it...?

Imagine you were in either the closed room in space or the closed room on Earth, but you weren't told which one. Is there any way you could tell which room it was? No, it would be impossible to perform an experiment to tell the closed room in space from the closed room on Earth.

So, gravity is **indistinguishable** from accelerated reference frames. This is the **equivalence principle**.

Reviewing the spacetime metric To understand General Relativity, we must first be familiar with the idea of **events**.

An event is *anything that happens*. This could be, "a spaceship flew through my window". Yes, that is an event.

We can describe the event by finding the position and time it occured, *relative* to a chosen reference frame:

- E.g. "a spaceship flew through my window at 5 meters left and 6 meters in front of my head, at 2 meters above sea level, at 2:30 pm, on January 15th, 2021"
- We have a x-coordinate (5 meters left of my head), a y-coordinate (6 meters in front of my head), a z-coordinate (2 meters above sea level), and a time coordinate (2:30 pm 1/15/21)

To describe the distance between two events, we use a spacetime metric. This could be the Euclidean metric $\delta_{\alpha\beta}$, the Minkowski metric $\eta_{\alpha\beta}$, or the general metric $g_{\alpha\beta}$. As we've seen before, the Minkowski metric in particular is given by $\eta_{\alpha\beta}$, where α and β represent the (α, β)-th entry of the matrix:

$$\eta_{\alpha\beta} = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \eta_{00} & \eta_{01} & \eta_{02} & \eta_{03} \\ \eta_{10} & \eta_{11} & \eta_{12} & \eta_{13} \\ \eta_{20} & \eta_{21} & \eta_{22} & \eta_{23} \\ \eta_{30} & \eta_{31} & \eta_{32} & \eta_{33} \end{pmatrix} \tag{1215}$$

A note about really confusing metric notation
In General Relativity, it is customary to count time as the 0th dimension rather than the 4th dimension. This is why α and β range from 0 to 3 instead of 1 to 4 (as you ordinarily might expect).

A note about signatures
The Minkowski metric is similar to the Euclidean metric $\delta_{\alpha\beta}$ with one major difference: $\eta_{00} = -1 \neq \delta_{00}$ (see second appendix for why). So, the metric tensor has a *signature* - that of $(-+++)$ along its diagonals. Note that sometimes, physicists will confusingly use a metric signature of $(+---)$, which returns the same distance in spacetime. These two metric signatures are *functionally equivalent* because the metric tensor is symmetric; the only difference is your personal preference. I will stick with $(-+++)$ for consistency here.

Recall that the line element can be written in terms of the product of two infinitesimal displacement vectors multiplied by the metric:

$$ds^2 = \begin{bmatrix} cdt & dx & dy & dz \end{bmatrix} \begin{bmatrix} cdt \\ dx \\ dy \\ dz \end{bmatrix} \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{1216}$$

We can denote one of these infinitesimal displacement vectors dx^α and the other dx^β, so we have:

$$ds^2 = \eta_{\alpha\beta} dx^\alpha dx^\beta \tag{1217}$$

Additionally, since the components of metric tensors can vary as spacetime is curved and distances change, we shouldn't expect that the metric of spacetime will always be $\eta_{\alpha\beta}$; in fact, that would only be true for flat (uncurved) spacetime. So, we replace $\eta_{\alpha\beta}$ with the more general form of the metric tensor $g_{\alpha\beta}$, which applies to all spacetime metrics. We finally arrive at the general metric tensor:

$$ds^2 = g_{\alpha\beta}dx^\alpha dx^\beta \tag{1218}$$

This is the most common form of the spacetime metric you will see, and it is the form we will use going forward.

Spacelike, Timelike, and Lightlike Intervals When finding the spacetime interval between two events, we can describe the interval using one of three terms:

- If the spacetime interval is **positive**, it is spacelike: the two events are separated by space
- If the spacetime interval is **negative**, it is timelike: the two events are separated by time
- If the spacetime interval is **zero**, it is lightlike: a beam of light could travel directly from one event to the other

The Einstein Summation Convention Previously, we have already been introduced to index notation - for example, we saw that position could be represented by $x^\mu = x, y, z$, and that an equation such as $v^\mu = \frac{dx^\mu}{dt}$ is actually a system of three equations, one each for x, y, and z. W've also seen that we can generally let the letters we use for tensor indices to be whatever we want, and the equations will still be consistent. For example, $g_{\alpha\beta} = g_{ij} = g_{\alpha\gamma} = g_{\mu\gamma}$. There is a difference between these two forms of indices, however.

The first type, where we have a single index, is called a **free index**. Free indices result in a different equation for each coodinate - for instance, given $F^\mu = ma^\mu$, then we have the system of equations:

$$\begin{cases} F^x = ma^x \\ F^y = ma^y \\ F^z = ma^z \end{cases} \tag{1219}$$

The second type, where we have an index that repeats once as a lower index and once as an upper index in a term, is used to stand for summation. For example, recall the multivariable chain rule:

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial t} + \frac{\partial f}{\partial z}\frac{\partial z}{\partial t} \tag{1220}$$

We can rewrite this with summation:

$$\frac{\partial f}{\partial t} = \sum_{i=1}^{3} \frac{\partial f}{\partial x_i}\frac{\partial x_i}{\partial t} \tag{1221}$$

Note that we have an index i that appears in the lower index and one that appears in the upper index, so by the Einstein summation convention, we can get rid of the summation sign:

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x_i}\frac{\partial x_i}{\partial t} \tag{1222}$$

And because the index is used for summation, it makes no difference if we change i to j or k or u, the equation will mean the same thing:

$$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial x_j}\frac{\partial x_j}{\partial t} \tag{1223}$$

This works the same way if are working with tensors. Consider the following tensor equation:

$$K_i = a_{ij}b^j \tag{1224}$$

Notice that j appears once as a top index, and once as a bottom index. Thus, j must be a summation index - in GR we call these **dummy indices**. In contrast, i and k don't appear twice on the top and bottom, so are free indices, representing a system of equations. Therefore, if we were to expand the dummy summation index, where j goes from 1 - 3, we'll get:

$$K_i = a_{i1}b^1 + a_{i2}b^2 + a_{i3}b^3 \tag{1225}$$

And we let $x = 1, y = 2, z = 3$, then:

$$K_i = a_{ix}b^x + a_{iy}b^y + a_{iz}b^z \tag{1226}$$

In comparison, remember that i is a free index that expands into a system of equations. So, using $i = (x, y, z)$, we have:

$$\begin{cases} K_x = a_{xx}b^x + a_{xy}b^y + a_{xz}b^z \\ K_y = a_{yx}b^x + a_{yy}b^y + a_{yz}b^z \\ K_z = a_{zx}b^x + a_{zy}b^y + a_{zz}b^z \end{cases} \tag{1227}$$

In general, dummy indices can be changed at will, but free indices cannot. This is because changing a dummy index is just changing the index you use for the summation, which is totally arbitrary, but changing a free index would change the system of equations into a completely different system of equations.

The use of the Einstein summation convention allows the equations of General Relativity to be written very compactly. For instance, take the definition of the Ricci tensor, given by:

$$R_{ij} = \frac{\partial \Gamma^k_{ij}}{\partial x^k} - \frac{\partial \Gamma^k_{ik}}{\partial x^j} + \Gamma^k_{ij}\Gamma^m_{km} - \Gamma^k_{im}\Gamma^m_{jk} \tag{1228}$$

Observe that only i and j are free indices - the other two indices m and k appear both on upper and on lower indices, making them summation indices. If we were to fully write out just the summation of the Ricci tensor, where m and k both sum from 0 to 3, and we assume $0 = t$, $1 = x$, $2 = y$, $3 = z$, we'd get:

$$R_{ij} = \frac{\partial \Gamma^t_{ij}}{\partial x^t} - \frac{\partial \Gamma^t_{it}}{\partial x^j} + \Gamma^t_{ij}\Gamma^t_{tt} - \Gamma^t_{it}\Gamma^t_{jt} + \frac{\partial \Gamma^x_{ij}}{\partial x^x} - \frac{\partial \Gamma^x_{ix}}{\partial x^j} + \Gamma^x_{ij}\Gamma^t_{xt} - \Gamma^x_{it}\Gamma^t_{jx} \tag{1229}$$

$$+ \frac{\partial \Gamma^y_{ij}}{\partial x^y} - \frac{\partial \Gamma^y_{iy}}{\partial x^j} + \Gamma^y_{ij}\Gamma^t_{yt} - \Gamma^y_{it}\Gamma^t_{jy} + \frac{\partial \Gamma^z_{ij}}{\partial x^z} - \frac{\partial \Gamma^z_{iz}}{\partial x^j} + \Gamma^z_{ij}\Gamma^t_{zt} - \Gamma^z_{it}\Gamma^t_{jz} \tag{1230}$$

$$+ \frac{\partial \Gamma^t_{ij}}{\partial x^t} - \frac{\partial \Gamma^t_{it}}{\partial x^j} + \Gamma^t_{ij}\Gamma^x_{tx} - \Gamma^t_{ix}\Gamma^x_{jt} + \frac{\partial \Gamma^x_{ij}}{\partial x^x} - \frac{\partial \Gamma^x_{ix}}{\partial x^j} + \Gamma^x_{ij}\Gamma^x_{xx} - \Gamma^x_{ix}\Gamma^x_{jx} \tag{1231}$$

$$+ \frac{\partial \Gamma^y_{ij}}{\partial x^y} - \frac{\partial \Gamma^y_{iy}}{\partial x^j} + \Gamma^y_{ij}\Gamma^x_{yx} - \Gamma^y_{ix}\Gamma^x_{jy} + \frac{\partial \Gamma^z_{ij}}{\partial x^z} - \frac{\partial \Gamma^z_{iz}}{\partial x^j} + \Gamma^z_{ij}\Gamma^x_{zx} - \Gamma^z_{ix}\Gamma^x_{jz} \tag{1232}$$

$$+ \frac{\partial \Gamma^t_{ij}}{\partial x^t} - \frac{\partial \Gamma^t_{it}}{\partial x^j} + \Gamma^t_{ij}\Gamma^y_{ty} - \Gamma^t_{iy}\Gamma^y_{jt} + \frac{\partial \Gamma^x_{ij}}{\partial x^x} - \frac{\partial \Gamma^x_{ix}}{\partial x^j} + \Gamma^x_{ij}\Gamma^y_{xy} - \Gamma^x_{iy}\Gamma^y_{jx} \tag{1233}$$

$$+ \frac{\partial \Gamma^y_{ij}}{\partial x^y} - \frac{\partial \Gamma^y_{iy}}{\partial x^j} + \Gamma^y_{ij}\Gamma^y_{yy} - \Gamma^y_{iy}\Gamma^y_{jy} + \frac{\partial \Gamma^z_{ij}}{\partial x^z} - \frac{\partial \Gamma^z_{iz}}{\partial x^j} + \Gamma^z_{ij}\Gamma^y_{zy} - \Gamma^z_{iy}\Gamma^y_{jz} \tag{1234}$$

$$+ \frac{\partial \Gamma^t_{ij}}{\partial x^t} - \frac{\partial \Gamma^t_{it}}{\partial x^j} + \Gamma^t_{ij}\Gamma^z_{tz} - \Gamma^t_{iz}\Gamma^z_{jt} + \frac{\partial \Gamma^x_{ij}}{\partial x^x} - \frac{\partial \Gamma^x_{ix}}{\partial x^j} + \Gamma^x_{ij}\Gamma^z_{xz} - \Gamma^x_{iz}\Gamma^z_{jx} \tag{1235}$$

$$+ \frac{\partial \Gamma^y_{ij}}{\partial x^y} - \frac{\partial \Gamma^y_{iy}}{\partial x^j} + \Gamma^y_{ij}\Gamma^z_{yz} - \Gamma^y_{iz}\Gamma^z_{jy} + \frac{\partial \Gamma^z_{ij}}{\partial x^z} - \frac{\partial \Gamma^z_{iz}}{\partial x^j} + \Gamma^z_{ij}\Gamma^z_{zz} - \Gamma^z_{iz}\Gamma^z_{jz} \tag{1236}$$

And remember, this is just expanding the summations! This isn't even writing out the system of equations for each free index! You can go and see the full Einstein field equations with each system of equations written out at https://github.com/bnschussler/Fully-Expanded-Einstein-Field-Equations, and it is 22 pages long!

Finally, in General Relativity, partial derivatives are often written in a compact way:

$$\frac{\partial}{\partial x^\mu} \Rightarrow \partial_\mu \tag{1237}$$

And if we have a partial derivative, it is considered a lower index in a tensor:

$$T^\mu{}_\nu = \partial_\nu A^\mu \tag{1238}$$

The Geodesic Equation We know from Newton's first law of motion that an object in motion stays in motion at constant speed - that is, it undergoes no acceleration. In other terms:

$$\frac{d^2x^\alpha}{d\tau^2} = 0 \tag{1239}$$

Where:

$$x^\alpha = x^1, x^2, x^3 ... \, x^n = x, y, z \, ... \, n \tag{1240}$$

This is why, for instance, a ball rolling along an infinitely long hallway will keep going in a path in the same direction - its velocity vector, and thus its directions, stays constant. In nice Euclidean space, we call this path a "straight line" - the effect of going ahead in the same direction forever.

As we know, in Euclidean space, a straight line is the shortest path between two points, which we call a *geodesic*. We might be tempted to phrase Newton's first law to say that "a particle in motion will travel along a straight line". However, in non-Euclidean geometries, a geodesic is not necessarily a straight line. So, we must generalize Newton's first law with modifications: **a particle in motion will move along a geodesic**.

To formulate this law mathematically, we can say that the action along the path $x^k(\lambda)$ between 2 points $A = x^k(0)$ and $B = x^k(1)$ must be minimized. We include the units of mc to get the units for action right, so the full action along the path $x^k(\lambda)$ is:

$$S = -mc \int \sqrt{-ds^2} \tag{1241}$$

About the line element
Here, the line element is negative to get rid of the annoying $-c^2dt^2$ element which would yield an imaginary number if square rooted. Because the distance along the path is the same whether you travel in the forwards direction ($\sqrt{ds^2}$) or in the backwards direction ($\sqrt{-ds^2}$), the result is equivalent.

We can expand this out by writing $ds^2 = g_{ij}dx^i dx^j$, so:

$$S = -mc \int \sqrt{-g_{ij}dx^i dx^j} \tag{1242}$$

Now, to actually be able to solve, we need to write the integrand in terms of our path parameter λ. To do this, we can divide dx^i and dx^j both by $d\lambda$, then, to keep the integrand the same, multiply by $d\lambda \cdot d\lambda = d\lambda^2$:

$$S = -mc \int \sqrt{-g_{ij}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda}d\lambda^2} \tag{1243}$$

We can then take out the $d\lambda$ from the square root to have:

$$S = -mc \int \sqrt{-g_{ij}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda}}d\lambda \tag{1244}$$

Knowing that the integrand of the action is the Lagrangian, we can write:

$$\mathcal{L} = \sqrt{-g_{ij}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda}} \tag{1245}$$

We can more specifically write it out that that metric g_{ij} is a function of x^k, so:

$$\mathcal{L} = \sqrt{-g_{ij}(x^k)\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda}} \tag{1246}$$

We can apply the familiar Euler-Lagrange equations to our Lagrangian, as we've done before, to find the equations of motion for the particle traveling along the path:

$$\frac{d}{d\lambda}\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{\partial \mathcal{L}}{\partial x^k} \tag{1247}$$

Let's first take the derivative with respect to x^k. We use the chain rule and the fact that $\frac{\partial}{\partial x}\sqrt{u} = \frac{1}{2}u^{-\frac{1}{2}}\frac{\partial u}{\partial x}$. In our Lagrangian, the only part that actually depends on x^k is g_{ij}, so the rest of the Lagrangian can be thought of as a constant. So we have:

$$\frac{\partial \mathcal{L}}{\partial x^k} = -\frac{1}{2}\left(g_{ij}(x^k)\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda}\right)^{-1/2}\frac{\partial g_{ij}}{\partial x^k}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} \tag{1248}$$

We can simplify this by noticing that the Lagrangian itself appears in its derivative, so we can write:

$$\frac{\partial \mathcal{L}}{\partial x^k} = -\frac{1}{2}\mathcal{L}^{-1}\frac{\partial g_{ij}}{\partial x^k}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} \tag{1249}$$

Which simplifies to:

$$\frac{\partial \mathcal{L}}{\partial x^k} = -\frac{1}{2\mathcal{L}}\frac{\partial g_{ij}}{\partial x^k}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} \tag{1250}$$

Now, let's take the derivative with respect to $\dot{x}^k$. Here, using the same Lagrangian substitution technique and square root derivative, we find that:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = -\frac{1}{2\mathcal{L}}\left(-\frac{\partial}{\partial \dot{x}^k}g_{ij}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda}\right) \tag{1251}$$

Remember that g_{ij} isn't dependent on $\dot{x}^k$, so we can simplify factor it out for now:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{2\mathcal{L}}g_{ij}\left(\frac{\partial}{\partial \dot{x}^k}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda}\right) \tag{1252}$$

To differentiate the part inside brackets of the previous expression, we use the product rule, namely $\frac{\partial}{\partial x}fg = \frac{\partial}{\partial x}f\cdot g + \frac{\partial}{\partial x}g\cdot f$, to get:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = -\frac{1}{2\mathcal{L}}g_{ij}\left(\frac{\partial \dot{x}^i}{\partial \dot{x}^k}\frac{dx^j}{d\lambda} + \frac{\partial \dot{x}^j}{\partial \dot{x}^k}\frac{dx^i}{d\lambda}\right) \tag{1253}$$

Now, we will use the Kronecker delta-partial derivative rule:

$$\frac{\partial \dot{x}^i}{\partial \dot{x}^k} = \delta^i{}_k \tag{1254}$$

This simplifies the expression to:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{2\mathcal{L}}g_{ij}\left(\delta^i{}_k\frac{dx^j}{d\lambda} + \delta^j{}_k\frac{dx^i}{d\lambda}\right) \tag{1255}$$

Finally, we distribute the expression with the metric g_{ij}:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{2\mathcal{L}}\left(g_{ij}\delta^i{}_k\frac{dx^j}{d\lambda} + g_{ij}\delta^j{}_k\frac{dx^i}{d\lambda}\right) \tag{1256}$$

Remembering that the Kronecker delta can be used to relabel indices:

$$g_{\alpha\beta}\delta^\alpha{}_\mu = g_{\beta\mu} \tag{1257}$$

We rewrite the expression as:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{2\mathcal{L}} \left(g_{jk} \frac{dx^j}{d\lambda} + g_{ik} \frac{dx^i}{d\lambda} \right) \tag{1258}$$

Finally, let's remember the Einstein summation convention, which tell us that dummy indices can be changed to whatever indices we want, because they are just summation indices. Here, since i and j appear both as lower indices and as upper indices, they are dummy indices. That means:

$$g_{jk} \frac{dx^j}{d\lambda} = g_{ik} \frac{dx^i}{d\lambda} \tag{1259}$$

So to simplify the expression, we can replace all the j's with i's (as we can do with dummy indices), to obtain:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{2\mathcal{L}} \left(g_{ik} \frac{dx^i}{d\lambda} + g_{ik} \frac{dx^i}{d\lambda} \right) \tag{1260}$$

Which simplifies to:

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{2\mathcal{L}} \left(2 g_{ik} \frac{dx^i}{d\lambda} \right) \tag{1261}$$

$$\frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{\mathcal{L}} g_{ik} \frac{dx^i}{d\lambda} \tag{1262}$$

Now we simply need to differentiate our previous result with respect to λ:

$$\frac{d}{d\lambda} \frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{d}{d\lambda} \left(\frac{1}{\mathcal{L}} g_{ik} \frac{dx^i}{d\lambda} \right) = \frac{1}{\mathcal{L}} \frac{d}{d\lambda} \left(g_{ik} \frac{dx^i}{d\lambda} \right) \tag{1263}$$

Here, we use the product rule again with the terms in the brackets, which give us:

$$\frac{d}{d\lambda} \frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{\mathcal{L}} \left(\frac{\partial g_{ik}}{\partial \lambda} \frac{dx^i}{d\lambda} + \frac{d^2 x^i}{d\lambda^2} g_{ik} \right) \tag{1264}$$

Now, we know that technically $g_{ij} = g_{ij}(x^k(\lambda))$, so we can use the chain rule to write:

$$\frac{\partial g_{ik}}{\partial \lambda} = \frac{\partial g_{ik}}{\partial x^a} \frac{dx^a}{d\lambda} \tag{1265}$$

Here, a can be any index that isn't already one of the free indices (the free indices are i and k here). We'll choose a, but really it can be anything.

So we have:

$$\frac{d}{d\lambda} \frac{\partial \mathcal{L}}{\partial \dot{x}^k} = \frac{1}{\mathcal{L}} \left(\frac{\partial g_{ik}}{\partial x^a} \frac{dx^a}{d\lambda} \frac{dx^i}{d\lambda} + \frac{d^2 x^i}{d\lambda^2} g_{ik} \right) \tag{1266}$$

Equating the two sides of the Euler-Lagrange equation, we have:

$$-\frac{1}{2\mathcal{L}} \frac{\partial g_{ij}}{\partial x^k} \frac{dx^i}{d\lambda} \frac{dx^j}{d\lambda} = \frac{1}{\mathcal{L}} \left(\frac{\partial g_{ik}}{\partial x^a} \frac{dx^a}{d\lambda} \frac{dx^i}{d\lambda} + \frac{d^2 x^i}{d\lambda^2} g_{ik} \right) \tag{1267}$$

We can move the left-hand side to the right to get:

$$\frac{1}{\mathcal{L}} \left(\frac{\partial g_{ik}}{\partial x^a} \frac{dx^a}{d\lambda} \frac{dx^i}{d\lambda} + \frac{d^2 x^i}{d\lambda^2} g_{ik} \right) - \frac{1}{2\mathcal{L}} \frac{\partial g_{ij}}{\partial x^k} \frac{dx^i}{d\lambda} \frac{dx^j}{d\lambda} = 0 \tag{1268}$$

And we can factor out the common factor of the Lagrangian and get rid of it by dividing it from both sides of the equation (remember zero divided by anything is still zero):

$$\left(\frac{\partial g_{ik}}{\partial x^a} \frac{dx^a}{d\lambda} \frac{dx^i}{d\lambda} + \frac{d^2 x^i}{d\lambda^2} g_{ik} \right) - \frac{1}{2} \frac{\partial g_{ij}}{\partial x^k} \frac{dx^i}{d\lambda} \frac{dx^j}{d\lambda} = 0 \tag{1269}$$

As well as rearranging the terms to put the second derivative in front:

$$\left(\frac{d^2x^i}{d\lambda^2}g_{ik} + \frac{\partial g_{ik}}{\partial x^a}\frac{dx^a}{d\lambda}\frac{dx^i}{d\lambda}\right) - \frac{1}{2}\frac{\partial g_{ij}}{\partial x^k}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} = 0 \tag{1270}$$

Now, we can remove the brackets:

$$\frac{d^2x^i}{d\lambda^2}g_{ik} + \frac{\partial g_{ik}}{\partial x^a}\frac{dx^a}{d\lambda}\frac{dx^i}{d\lambda} - \frac{1}{2}\frac{\partial g_{ij}}{\partial x^k}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} = 0 \tag{1271}$$

Notice that this equation has **three** dummy indices - i (which appears both in d^2x^i as upper index and g_{ik} as lower index), a (which appears both in the lower part of a partial derivative and upper part of another derivative dx^a), and j (which appears as a lower index in g_{ij} and an upper index in dx^j). Remember this! It'll be very important later!

Let's take a close look at the middle term:

$$\frac{\partial g_{ik}}{\partial x^a}\frac{dx^a}{d\lambda}\frac{dx^i}{d\lambda} \tag{1272}$$

It can be seen that it can be alternatively written as:

$$\frac{1}{2}\left(\frac{\partial g_{ik}}{\partial x^a} + \frac{\partial g_{ik}}{\partial x^a}\right)\frac{dx^a}{d\lambda}\frac{dx^i}{d\lambda} \tag{1273}$$

Let's distribute this:

$$\frac{1}{2}\frac{\partial g_{ik}}{\partial x^a}\frac{dx^a}{d\lambda}\frac{dx^i}{d\lambda} + \frac{1}{2}\frac{\partial g_{ik}}{\partial x^a}\frac{dx^a}{d\lambda}\frac{dx^i}{d\lambda} \tag{1274}$$

Note here that again, a and i are dummy indices - a appears on the lower partial derivative and upper partial derivative terms, and i appears in the lower g_{ik} and upper dx^i term. So let's do two index substitutions which will make the equation so much easier to solve. First, note that the equation we extracted this expression from had three dummy indices - let's reduce it to just two by swapping a with j:

$$\frac{1}{2}\frac{\partial g_{ik}}{\partial x^j}\frac{dx^j}{d\lambda}\frac{dx^i}{d\lambda} + \frac{1}{2}\frac{\partial g_{ik}}{\partial x^j}\frac{dx^j}{d\lambda}\frac{dx^i}{d\lambda} \tag{1275}$$

Second, in a somewhat bizarre move, we will do a twin substitution $i \to j$ and $j \to i$ in the second term, while leaving the first term alone:

$$\frac{1}{2}\frac{\partial g_{ik}}{\partial x^j}\frac{dx^j}{d\lambda}\frac{dx^i}{d\lambda} + \frac{1}{2}\frac{\partial g_{jk}}{\partial x^i}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} \tag{1276}$$

Third, we will switch the order of the ordinary derivative terms in both terms, which we can do, as they are commutative products:

$$\frac{1}{2}\frac{\partial g_{ik}}{\partial x^j}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} + \frac{1}{2}\frac{\partial g_{jk}}{\partial x^i}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} \tag{1277}$$

Plugging our modified but technically identical version of the middle term back into the equation, we have:

$$\frac{d^2x^i}{d\lambda^2}g_{ik} + \frac{1}{2}\frac{\partial g_{ik}}{\partial x^j}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} + \frac{1}{2}\frac{\partial g_{jk}}{\partial x^i}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} - \frac{1}{2}\frac{\partial g_{ij}}{\partial x^k}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} = 0 \tag{1278}$$

We now have common factors, which we can combine to form:

$$\frac{d^2x^i}{d\lambda^2}g_{ik} + \frac{1}{2}\left(\frac{\partial g_{ik}}{\partial x^j} + \frac{\partial g_{kj}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^k}\right)\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} = 0 \tag{1279}$$

Now we want to get rid of the g_{ik} term. To do this, we will multiply both sides of the equation with the inverse metric $g^{\mu k}$. This gives:

$$g^{\mu k}\frac{d^2x^i}{d\lambda^2}g_{ik} + \frac{1}{2}g^{\mu k}\left(\frac{\partial g_{ik}}{\partial x^j} + \frac{\partial g_{kj}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^k}\right)\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} = 0 \tag{1280}$$

Let's focus on the first term:

$$g^{\mu k}\frac{d^2x^i}{d\lambda^2}g_{ik} \tag{1281}$$

We'll make the tensor contractions easier to see by moving the terms around:

$$g^{\mu k}g_{ik}\frac{d^2x^i}{d\lambda^2} \tag{1282}$$

Recall that $g^{\mu k}g_{ik} = \delta^{\mu}{}_{i}$ by the rules of tensor contraction, so we have:

$$\delta^{\mu}{}_{i}\frac{d^2x^i}{d\lambda^2} \tag{1283}$$

Now, the upper index on the derivative x^i and the lower index i of the Kronecker delta cancel to relabel i to μ, as we saw before in tensor calculus:

$$\delta^{\mu}{}_{i}\frac{d^2x^i}{d\lambda^2} = \frac{d^2x^{\mu}}{d\lambda^2} \tag{1284}$$

So our full equation is:

$$\frac{d^2x^{\mu}}{d\lambda^2} + \frac{1}{2}g^{\mu k}\left(\frac{\partial g_{ik}}{\partial x^j} + \frac{\partial g_{kj}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^k}\right)\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} = 0 \tag{1285}$$

We can further simplify this equation by extracting out the partial derivatives terms in the middle as the **Christoffel symbols**:

$$\Gamma^{\mu}_{ij} = \frac{1}{2}g^{\mu k}\left(\frac{\partial g_{ik}}{\partial x^j} + \frac{\partial g_{kj}}{\partial x^i} - \frac{\partial g_{ij}}{\partial x^k}\right) \tag{1286}$$

Where (this is clearer if we expand out the partial derivatives term by term, but it can be seen by just glancing at the equation too) the free indices are μ, j, and i (which is why they appear on the Christoffel symbol itself), while the dummy index k is used for summation (which is why it's only present in the partial derivatives).

So we finally arrive at the **geodesic equation**:

$$\frac{d^2x^{\mu}}{d\lambda^2} + \Gamma^{\mu}_{ij}\frac{dx^i}{d\lambda}\frac{dx^j}{d\lambda} = 0 \tag{1287}$$

Note that for particles with mass, the paramter λ is interpreted to be the proper time τ, but really λ can be any invariant parameter along a particle's trajectory. If we set $\lambda = \tau$, then the equations become:

$$\frac{d^2x^{\mu}}{d\tau^2} + \Gamma^{\mu}_{ij}\frac{dx^i}{d\tau}\frac{dx^j}{d\tau} = 0 \tag{1288}$$

Any path that obeys the geodesic equation in spacetime is a geodesic. Since spacetime can be curved, these geodesics are not straight lines. The curvature of spacetime - what we experience as gravity - causes distances to change. This causes the metric to change, which in turn affects the paths of particles.

General Relativity, Part 2

Previously, we built up the idea of acceleration as being indistinguishable from gravitational force, and found the geodesic equation, which governs how objects move in curved spacetime. But to truly accurately describe relativity, we also need a mathematical description of what curvature in spacetime is. This will be what we'll explore in this section.

The Covariant Derivative As we've seen by this point, the laws of physics are primarily written in partial differential equations, and so it would be natural to think that General Relativity can be characterized by partial differential equations too. The issue is, consider a vector $\vec{V} = V^a e_a$. If we were to take its partial derivative, we'd have (using the product rule):

$$\frac{\partial \vec{V}}{\partial x^b} = \frac{\partial V^a}{\partial x^b} e_a + \frac{\partial e_a}{\partial x^b} V^a \tag{1289}$$

But remember that tensors should transform like tensors, where each component is at most a partial derivative multiplied by the original tensor? The additional $\frac{\partial e_a}{\partial x^b} V^a$ term means that the regular partial derivative doesn't transform like a tensor. So instead of partial derivatives, we need to define a new type of derivative, the **covariant derivative**, which compensates for the additional term in the partial derivative. The covariant derivative takes the form:

$$\nabla_b V^a = \frac{\partial V^a}{\partial x^b} + V^k \Gamma^a_{kb} \tag{1290}$$

for vectors (those with an upper index), and the form:

$$\nabla_b V_a = \frac{\partial V_a}{\partial x^b} - V_k \Gamma^k_{ab} \tag{1291}$$

for covectors (those with a lower index). To take the covariant derivative of tensors formed from both vectors and covectors, such as the metric tensor $g_{\mu\nu}$, we add a term for each upper index the tensor has and subtract a term for each lower index the tensor has (you'll see how this works in just a moment). For example, for the metric tensor, we first write out the covariant derivative as a partial derivative, plus an unknown term:

$$\nabla_b g_{\mu\nu} = \frac{\partial g_{\mu\nu}}{\partial x^b} + \dots \tag{1292}$$

Then, we notice that the metric tensor has two lower indices, so we need two correction terms. The first correction term is for the index μ, and the second correction term is for the index ν. To emphasize which index each correction term is for, there is a little hat on that index:

$$\nabla_b g_{\mu\nu} = \frac{\partial g_{\mu\nu}}{\partial x^b} - g_{\hat{\mu}\nu} A - g_{\mu\hat{\nu}} B \tag{1293}$$

Now comes the slightly bizarre part. We're going to replace whichever index we're interested in (the one with a hat on) with a dummy index α. This is so that the rules of tensor algebra work out such that the covariant derivative transforms like a tensor. So:

$$\frac{\partial g_{\mu\nu}}{\partial x^b} - g_{\alpha\nu} A - g_{\mu\alpha} B \tag{1294}$$

To figure out the correct index convention for A and B, we use the rule that we multiply $\Gamma^\alpha_{\gamma b}$ for each lower index correction term, and multiply $\Gamma^\gamma_{\alpha b}$ for each upper index correction term. Here:

- α is the dummy index we're using
- γ is the index of the term we're interested in
- b is the index we take the covariant derivative with respect to.

So $A = \Gamma^\alpha_{\mu b}$ and $B = \Gamma^\alpha_{\nu b}$. Thus we have:

$$\nabla_b g_{\mu\nu} = \frac{\partial g_{\mu\nu}}{\partial x^b} - g_{\alpha\nu} \Gamma^\alpha_{\mu b} - g_{\mu\alpha} \Gamma^\alpha_{\nu b} \tag{1295}$$

Also, it should be noted that "covariant derivative" is a bit of a misnomer - here, the definition of the word "covariant" pre-dates the idea of contra- and covariant tensors (tensors with upper/lower indices), and referes to the earlier definition of "invariant". Thus, the covariant derivative is really just a fancy way of saying a derivative of a tensor that is invariant of the coordinates used.

Lastly, the covariant derivative of a field with respect to the same index as the index of the field is equal to simply the divergence:

$$\nabla_b V^b = \nabla \cdot \vec{V} \tag{1296}$$

The Riemann tensor What can we use to measure the curvature of spacetime? We already know that with the covariant derivative, we can take fully-invariant derivatives in spacetime. But if spacetime is to be curved, then if we take a derivative of a vector along direction μ, then another along direction ν, we'd expect to get a different result than if we were to take a derivative along direction ν, then along direction μ. We can qualitatively describe this as:

$$\nabla_\mu \nabla_\nu V^\alpha \neq \nabla_\nu \nabla_\mu V^\alpha \tag{1297}$$

The difference between the two sets of derivatives is going to tell us how much the curvature of spacetime varies between the two points. So we simply need to compute:

$$\nabla_\mu \nabla_\nu V^\alpha - \nabla_\nu \nabla_\mu V^\alpha \tag{1298}$$

First, we expand out the covariant derivatives:

$$\nabla_\mu (\nabla_\nu V^\alpha) - \nabla_\nu (\nabla_\mu V^\alpha) \tag{1299}$$

$$\nabla_\mu (\partial_\nu V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\nu}) - \nabla_\nu (\partial_\mu V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\mu}) \tag{1300}$$

Let's take this step by step, and we'll start with the first covariant derivative:

$$\nabla_\mu (\partial_\nu V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\nu}) \tag{1301}$$

Notice that because σ is a dummy index and contracts, we can rewrite the inside of the parentheses as another tensor:

$$C^\alpha_\nu = \partial_\nu V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\nu} \tag{1302}$$

If we take the covariant derivative of C^α_ν, we know that we have a partial derivative, plus several other correction terms::

$$\nabla_\mu C^\alpha_\nu = \partial_\mu C^\alpha_\nu + \dots \tag{1303}$$

We can write out the remaining correction terms as the tensor multiplied by several coefficients (add correction term if upper index, subtract correction term if lower index), with hats indicating which terms we're interested in:

$$\nabla_\mu C^\alpha_\nu = \partial_\mu C^\alpha_\nu + C^{\hat{\alpha}}_\nu A - C^\alpha_{\hat{\nu}} B \tag{1304}$$

We replace the hatted indices with our dummy index λ (we chose a new variable so as not to cause confusion with the existing variables):

$$\nabla_\mu C^\alpha_\nu = \partial_\mu C^\alpha_\nu + C^\lambda_\nu A - C^\alpha_\lambda B \tag{1305}$$

Using the correction term coefficient rule described earlier for the Christoffel symbols, we recall that:

- For an upper index coefficient term we have the dummy index on the bottom and the interested index on the top

- For a lower index coefficient term we have the dummy index on top and the interested index on the bottom
- The rightmost lower term on the coefficient is always the index we're taking the covariant derivative with respect to (in our case μ)

So we can figure out that $A = \Gamma^{\alpha}_{\lambda\mu}$, and $B = \Gamma^{\lambda}_{\nu\mu}$. So:

$$\nabla_\mu C^\alpha_\nu = \partial_\mu C^\alpha_\nu + C^\lambda_\nu \Gamma^\alpha_{\lambda\mu} - C^\alpha_\lambda \Gamma^\lambda_{\nu\mu} \tag{1306}$$

But recall that:

$$C^\alpha_\nu = \partial_\nu V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\nu} \tag{1307}$$

From which we can perform index substitutions on every index to get:

$$C^\lambda_\nu = \partial_\nu V^\lambda + V^\sigma \Gamma^\lambda_{\sigma\nu} \tag{1308}$$

$$C^\alpha_\lambda = \partial_\lambda V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\lambda} \tag{1309}$$

Note
The last two were obtained by changing the indices $\alpha \to \lambda$ and $\nu \to \lambda$. Shouldn't this be illegal in tensor algebra, where we're not supposed to swap free indices in the same equation? There is a nuance here - we can swap free indices **only** if we subtitute each and every index with a corresponding different index. That, is, if you have an equation $x^i = g^{ij} b_j$, you can't simply say "I want to swap $j \to i$ and make the equation $x^i = g^{ii} b_i$". Here, you're selectively substituting $j \to i$ without making a substitution for i, so the equation is wrong! But you can say that "I'll rewrite the equation using different indices, substituting $i \to a$ and $j \to b$, so I have $x^a = g^{ab} b_b$". Since we swapped *every* index with a *unique* different index, this is acceptable.

We can therefore rewrite the covariant derivative of C^α_ν as:

$$\nabla_\mu C^\alpha_\nu = \partial_\mu(\partial_\nu V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\nu}) + (\partial_\nu V^\lambda + V^\sigma \Gamma^\lambda_{\sigma\nu})\Gamma^\alpha_{\lambda\mu} - (\partial_\lambda V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\lambda})\Gamma^\lambda_{\nu\mu} \tag{1310}$$

Be careful! The second and third terms are just products, but the first term is a derivative, so we have to use the product rule to expand - $\partial_\mu(\partial_\nu V^\alpha + V^\sigma \Gamma^\alpha_{\sigma\nu}) = \partial_\mu \partial_\nu V^\alpha + \partial_\mu(V^\sigma \Gamma^\alpha_{\sigma\nu})$. Using that, and expanding the rest of the terms out, we get:

$$\nabla_\mu \nabla_\nu V^\alpha = \nabla_\mu C^\alpha_\nu = \partial_\mu \partial_\nu V^\alpha + \partial_\mu(V^\sigma \Gamma^\alpha_{\sigma\nu}) + \Gamma^\alpha_{\lambda\mu} \partial_\nu V^\lambda + V^\sigma \Gamma^\lambda_{\sigma\nu} \Gamma^\alpha_{\lambda\mu} - \partial_\lambda V^\alpha \Gamma^\lambda_{\nu\mu} - V^\sigma \Gamma^\alpha_{\sigma\lambda} \Gamma^\lambda_{\nu\mu} \tag{1311}$$

Phew! We're almost there, just hang in there for the remaining derivation. Good news! Things are going to look simpler from this point on. We've already solved the left double covariant derivative, $\nabla_\mu \nabla_\nu V^\alpha$. The right double covariant derivative is just the left double covariant derivative with an index swap $\mu \leftrightarrow \nu$ (that means every time we see a μ, we replace it with a ν, and every time we see a ν, we replace it with a μ). So it is:

$$\nabla_\nu \nabla_\mu V^\alpha = \partial_\nu \partial_\mu V^\alpha + \partial_\nu(V^\sigma \Gamma^\alpha_{\sigma\mu}) + \Gamma^\alpha_{\lambda\nu} \partial_\mu V^\lambda + V^\sigma \Gamma^\lambda_{\sigma\mu} \Gamma^\alpha_{\lambda\nu} - \partial_\lambda V^\alpha \Gamma^\lambda_{\mu\nu} - V^\sigma \Gamma^\alpha_{\sigma\lambda} \Gamma^\lambda_{\mu\nu} \tag{1312}$$

Now is the glorious part - when we subtract one from the other, the terms cancel each other out. Because second partial derivatives are equal no matter what order you take them, $\partial_\mu \partial_\nu = \partial_\nu \partial_\mu$, so those cancel. The last two terms are identical for both (given the symmetry of the Christoffel symbols, so they cancel as well. We're left with:

$$\nabla_\mu \nabla_\nu V^\alpha - \nabla_\nu \nabla_\mu V^\alpha = (\partial_\mu(V^\sigma \Gamma^\alpha_{\sigma\nu}) + \Gamma^\alpha_{\lambda\mu} \partial_\nu V^\lambda - V^\sigma \Gamma^\lambda_{\sigma\nu} \Gamma^\alpha_{\lambda\mu}) - (\partial_\nu(V^\sigma \Gamma^\alpha_{\sigma\mu}) + \Gamma^\alpha_{\lambda\nu} \partial_\mu V^\lambda - V^\sigma \Gamma^\lambda_{\sigma\mu} \Gamma^\alpha_{\lambda\nu}) \tag{1313}$$

Notice a third less obvious cancellation where $\partial_\mu(V^\sigma\Gamma^\alpha_{\sigma\nu}) = \partial_\mu V^\sigma\Gamma^\alpha_{\sigma\nu} + \partial_\mu\Gamma^\alpha_{\sigma\nu}V^\sigma$ which cancels out $\Gamma^\alpha_{\lambda\nu}\partial_\mu V^\lambda$ on the right (because dummy indices don't matter). This simplifies the expression to:

$$\nabla_\mu\nabla_\nu V^\alpha - \nabla_\nu\nabla_\mu V^\alpha = \partial_\mu\Gamma^\alpha_{\sigma\nu}V^\sigma - \partial_\nu\Gamma^\alpha_{\sigma\mu}V^\sigma + V^\sigma\Gamma^\lambda_{\sigma\nu}\Gamma^\alpha_{\lambda\mu} - V^\sigma\Gamma^\lambda_{\sigma\mu}\Gamma^\alpha_{\lambda\nu} \tag{1314}$$

We can now finally (!!!) factor out the V^σ from the expression, to get:

$$\nabla_\mu\nabla_\nu V^\alpha - \nabla_\nu\nabla_\mu V^\alpha = V^\sigma[\partial_\mu\Gamma^\alpha_{\sigma\nu} - \partial_\nu\Gamma^\alpha_{\sigma\mu} + \Gamma^\lambda_{\sigma\nu}\Gamma^\alpha_{\lambda\mu} - \Gamma^\lambda_{\sigma\mu}\Gamma^\alpha_{\lambda\nu}] \tag{1315}$$

The term in the brackets can be written as a new tensor:

$$R^\alpha_{\sigma\mu\nu} = \partial_\mu\Gamma^\alpha_{\sigma\nu} - \partial_\nu\Gamma^\alpha_{\sigma\mu} + \Gamma^\lambda_{\sigma\nu}\Gamma^\alpha_{\lambda\mu} - \Gamma^\lambda_{\sigma\mu}\Gamma^\alpha_{\lambda\nu} \tag{1316}$$

The is the **Riemann curvature tensor**, and it measures how vectors diverge due to the curvature of space. It is a monster tensor - it has 256 components in 4D space, making it a $4\times4\times4\times4$ matrix.

To make this tensor easier to work with, we often contract it by making the 1st and 3rd indices identical, creating the **Ricci tensor**, which is defined by:

$$R_{\sigma\nu} = R^\mu_{\sigma\mu\nu} = \partial_\mu\Gamma^\mu_{\sigma\nu} - \partial_\nu\Gamma^\mu_{\sigma\mu} + \Gamma^\lambda_{\sigma\nu}\Gamma^\mu_{\lambda\mu} - \Gamma^\lambda_{\sigma\mu}\Gamma^\mu_{\lambda\nu} \tag{1317}$$

We can further contract the Ricci tensor by multiplying it by the inverse metric, giving the **Ricci scalar**:

$$R = g^{\mu\nu}R_{\mu\nu} \tag{1318}$$

We now have all the tensors we need to derive the ultimate equation - the **Einstein Field Equations**.

General Relativity, Part 3

After having explored geodesics, the metric tensor, and the curvature tensors, we are ready to tackle the formidable task of finally deriving Einstein's equations!

Deriving the Einstein Field Equations As with before, we can use the Euler-Lagrange equations and the principle of least action to obtain the Einstein Field Equations.

The action for General Relativity in empty spacetime can be generalized as:

$$S = \kappa \int R\sqrt{-g}\ d^4x \tag{1319}$$

Here, $g = \det(g_{\mu\nu})$, $d^4x = dt\,dx\,dy\,dz$ and κ is simply a proportionality constant. Note that while it describes a vacuum, that spacetime can still be curved. For example, you could say that the spacetime *outside* of a black hole is a vacuum (because there is no matter), but the spacetime would still be curved (because the black hole warps its surrounding spacetime, even if we only include the spacetime around a black hole and not the black hole itself).

The action can be derived from one of two ways. It can be shown to be correct through dimensional analysis - the units on the left and right side of the equation match up. However, there is also a more intuitive way to illustrate this.

The action must be composed of scalar-valued functions (or scalars), as it is an integral over all spacetime, and multidimensional integrals can only take scalar-valued functions or scalars to integrate over (see for yourself that this must be true). But it must also include information about the curvature of spacetime and spacetime itself. As we know, all the information about the curvature of spacetime is captured in the Riemann tensor. But the Riemann tensor is not a scalar-valued function - it is instead a (rank-4) tensor-valued function. So we have to find a way to get a scalar from the Riemann tensor. We already know of a scalar that can be formed from the Riemann tensor - the Ricci scalar. We want to add an additional proportionality constant in front, which is also a scalar, because we'd expect to see constants in our final field equations as well. We can always set the constant $\kappa = 1$ if we find it's not necessary later. Since both the curvature of spacetime and the matter and energy present within spacetime should act on the metric, we add them together. Finally, since spacetime is often curved, we need a factor of $\sqrt{-g}$ to make sure the volume element d^4x is the same size no matter what coordinates or what spacetime we use. So from there, we obtain the action.

From our action, we know that the Lagrangian is:

$$L = \kappa R\sqrt{-g} \tag{1320}$$

We will use the Euler-Lagrange field equations, a slight variation of the original Euler-Lagrange equations we derived:

$$\frac{\partial L}{\partial \varphi} - \frac{\partial}{\partial x^\beta}\left(\frac{\partial L}{\partial(\partial_\beta \varphi)}\right) = 0 \tag{1321}$$

Here, φ is the field, and in our case, the field is the metric tensor field $g_{\mu\nu}(x^\beta)$, thus $\varphi = g_{\mu\nu}$, so if we substitute, we have:

$$\frac{\partial L}{\partial g_{\mu\nu}} - \frac{\partial}{\partial x^\beta}\left(\frac{\partial L}{\partial(\partial_\beta g_{\mu\nu})}\right) = 0 \tag{1322}$$

Note that we use the curly L for the Lagrangian because it is not technically the Lagrangian per se, but the field equivalent of the Lagrangian, known as the **Lagrangian density**. But we'll just call it the Lagrangian here. The distinction between the Lagrangian density and the Lagrangian isn't important here; the practical difference here is that the Lagrangian uses the typical Euler-Lagrange equation, while the Lagrangian density uses the Euler-Lagrange *field* equation.

We notice in the Euler-Lagrange field equations that the second term contains the partial derivative with respect to the derivatives of the metric. But note that in our Lagrangian, there are no terms

that take the derivative of the metric as input. So the second term vanishes, and we are left with a comparatively easier equation:

$$\frac{\partial L}{\partial g_{\mu\nu}} = 0 \tag{1323}$$

Before we take this derivative, let us first rewrite our Lagrangian as:

$$L = \kappa g^{\mu\nu} R_{\mu\nu} \sqrt{-g} \tag{1324}$$

Now, we can finally take the derivative with respect to the metric:

$$\frac{\partial L}{\partial g_{\mu\nu}} = \kappa \frac{\partial}{\partial g_{\mu\nu}} \left(g^{\mu\nu} R_{\mu\nu} \sqrt{-g} \right) = 0 \tag{1325}$$

We immediately run into a hurdle! The Lagrangian has three multiplied functions, the inverse metric, the Ricci tensor, and the square root of the determinant of the metric. How do we differentiate a triple product? We can use the triple product rule:

$$(f \cdot g \cdot h)' = f'gh + fg'h + fgh' \tag{1326}$$

Another problem! How do we differentiate the inverse metric with respect to the metric? The answer comes from a matrix calculus identity, which, translated to tensor notation, is this:

$$\frac{\partial g^{\mu\nu}}{\partial g_{\mu\nu}} = -g^{\mu\nu} g^{\mu\nu} \tag{1327}$$

Final problem! How do we differentiate the determinant of the metric with respect to the metric? This answer also comes from a matrix calculus identity, which is this:

$$\frac{\partial \det(g_{\mu\nu})}{\partial g_{\mu\nu}} = \frac{\partial g}{\partial g_{\mu\nu}} = g g^{\mu\nu} \tag{1328}$$

With all this in mind, we can finally compute the derivatives. The first term of the derivative is just the derivative of the inverse metric, multiplied by the other two terms in the triple product. The derivative of the Ricci tensor with respect to the metric is zero (it doesn't depend on the metric), so the second term of the derivative of the triple product is zero. In the third term, we need to use the chain rule to differentiate the square root. The final result is this:

$$-\kappa R_{\mu\nu} g^{\mu\nu} g^{\mu\nu} - \kappa \frac{1}{2\sqrt{-g}} g g^{\mu\nu} g^{\mu\nu} R_{\mu\nu} = 0 \tag{1329}$$

We can clean this up a bit. First, we can multiply both sides by -1, to get:

$$\kappa R_{\mu\nu} g^{\mu\nu} g^{\mu\nu} + \kappa \frac{1}{2\sqrt{-g}} g g^{\mu\nu} g^{\mu\nu} R_{\mu\nu} = 0 \tag{1330}$$

Then, we can multiply both sides of the equation by $\frac{1}{\sqrt{-g}}$, which results in:

$$\kappa R_{\mu\nu} g^{\mu\nu} g^{\mu\nu} - \kappa \frac{1}{2} g^{\mu\nu} g^{\mu\nu} R_{\mu\nu} = 0 \tag{1331}$$

We remember that $R = g^{\mu\nu} R_{\mu\nu}$, so we can substitute it in:

$$\kappa R_{\mu\nu} g^{\mu\nu} g^{\mu\nu} - \kappa \frac{1}{2} g^{\mu\nu} R = 0 \tag{1332}$$

We want to get rid of the double $g^{\mu\nu}$ terms, so we can multiply both sides of the equation by $g_{\mu\nu} g_{\mu\nu}$, to get:

$$\kappa R_{\mu\nu} g^{\mu\nu} g^{\mu\nu} g_{\mu\nu} g_{\mu\nu} - \kappa \frac{1}{2} g^{\mu\nu} g_{\mu\nu} g_{\mu\nu} R = 0 \tag{1333}$$

The inverse metric contracts with the metric:

$$g^{\mu\nu}g^{\mu\nu}g_{\mu\nu}g_{\mu\nu} = g^{\mu\nu}g_{\mu\nu} = \delta^{\mu}_{\mu} = \sum_{i=0}^{3} 1 = 4 \tag{1334}$$

So this entire expression becomes:

$$\kappa 4R_{\mu\nu} - \kappa 2Rg_{\mu\nu} = 0 \tag{1335}$$

But we can divide by 4 right after as the right-hand side is zero, to yield:

$$\kappa R_{\mu\nu} - \kappa\frac{1}{2}Rg_{\mu\nu} = 0 \tag{1336}$$

We can factor out the constant:

$$\kappa\left(R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu}\right) = 0 \tag{1337}$$

The term inside the parentheses is called the **Einstein tensor** and describes the curvature and characteristics of spacetime:

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} \tag{1338}$$

In vacuum, the equation we just derived is the Einstein Field Equation:

$$G_{\mu\nu} = 0 \tag{1339}$$

Now, there is matter and energy within space, then we use a modified action, where $\mathcal{M}$ is the contribution to the action of the gravitating matter and energy:

$$S = \int (\kappa R - \mathcal{M})\sqrt{-g}\ d^4x \tag{1340}$$

So the Lagrangian is:

$$L = (\kappa R - \mathcal{M})\sqrt{-g} \tag{1341}$$

Using the Euler-Lagrange field equations, this becomes:

$$-\kappa R_{\mu\nu}g^{\mu\nu}g^{\mu\nu}\sqrt{-g} - \kappa\frac{1}{2\sqrt{-g}}gg^{\mu\nu}g^{\mu\nu}R_{\mu\nu} - \frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}\sqrt{-g} + \frac{1}{2\sqrt{-g}}gg^{\mu\nu}\mathcal{M} = 0 \tag{1342}$$

First, we multiply by -1:

$$\kappa R_{\mu\nu}g^{\mu\nu}g^{\mu\nu}\sqrt{-g} + \kappa\frac{1}{2\sqrt{-g}}gg^{\mu\nu}g^{\mu\nu}R_{\mu\nu} + \frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}\sqrt{-g} - \frac{1}{2\sqrt{-g}}gg^{\mu\nu}\mathcal{M} = 0 \tag{1343}$$

Then we multiply by $\frac{1}{\sqrt{-g}}$:

$$\kappa R_{\mu\nu}g^{\mu\nu}g^{\mu\nu} - \kappa\frac{1}{2}g^{\mu\nu}g^{\mu\nu}R_{\mu\nu} + \frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}\sqrt{-g} + \frac{1}{2}g^{\mu\nu}\mathcal{M} = 0 \tag{1344}$$

We use the definition $R = g^{\mu\nu}R_{\mu\nu}$:

$$\kappa R_{\mu\nu}g^{\mu\nu}g^{\mu\nu} - \kappa\frac{1}{2}g^{\mu\nu}R + \frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}\sqrt{-g} + \frac{1}{2}g^{\mu\nu}\mathcal{M} = 0 \tag{1345}$$

And by contraction with $g_{\mu\nu}g_{\mu\nu}$ we have:

$$\kappa R_{\mu\nu} - \kappa\frac{1}{2}g_{\mu\nu}R + \frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}g_{\mu\nu}g_{\mu\nu}\sqrt{-g} + \frac{1}{2}g_{\mu\nu}\mathcal{M} = 0 \tag{1346}$$

We can move the second and third terms, which depend on $\mathcal{M}$ to the right of the equation:

$$\kappa R_{\mu\nu} - \kappa\frac{1}{2}g_{\mu\nu}R = -\frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}g_{\mu\nu}g_{\mu\nu}\sqrt{-g} - \frac{1}{2}g_{\mu\nu}\mathcal{M} \tag{1347}$$

And factor the left-hand side of the equation:

$$\kappa\left(R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R\right) = -\frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}g_{\mu\nu}g_{\mu\nu}\sqrt{-g} - \frac{1}{2}g_{\mu\nu}\mathcal{M} \tag{1348}$$

We recognize our familiar friend, the Einstein tensor, on the left. If we define a tensor $T_{\mu\nu}$ to equal the right-hand side:

$$T_{\mu\nu} = -\frac{\partial\mathcal{M}}{\partial g_{\mu\nu}}g_{\mu\nu}g_{\mu\nu}\sqrt{-g} - \frac{1}{2}g_{\mu\nu}\mathcal{M} \tag{1349}$$

Then we have the complete field equations:

$$G_{\mu\nu} = \frac{1}{\kappa}T_{\mu\nu} \tag{1350}$$

The tensor $T_{\mu\nu}$ on the right is called the **stress-energy tensor**. There is no one "general formula" for the stress-energy tensor; we can define different expressions for the stress-energy tensor depending on what matter, energy, momentum, and stresses are present within the region of spacetime being analyzed, with the only real rule being that the resulting expression follow tensor algebra conventions (e.g. same number of free indices on both sides of the equation). One of the simplest expressions for a stress-energy tensor is:

$$T_{\mu\nu} = \rho U_\mu U_\nu \tag{1351}$$

Here, U_μ and U_ν are four-velocities, as shown before in special relativity, and ρ is the density of the gravitating matter-energy.

But back to the equation:

$$G_{\mu\nu} = \frac{1}{\kappa}T_{\mu\nu} \tag{1352}$$

What is the constant κ? We will need to use the Newtonian limit of relativity to answer that question. When gravity is weak, and objects are moving much slower than the speed of light, we expect that we can recover Poisson's equation from the field equation. We will cover that in the following derivation.

Given that four-velocity is defined as $U_\mu = (\gamma c, \gamma v)$, and we defined objects to be moving much slower than the speed of light, the 0th component of four-velocity, so slow that their speeds are effectively zero compared to the speed of light, we can effectively say that $\gamma \approx 1$ and $U_\mu \approx (c, 0, 0, 0)$. Therefore, the component T_{00} of the stress-energy tensor is just ρc^2, and all other components of the stress-energy tensor are zero.

Given a static metric, that is, one that doesn't change much in time, we can also say that $\partial_0 g_{\mu\nu} = 0$. And because we expect spacetime to be very close to Minkowski spacetime, we will set:

$$g_{\mu\nu} = \eta_{\mu\nu} + h_{\mu\nu} \tag{1353}$$

Where $h_{\mu\nu}$ is a tiny bit of metric that accounts for Newtonian gravity. Also, given our low velocity assumtions, we already mentioned that:

$$U^\mu = \left(\frac{dt}{d\tau}, \frac{dx}{d\tau}, \frac{dy}{d\tau}, \frac{dz}{d\tau}\right) \approx (c, 0, 0, 0) \tag{1354}$$

We consider the geodesic equations:

$$\frac{d^2x^\mu}{d\tau^2} + \Gamma^\mu_{\gamma\sigma}\frac{dx^\gamma}{d\tau}\frac{dx^\sigma}{d\tau} = 0 \tag{1355}$$

We can simplify given that given that only the $\frac{dt}{d\tau}$ component matters, because all the other velocities are zero:

$$\frac{d^2x^\mu}{d\tau^2} + \Gamma^\mu_{00}\frac{dt}{d\tau}\frac{dt}{d\tau} = 0 \tag{1356}$$

We can rewrite this as:

$$\frac{d^2x^\mu}{d\tau^2} = -\Gamma^\mu_{00}\frac{dt}{d\tau}\frac{dt}{d\tau} \tag{1357}$$

And recalling $\frac{dt}{d\tau} \approx c$, we have:

$$\frac{d^2x^\mu}{d\tau^2} = -\Gamma^\mu_{00}c^2 \tag{1358}$$

Now, we can compare this to Newton's equation for gravity:

$$\frac{d^2r}{dt^2} = -\nabla\phi \tag{1359}$$

Therefore:

$$\Gamma^\mu_{00}c^2 = \nabla\phi \tag{1360}$$

$$\Gamma^\mu_{00} = \frac{1}{c^2}\nabla\phi \tag{1361}$$

Now, given that:

$$\Gamma^i_{kl} = \frac{1}{2}g^{im}(\partial_l g_{mk} + \partial_k g_{ml} - \partial_m g_{kl}) \tag{1362}$$

If we substitute $i \to \mu, k \to t, l \to t$, we have:

$$\Gamma^\mu_{00} = \frac{1}{2}g^{\mu m}(\partial_t g_{mt} + \partial_t g_{mt} - \partial_m g_{00}) \tag{1363}$$

But recall that $\partial_t g_{\mu\nu} = 0$ given our static metric assumption, so:

$$\Gamma^\mu_{00} = -\frac{1}{2}g^{\mu m}\partial_m g_{00} \tag{1364}$$

Since $g^{\mu\nu} = \eta^{\mu\nu} - h^{\mu\nu}$, we have:

$$\Gamma^\mu_{00} = -\frac{1}{2}(\eta^{\mu m} - h^{\mu m})(\partial_m \eta_{00} - \partial_m h_{00}) \tag{1365}$$

We can simplify this by noting that because $\eta_{00} = -1$, its derivative is zero, so:

$$\Gamma^\mu_{00} = -\frac{1}{2}(\eta^{\mu m} - h^{\mu m})\partial_m h_{00} \tag{1366}$$

If we expand this out, we would get:

$$\Gamma^\mu_{00} = -\frac{1}{2}\eta^{\mu m}\partial_m h_{00} + h^{\mu m}\partial_m h_{00} \tag{1367}$$

But the second term is very tiny, so we can effectively say it is zero, and given $\eta^{\mu m} = -1$, we get:

$$\Gamma^\mu_{00} = \frac{1}{2}\partial_m h_{00} \tag{1368}$$

Note
How did we know that $\eta_{\mu m} = -1$? This is because $\eta_{\mu m}$ is the first column of the Minkowski metric (as m is a dummy index, so you can think of it as $\eta_{0m} \dots \eta_{3m}$), and the only non-zero term in its first column is η_{00}.

The partial derivative with respect to an arbitrary coordinate is just the gradient:

$$\Gamma^{\mu}_{00} = \frac{1}{2}\nabla h_{00} \tag{1369}$$

We already know what Γ^{μ}_{00} is though, so:

$$\frac{1}{c^2}\nabla\phi = \frac{1}{2}\nabla h_{00} \tag{1370}$$

So we find that $h_{00} = \frac{2\phi}{c^2}$, and therefore $g_{00} = -1 - \frac{2\phi}{c^2}$.

Now, we have everything we need to compute the Einstein tensor. Using the definition of the Ricci tensor, we have:

$$R_{ij} = \partial_k\Gamma^k_{ij} - \partial_j\Gamma^k_{ik} + \Gamma^k_{ij}\Gamma^m_{km} - \Gamma^k_{im}\Gamma^m_{jk} \tag{1371}$$

Given our knowledge of Γ^{μ}_{00}, and with substitutions $k \to \mu, i \to t, j \to t$, we have:

$$R_{00} = \partial_\mu\Gamma^{\mu}_{00} - \partial_t\Gamma^{\mu}_{t\mu} + \Gamma^{\mu}_{00}\Gamma^m_{\mu m} - \Gamma^{\mu}_{tm}\Gamma^m_{t\mu} \tag{1372}$$

But recall that the time derivative of the metric must be zero, so the second term cancels out. And recall that all the Christoffel symbols that are not Γ^{μ}_{00} are zero. If we expand out the dummy summation indices, we find that they all go to zero. Therefore, we just have:

$$R_{00} = \partial_\mu\Gamma^{\mu}_{00} \tag{1373}$$

Which becomes:

$$R_{00} = \frac{1}{c^2}\nabla^2\phi \tag{1374}$$

Recall the definition of the Einstein tensor:

$$G_{\mu\nu} = R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}g^{\mu\nu}R_{\mu\nu} \tag{1375}$$

If we substitute, we have:

$$G_{00} = R_{00} - \left[\left(-1 - \frac{2\phi}{c^2}\right)R_{00}\right] = \frac{1}{c^2}\nabla^2\phi + \frac{1}{c^2}\nabla^2\phi + \frac{2\phi}{c^4}\nabla^2\phi \tag{1376}$$

Now, since c^4 is a massive number, $\frac{2\phi}{c^4} \approx 0$, so:

$$G_{00} = \frac{2}{c^2}\nabla^2\phi \tag{1377}$$

Using the Einstein field equations:

$$G_{\mu\nu} = \frac{1}{\kappa}T_{\mu\nu} \tag{1378}$$

We can substitute in our values for the Einstein and stress-energy tensors:

$$\frac{2}{c^2}\nabla^2\phi = \frac{1}{\kappa}\rho c^2 \tag{1379}$$

$$\nabla^2\phi = \frac{1}{2\kappa}\rho c^4 \tag{1380}$$

Compare this with Poisson's equation:

$$\nabla^2\phi = 4\pi G\rho \tag{1381}$$

This means that:

$$\frac{1}{2\kappa}\rho c^4 = 4\pi G\rho \tag{1382}$$

$$\kappa = \frac{c^4}{8\pi G} \tag{1383}$$

Remember the field equations:

$$G_{\mu\nu} = \frac{1}{\kappa}T_{\mu\nu} \tag{1384}$$

Now knowing the value of κ, we need only substitute to get:

$$G_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} \tag{1385}$$

This elegant equation is the apotheosis of general relativity, and it rightfully deserves its place as one of the most famous equations in all of physics.

Note that sometimes, there is an alternate form of the Einstein Field Equations that is easier to solve. To do this, we expand out the full equations:

$$R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} \tag{1386}$$

We now multiply both sides by $g^{\mu\nu}$:

$$g^{\mu\nu}R_{\mu\nu} - \frac{1}{2}Rg_{\mu\nu}g^{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu}g^{\mu\nu} \tag{1387}$$

Using the fact that $g_{\mu\nu} = g^{\mu\nu} = 4$ and $T_{\mu\nu}g^{\mu\nu} = T$, this becomes:

$$R - \frac{1}{2}(4R) = \frac{8\pi G}{c^4}T \Rightarrow -R = \frac{8\pi G}{c^4}T \tag{1388}$$

So, substituting back into the original EFEs:

$$R_{\mu\nu} + \frac{1}{2}\frac{8\pi G}{c^4}Tg_{\mu\nu} = \frac{8\pi G}{c^4}T_{\mu\nu} \tag{1389}$$

$$R_{\mu\nu} = \frac{8\pi G}{c^4}\left(T_{\mu\nu} - \frac{1}{2}g_{\mu\nu}T\right) \tag{1390}$$

This makes the field equations simpler for vacuum solutions, where $T_{\mu\nu} = T = 0$. Thus, the equations just become:

$$R_{\mu\nu} = 0 \tag{1391}$$

which is still incredibly hard to solve, but more manageable than the typical case.

Finally, there is one more important fact about the field equations: taking the covariant derivatives of both sides is equal to zero. This means that:

$$\nabla_\mu T_{\mu\nu} = 0 \tag{1392}$$

This expression may look familiar if we recall that the covariant derivative with a repeated index is just the divergence of a field. What this is saying is that the total change in matter-energy flux in all of spacetime is zero - essentially, the conservation of energy.

A recap with intuition After doing so much math, it is helpful to reconnect with what the math is actually *saying*. That is, we want to regain our physical intuition for what the math describes.

Gravity is a fictitious force, caused by the curvature of spacetime. When spacetime isn't curved, particles undergo no acceleration, and thus feel no gravitational force. But when spacetime is curved, which happens whenever masses are present in spacetime, particles undergo a definite acceleration. Due to the equivalence principle, the effect of gravity is indistinguishable from the effect of an acceleration, so therefore particles that are being accelerated feel like a force is acting on them.

The gravitational field is an object that extends through all space that gives each point a vector proportional to the gravitational force. Masses create and vary the gravitational field, and in turn the field exerts a force on masses within the field.

The gravitational potential is a function whose slope is equal to the gravitational field. It can be thought of as a landscape that masses are placed in. Where that landscape is very steep, the gravitational force is very strong; where that landscape is very flat, the gravitational force is very weak.

The metric tensor is a mathematical description of a spacetime. The classical analogue of it is the gravitational potential. Just as the gravitational potential influences the force of gravity, the metric tensor influences the curvature of spacetime, which particles experience as gravity.

We can measure distances in typical Euclidean space using a fixed grid, where the increments between the grid line are measured by constant basis vectors. In curved spacetime, basis vectors are no longer constant. The Christoffel symbols are a precise measure of how basis vectors changes in spacetime, or essentially, how the Euclidean constant grid gets distorted in spacetime. It is roughly analogous to the gravitational field. Just as a gravitational field disappears in empty space far away from any masses, the Christoffel symbols vanish too.

The Riemann tensor describes how the curvature of spacetime changes a vector as you move it in different directions in spacetime. The Ricci tensor describes how a volume in spacetime located at a given point in spacetime becomes contracted due to the curvature of spacetime. The Einstein tensor is the *average* value of this contraction of volume across all of the region of spacetime being studied.

The stress-energy tensor describes the matter-energy fluxes within a region of spacetime. The classical analogue would be matter density. Finally, just as matter is the source for gravity in Newtonian mechanics, matter is the source of spacetime curvature in General Relativity, and spacetime curvature is what we feel as gravity. John Archibald Wheeler famously summarized all of these ideas with the succint observation:

> "Spacetime tells matter how to move; matter tells spacetime how to curve"

Quantum field theory, Part 1

Up to this point, we have seen both Special Relativity and General Relativity, both of which are written in the language of tensors. But we have only seen relativity on big scales acting on macroscopic objects, not the small scales of quantum mechanics. So it natural to ask whether we can describe *quantum particles* in a relativistic fashion.

The answer is yes, and in fact, the study of the intersection of relativity and quantum mechanics - called **relativistic quantum field theory** - is central to modern physics. It also has useful applications in giving a theoretical explanation of previously-unknown atomic transitions and performing incredibly-precise quantum calculations for atoms, particularly in extreme high-energy conditions. Some features of quantum mechanics we take for granted - like the photon as the quantum of light - are actually *only* possible to fully explain using quantum field theory. So let's dive in!

Introduction to quantum electrodynamics Throughout our discussion of spontaneous and stimulated emission, we have discussed photons everywhere. But photons are not something that can be completely described using the Schrödinger equtaion. Photons must be described by **quantum electrodynamics**, the quantum theory of the electromagnetic field.

> **Important**
> We use Gaussian units instead of SI units for this section. Consult a conversions chart to convert Gaussian units to SI units.

Remember that light is classically-described as a wave, which obeys the solution to the Maxwell equations of electromagnetism in free space:

$$\frac{\partial^2 \mathbf{E}}{\partial t^2} - c^2 \nabla^2 \mathbf{E} = 0 \tag{1393}$$

$$\frac{\partial^2 \mathbf{B}}{\partial t^2} - c^2 \nabla^2 \mathbf{B} = 0 \tag{1394}$$

But in classical electrodynamics, we find that due to relativistic effects, the electric field and magnetic field can no longer be thought of as distinct entities. Rather, they must be considered together as part of an **electromagnetic field**. Instead of separate electric and magnetic fields, we define a common **electromagnetic 4-potential**, which is a more fundamental physical quantity from which the fields arise. The electromagnetic 4-potential is given by $A^\mu = (\varphi, A_x, A_y, A_z)$ where φ is the electric scalar potential and $\mathbf{A} = (A_x, A_y, A_z)$ is the magnetic vector potential. The definitions of φ and $\mathbf{A}$ are:

$$\mathbf{E} = -\nabla\varphi - \frac{1}{c}\frac{\partial \mathbf{A}}{\partial t} \tag{1395}$$

$$\mathbf{B} = \nabla \times \mathbf{A} \tag{1396}$$

The electromagnetic 4-potential, being a *potential*, can be shifted by an arbitrary constant without changing the fields (and thus the physics). Indeed, they can even be shifted by the gradient of a function $f(\mathbf{r}, t)$ without changing the fields, with the transformations:

$$\varphi = \varphi - \frac{\partial f}{\partial t} \tag{1397}$$

$$\mathbf{A} = \mathbf{A} + \nabla f \tag{1398}$$

This is known as **gauge freedom**, and therefore we must impose certain conditions to restrict the electromagnetic four-potential to a particular form, just as we define a reference position in classical mechanics to define the classical potential energy. Applying these chosen conditions (called "gauges")

is called **gauge fixing**. There are two common gauges used: the Coulomb gauge, commonly used for non-relativistic quantum field theory, and the Lorenz (note: not "Lorentz") gauge, usually used for relativistic quantum field theory. As we work in the non-relativistic regime, we will choose the Coulomb gauge, which sets the condition $\nabla \cdot \mathbf{A} = 0$. Thus, the equations of motion become:

$$\frac{\partial^2 \varphi}{\partial t^2} - c^2 \nabla^2 \varphi = 0 \tag{1399}$$

$$\frac{\partial^2 \mathbf{A}}{\partial t^2} - c^2 \nabla^2 \mathbf{A} = 0 \tag{1400}$$

The general, normalized solutions to the system of PDEs in a region of volume V is given by:

$$\mathbf{A}(\mathbf{r}, t) = \frac{1}{V\sqrt{2}} \sum_{\mathbf{k}} \left[\mathbf{A}_{\mathbf{k}}(\mathbf{k}) e^{i\mathbf{k}\cdot\mathbf{r}} + \mathbf{A}^*_{\mathbf{k}}(\mathbf{k}) e^{-i\mathbf{k}\cdot\mathbf{r}} \right] \tag{1401}$$

Where $\mathbf{k}$ is the **wavevector**, which is directly related to the momentum of a wave as well as its wavelength by $|\mathbf{k}| = \dfrac{2\pi}{\lambda}$, and where $\mathbf{A}_{\mathbf{k}}(\mathbf{k})$ represents the polarization (directionality) and field strength of the wave, which are given respectively by the direction and the magnitude of $\mathbf{A}_{\mathbf{k}}(\mathbf{k})$. Here, $\mathbf{A}^*_{\mathbf{k}}$ represents the complex conjugate of $\mathbf{A}_{\mathbf{k}}(\mathbf{k})$. Those who are aware may recognize this as a Fourier expansion in terms of the normal modes; if that terminology is unfamiliar, don't worry about it. Importantly, the wavevector is related to the **angular frequency** by $\omega_{\mathbf{k}} = \mathbf{k}c$, which will be significant later.

> **Note**
> The series is significant, as pure plane waves in the form $e^{i\mathbf{k}\cdot\mathbf{r}}$, which are the simplest solutions to the PDEs, are mathematically consistent but physically impossible (they would need an infinite amount of energy to create). The sum over plane waves of different $\mathbf{k}$ and $\mathbf{A}_{\mathbf{k}}(\mathbf{k})$, however, produces a physical solution (as long as we just take the real part of the complex-valued waves).

The classical field theory of electromagnetism Now, we will use the tools of *classical field theory* to be able to analyze the electromagnetic field, because classical quantities of fields translate into operators in quantum field theory. In classical field theory, we define a **Lagrangian** $\mathcal{L}(q, \dot{q}, \nabla q)$ based on the coordinate q as well as its time and space derivatives (in more precise terms we should write q^μ to denote that this is a coordinate). In our case, $q = A^\mu$ is the electromagnetic four-potential. We may also define a **Hamiltonian** $\mathcal{H}(q, p)$ based on the coordinate q as well as the *canonical momentum* p obtained from $p = \dfrac{\partial \mathcal{L}}{\partial \dot{q}}$ (this would also be more precisely be written as p^μ). The Lagrangian of the electromagnetic field in free space is given by:

$$\mathcal{L} = \frac{1}{8\pi}(\mathbf{E}^2 - \mathbf{B}^2) \tag{1402}$$

$$= \frac{1}{8\pi}\left(-\nabla\varphi - \frac{1}{c}\frac{\partial \mathbf{A}}{\partial t}\right)^2 - \frac{1}{8\pi}(\nabla \times \mathbf{A})^2 \tag{1403}$$

$$\tag{1404}$$

We can also write this in terms of the **canonical momentum**. First, let us derive it by findind the derivative of the Lagrangian with respect to $\dfrac{\partial \mathbf{A}}{\partial t}$. This comes out to:

$$\mathbf{p} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{A}}} \tag{1405}$$
$$= \frac{\partial \mathcal{L}}{\partial(\partial_t \mathbf{A})} \tag{1406}$$
$$= \frac{1}{4\pi c}\left(\nabla\varphi + \frac{1}{c}\frac{\partial \mathbf{A}}{\partial t}\right) \tag{1407}$$
$$= -\frac{1}{4\pi c}\mathbf{E} \tag{1408}$$

That is to say, we have found that the canonical momentum of the electromagnetic four-potential $\mathbf{p}$ is related to the electric field by $\mathbf{E} = -4\pi c\mathbf{p}$. This result allows us to write the Lagrangian in the simplified form:

$$\mathcal{L} = \frac{1}{8\pi}(\mathbf{E}^2 - \mathbf{B}^2) \tag{1409}$$
$$= \frac{1}{8\pi}(-4\pi c\mathbf{p})^2 - \frac{1}{8\pi}(\nabla \times \mathbf{A})^2 \tag{1410}$$
$$= 2\pi c^2\mathbf{p}^2 - \frac{1}{8\pi}(\nabla \times \mathbf{A})^2 \tag{1411}$$

It also means that we can rearrange to express $\dfrac{\partial \mathbf{A}}{\partial t}$ in terms of $\mathbf{p}$, something that will be very useful later:

$$\mathbf{p} = \frac{1}{4\pi c}\left(\nabla\varphi + \frac{1}{c}\frac{\partial \mathbf{A}}{\partial t}\right) \tag{1412}$$
$$4\pi c\mathbf{p} = \nabla\varphi + \frac{1}{c}\frac{\partial \mathbf{A}}{\partial t} \tag{1413}$$
$$c(4\pi c\mathbf{p} - \nabla\varphi) = \frac{\partial \mathbf{A}}{\partial t} \tag{1414}$$

We may use these results to find the Hamiltonian of the electromagnetic field:

$$\mathcal{H} = \mathbf{p}\cdot\dot{\mathbf{q}} - \mathcal{L} \tag{1415}$$
$$= \mathbf{p}\cdot\frac{\partial \mathbf{A}}{\partial t} - 2\pi c^2\mathbf{p}^2 + \frac{1}{8\pi}(\nabla \times \mathbf{A})^2 \tag{1416}$$
$$= \mathbf{p}\cdot c(4\pi c\mathbf{p} - \nabla\varphi) - 2\pi c^2\mathbf{p}^2 + \frac{1}{8\pi}(\nabla \times \mathbf{A})^2 \tag{1417}$$
$$= 2\pi c^2\mathbf{p}^2 - c\mathbf{p}\cdot\nabla\varphi + \frac{1}{8\pi}(\nabla \times \mathbf{A})^2 \tag{1418}$$

Up to this point, we have been very sloppy with our language: what we have been calling the Lagrangian is actually called the **Lagrangian density**, and what we have been calling the Hamiltonian is actually called the **Hamiltonian density**. We may find the Lagrangian and the Hamiltonian by integrating the Lagrangian and Hamiltonian densities over space (here $d^3x = dx\,dy\,dz$):

$$L = \int \mathcal{L}\, d^3x \tag{1419}$$
$$= \int d^3x \left[2\pi c^2\mathbf{p}^2 - \frac{1}{8\pi}(\nabla \times \mathbf{A})^2\right] \tag{1420}$$
$$H = \int \mathcal{H}\, d^3x \tag{1421}$$
$$= \int d^3x \left[2\pi c^2\mathbf{p}^2 - c\mathbf{p}\cdot\nabla\varphi + \frac{1}{8\pi}(\nabla \times \mathbf{A})^2\right] \tag{1422}$$

But notice something peculiar. If we explicitly evaluate the middle term, using the results we have derived previously, we get:

$$c\mathbf{p}\cdot\nabla\varphi = c\left(-\frac{1}{4\pi c}\mathbf{E}\right)\cdot\nabla\varphi \tag{1423}$$

$$= \underbrace{-\frac{1}{4\pi}(\nabla\cdot\mathbf{E})\varphi}_{\mathbf{F}\cdot(\nabla\phi)=(\mathbf{F}\cdot\nabla)\phi} \tag{1424}$$

$$= 0 \tag{1425}$$

Which comes from the vector calculus identity $\mathbf{F}\cdot(\nabla\phi) = (\mathbf{F}\cdot\nabla)\phi$ and the fact that $\nabla\cdot\mathbf{E} = 0$ in free space. Thus the middle term in our expression for the Hamiltonian drops out, leaving us with just:

$$H = \int d^3x \left[2\pi c^2\mathbf{p}^2 + \frac{1}{8\pi}(\nabla\times\mathbf{A})^2\right] \tag{1426}$$

> **Note**
> Further, the Coulomb gauge enforces the requirement that $\varphi = 0$ in our given conditions, so we could have done all of this without needing to do the math.

This, together with the general solution we found for $\mathbf{A}(\mathbf{r},t)$, lays the classical foundation for quantizing the electromagnetic field.

Quantization of the electromagnetic field Before we proceed onto the process of treating the electromagnetic field in a fully quantum way, let us take some time to review the **quantum harmonic oscillator**. The quantum harmonic oscillator has a Hamiltonian $\hat{H}$ given by:

$$\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2}m\omega^2\hat{q}^2 \tag{1427}$$

Where $\hat{q}$ is the coordinate used, which, in non-QFT, is equal to the position operator $\hat{x}$. One may write the Hamiltonian in a more useful form by defining the operators $\hat{a}$ and $\hat{a}^\dagger$, as follows:

$$\hat{a} = \sqrt{\frac{m\omega}{2\hbar}}\hat{q} + i\frac{1}{\sqrt{2m\omega\hbar}}\hat{p} \tag{1428}$$

$$\hat{a}^\dagger = \sqrt{\frac{m\omega}{2\hbar}}\hat{q} - i\frac{1}{\sqrt{2m\omega\hbar}}\hat{p} \tag{1429}$$

$$\tag{1430}$$

Which allows the Hamiltonian to be written as:

$$\hat{H} = \hbar\omega\left(\hat{a}\hat{a}^\dagger - \frac{1}{2}\right) \tag{1431}$$

We may now take these results for the non-QFT quantum harmonic oscillator and generalize it to the quantized electromagnetic field. To do so, we promote the *classical quantities* to their quantum operator analogues: we replace the generalized coordinate $\hat{q}$ with our field expression $\hat{q} \to \mathbf{A}_\mathbf{k}$, and we replace the momentum operator $\hat{p}$ with the canonical momentum we just derived earlier, and substitute them into the $\hat{a}$ and $\hat{a}^\dagger$ operators. For the momentum, we know that $\mathbf{p} = -\frac{1}{4\pi c}\mathbf{E}$; using the relationship $\mathbf{E} = -\nabla\varphi - \frac{1}{c}\frac{\partial\mathbf{A}}{\partial t}$, which just becomes $\mathbf{E} = -\frac{1}{c}\frac{\partial\mathbf{A}}{\partial t}$ in our case, yields:

$$\mathbf{p} = -\frac{1}{4\pi c}\mathbf{E} \tag{1432}$$
$$= \frac{1}{4\pi c^2}\frac{\partial \mathbf{A}}{\partial t} \tag{1433}$$
$$= \frac{-i}{8\pi c^2\sqrt{V}}\omega_{\mathbf{k}} \sum_{\mathbf{k}} \left[\mathbf{A}_{\mathbf{k}}(\mathbf{k})e^{i\mathbf{k}\cdot\mathbf{r}} + \mathbf{A}^*_{\mathbf{k}}(\mathbf{k})e^{-i\mathbf{k}\cdot\mathbf{r}}\right] \tag{1434}$$
$$= \frac{1}{2\sqrt{V}} \sum_{\mathbf{k}} \left[\mathbf{p}_{\mathbf{k}}e^{i\mathbf{k}\cdot\mathbf{r}} + \mathbf{p}^*_{\mathbf{k}}e^{-i\mathbf{k}\cdot\mathbf{r}}\right] \tag{1435}$$
$$\Rightarrow \mathbf{p}_{\mathbf{k}} = \frac{-i\omega_{\mathbf{k}}}{4\pi c^2}\mathbf{A}_{\mathbf{k}} \tag{1436}$$

From here, we may define the operators $\hat{Q}$ and $\hat{P}$, by:

$$\hat{Q} = \sqrt{\frac{m\omega_{\mathbf{k}}}{\hbar}}\hat{q} \tag{1437}$$
$$= \sqrt{\frac{m\omega_{\mathbf{k}}}{\hbar}}\mathbf{A}_{\mathbf{k}} \tag{1438}$$
$$\hat{P} = \frac{1}{\sqrt{m\hbar\omega_{\mathbf{k}}}}\hat{p} \tag{1439}$$
$$= \frac{1}{\sqrt{m\hbar\omega_{\mathbf{k}}}}\left(\frac{-i\omega_{\mathbf{k}}}{4\pi c^2}\mathbf{A}_{\mathbf{k}}\right) \tag{1440}$$
$$= \frac{-i}{4\pi c^2}\frac{1}{\sqrt{m\hbar\omega_{\mathbf{k}}}}\mathbf{A}_{\mathbf{k}} \tag{1441}$$

For which we then have:

$$\hat{a} = \frac{1}{\sqrt{2}}(\hat{Q} + i\hat{P}) \tag{1442}$$
$$= \sqrt{\frac{m\omega_{\mathbf{k}}}{2\hbar}}\mathbf{A}_{\mathbf{k}} + \frac{1}{4\pi c^2}\frac{1}{\sqrt{2m\hbar\omega_{\mathbf{k}}}}\mathbf{A}_{\mathbf{k}} \tag{1443}$$
$$\hat{a}^\dagger = \frac{1}{\sqrt{2}}(\hat{Q} - i\hat{P}) \tag{1444}$$
$$= \sqrt{\frac{m\omega_{\mathbf{k}}}{2\hbar}}\mathbf{A}_{\mathbf{k}} - \frac{1}{4\pi c^2}\frac{1}{\sqrt{2m\hbar\omega_{\mathbf{k}}}}\mathbf{A}_{\mathbf{k}} \tag{1445}$$

The Hamiltonian becomes the Hamiltonian density, and so it must be integrated over all of space:

$$\hat{H} = \int d^3x \left[\hbar\omega_{\mathbf{k}}\left(\hat{a}\hat{a}^\dagger - \frac{1}{2}\right)\right] \tag{1446}$$

It is common to denote the states of the quantum electromagnetic field with $|n\rangle$, where $|0\rangle$ is the ground state, $|1\rangle$ is the first excited state, $|2\rangle$ is the second excited state, and so on. The creation and annihilation operators have the properties that:

$$\hat{a}|n\rangle = c_n|n-1\rangle \tag{1447}$$
$$\hat{a}^\dagger|n\rangle = c^*_n|n+1\rangle \tag{1448}$$
$$\tag{1449}$$

That is to say, applying the creation operator $\hat{a}^\dagger$ raises the electromagnetic field from state $|n\rangle$ to a higher-energy state $|n+1\rangle$, while applying the annihilation operator $\hat{a}$ lowers the electromagnetic field from state $|n\rangle$ to a lower-energy state $|n-1\rangle$, with c_n, c^*_n being normalization constants given by:

$$c_n = \sqrt{n} \tag{1450}$$
$$c_n^* = \sqrt{n+1} \tag{1451}$$

This process continues until we reach the ground state, where $n = 0$, and thus applying the annihilation operator on the ground state yields $\hat{a}|0\rangle = (\sqrt{0})|0\rangle = 0$.

Calculating the ground-state energy of the electromagnetic field When we say "ground-state energy", we should be careful about our use of words: there is no *singular* ground-state energy for a quantum field, as it constantly fluctuates in energy due to the Heisenberg uncertainty principle $\Delta E \Delta t \geq \frac{\hbar}{2}$. Thus the ground-state energy we speak of is in fact the *average* ground-state and would be given by the *expectation value* of the Hamiltonian:

$$\langle E_0 \rangle = \langle 0|\hat{H}|0\rangle \tag{1452}$$
$$= \langle 0| \int d^3x \left[\hbar\omega_{\mathbf{k}} \left(\hat{a}\hat{a}^\dagger - \frac{1}{2}\right)\right] |0\rangle \tag{1453}$$
$$= \langle 0| \left(\int d^3x \left(\hbar\omega_{\mathbf{k}} \hat{a}\hat{a}^\dagger\right) - \frac{1}{2}\int d^3x\, \hbar\omega_{\mathbf{k}}\right) |0\rangle \tag{1454}$$

Remember that the annihilation operator annihilates the ground state, so the first term in the integral is zero. We are simply left with:

$$\langle E_0 \rangle = \langle 0| \left(-\frac{1}{2}\int d^3x\, \hbar\omega_{\mathbf{k}}\right) |0\rangle \tag{1455}$$

But wait! This integral looks divergent! Indeed it is, since every point in space has a nonzero energy $\frac{1}{2}\hbar\omega_{\mathbf{k}}$, and therefore if we integrate this over *all space*, the integral becomes infinite and unphysical. Thus, we usually just ignore this integral by redefining the energy of a state as the energy *above the ground state*. By this redefinition, the ground state would then be at $E_0 = 0$, and the successive excited states would have non-infinite energies $E_n > 0$.

However, if we *wanted*, we could restrict our integral over a particular region, such as a cubical box of side length L. Then, assuming that the electromagnetic field is restricted to one wavelength within the box, the integral could be evaluated, and we get a reasonable value for the energy:

$$\langle E_0 \rangle = \langle 0| \left(-\frac{1}{2}\int d^3x\, \hbar\omega_{\mathbf{k}}\right) |0\rangle \tag{1456}$$
$$= \langle 0| \left(-\frac{1}{2}\int_0^L \int_0^L \int_0^L dx\, dy\, dz\, \hbar\omega_{\mathbf{k}}\right) |0\rangle \tag{1457}$$
$$= -\langle 0| \frac{L^3\hbar\omega_{\mathbf{k}}}{2} |0\rangle \tag{1458}$$
$$= -\frac{L^3\hbar\omega_{\mathbf{k}}}{2}\langle 0|0\rangle \tag{1459}$$
$$= -\frac{L^3\hbar\omega_{\mathbf{k}}}{2} \tag{1460}$$

This is exactly what happens in the **Casimir effect**, an effect that arises due to the nonzero ground-state energy of the quantum electrodynamical vacuum, which leads to a force between two metal plates brought close together. Within finite distances and volumes, it is indeed possible to calculate the ground-state energy, and this is essential to many quantum effects.

Gauge coupling Unlike the quantum electrodynamical field $\hat{A}_\mu$, which reduces to the *classical* electromagnetic 4-potential (field) A_μ in the classical limit, the quantum electron field ψ has no classical field it reduces to. In the classical picture, A_μ satisfies a wave equation, and therefore describes (among other things) electromagnetic waves; light is an electromagnetic wave and can be described by A_μ. But in the classical picture, there is no analog for ψ; an electron is just a particle, it isn't a field. In fact, *only* massless particles with integer spin are associated with *classical fields* (the only two particles that satisfy this requirement are the photon and graviton, but the latter has yet to be experimentally confirmed and remains purely theoretical for now).

QED correction to the Coulomb potential Uehling potential

Multi-state systems with the quantized electromagnetic field Derive the Einstein A and B coefficients for spontaneous and stimulated emission.

The nonrelativistic and classical limit

Other corrections See https://books.google.com/books?id=nxz2CAAAQBAJ&lpg=PA103&pg=PA99#v=
This guide is based on:

- https://www.phys.ksu.edu/personal/wysin/notes/quantumEM.pdf
- https://www.damtp.cam.ac.uk/user/tong/qft/six.pdf
- https://www.damtp.cam.ac.uk/user/tong/aqm/aqmeight.pdf

0.3.5 Theoretical topics overview

By applying theoretical physics under the most extreme conditions, it may be possible to realize certain technologies that seem unimaginable today, including warp drives, wormholes, and vacuum energy extraction. At the moment, these are highly-speculative concepts, but they may lead to technologies that can be built by future generations. Thus, in the spirit of sharing and preserving knowledge, we believe that a discussion of even the most speculative concepts is merited.

The Alcubierre Metric

Interstellar travel is slow, because the distances between the stars are so long. Even our nearest star, Proxima Centauri, is over 4 light-years away - meaning that light, the fastest thing in the universe, would take more than 4 years to travel there. Our fastest spacecraft today would take hundreds of thousands of years to arrive at Proxima Centauri.

But what if we could take a ride on spacetime itself? We know from cosmological research that spacetime isn't itself subject to the cosmic speed limit, c: in fact, the most distant galaxies are receding from us faster than the speed of light. In theory, a spacetime geometry (albeit one unlike anything we see naturally in the universe) could allow for the possibility of faster-than-light interstellar travel. One such metric is the **Alcubierre metric**. By causing the expansion of spacetime in front of, and the contraction of spacetime behind, an isolated "shell region" of spacetime, the metric allows the "shell region" to move at an arbritrary speed. We will explore the mathematics of the Alcubierre metric in this chapter: both as a way to better understand general relativity, and to answer the question: is it possible to build a warp drive in reality?

Derivation of the Alcubierre Metric It is important to note here that this derivation is not a "derivation" in the strictest sense. This is because the Alcubierre Metric is more so a constructed rather than derived metric found by solving the Einstein Field Equations for a specific stress-energy tensor.

Alcubierre constructed his metric from the general form of all metrics in the ADM formalism:

$$ds^2 = -(\alpha^2 - \beta_i\beta^i)dt^2 + 2\beta_i dx^i dt + \gamma_{ij}dx^i dx^j \tag{1461}$$

Then, by choosing $\alpha = 1$, $\beta^x = -v_s f(r_s)$, where v_s is the ship speed, $\beta^y = \beta^z = 0$, and setting γ_{ij} to the Euclidean metric tensor δ_{ij}, we arrive at:

$$ds^2 = -dt^2 + (dx - v_s f(r_s)dt)^2 + dy^2 + dz^2 \tag{1462}$$

Where $x_s(t)$ is a function of the spacecraft's position over time, and $f(r_s(t))$ is a "top hat" shaping function to modify the metric such that it would vanish where $r_s > R$, and "push" the spacecraft forward:

$$f(r_s(t)) = \frac{\tanh\sigma(r_s(t) + R) - \tanh\sigma(r_s(t) - R)}{2\tanh\sigma R} \tag{1463}$$

$$r_s(t) = \sqrt{(x - x_s(t))^2 + y^2 + z^2} \tag{1464}$$

$$v_s(t) = \frac{dx_s(t)}{dt} \tag{1465}$$

Calculations using the Alcubierre metric We start with the general form of the Alcubierre metric, as it is typically presented in papers:

$$ds^2 = -dt^2 + (dx - v_s f(r_s)dt)^2 + dy^2 + dz^2 \tag{1466}$$

If we expand out the second term, we have:

$$ds^2 = -dt^2 + dx^2 - 2v_s f(r_s)dxdt + (v^2 f(r_s)^2)dt^2 + dy^2 + dz^2 \tag{1467}$$

We can rearrange to find:

$$ds^2 = (-dt^2 + v^2 f(r_s)^2 dt^2) - 2v_s f(r_s)dxdt + dx^2 + dy^2 + dz^2 \tag{1468}$$

Now, we can factor the first term to get:

$$ds^2 = -(1 - v^2 f(r_s)^2)dt^2 - 2v_s f(r_s)dxdt + dx^2 + dy^2 + dz^2 \tag{1469}$$

But how do we write this in matrix form? Remember that the line element is given by:

$$ds^2 = g_{\mu\nu}dx^\mu dx^\nu \tag{1470}$$

Expanding the 16 terms of the summation for a 4D spacetime metric, we have:

$$ds^2 = g_{tt}dtdt + g_{tx}dtdx + g_{ty}dydt + \cdots + g_{xx}dxdx + g_{xt}dxdt + \ldots g_{yy}dydy + \ldots g_{zz}dzdz \tag{1471}$$

We can group the terms that have products of the same differentials together, using $dq^2 = dqdq$:

$$ds^2 = g_{tt}dt^2 + g_{tx}dtdx + g_{ty}dydt + \cdots + g_{xx}dx^2 + g_{xt}dxdt + \ldots g_{yy}dy^2 + \ldots g_{zz}dz^2 \tag{1472}$$

As the metric is symmetric, $g_{tx} = g_{xt}$, and naturally $dtdx = dxdt$, we can say that $g_{tx}dtdx + g_{xt}dxdt = 2g_{xt}dxdt$, so our sum simplifies to:

$$ds^2 = g_{tt}dt^2 + 2g_{xt}dxdt + \cdots + g_{xx}dx^2 + \cdots + g_{yy}dy^2 + \cdots + g_{zz}dz^2 \tag{1473}$$

Finally, as the Alcubierre metric has no other terms other than the ones shown explictly in the sum, we can set all other terms in the sum to zero, so:

$$ds^2 = g_{tt}dt^2 + 2g_{xt}dxdt + g_{xx}dx^2 + g_{yy}dy^2 + g_{zz}dz^2 \tag{1474}$$

Note how this perfectly corresponds term by term with:

$$ds^2 = -(1 - v^2 f(r_s)^2)dt^2 - 2v_s f(r_s)dxdt + dx^2 + dy^2 + dz^2 \tag{1475}$$

So from there, we can determine that:

$$g_{tt} = -(1 - v^2 f(r_s)^2) \tag{1476}$$

$$g_{xt} = g_{tx} = -v_s f(r_s) \tag{1477}$$

$$g_{xx} = 1 \tag{1478}$$

$$g_{yy} = 1 \tag{1479}$$

$$g_{zz} = 1 \tag{1480}$$

Which results in the following matrix form of the metric:

$$g_{\mu\nu} = \begin{bmatrix} -(1 - v^2 f(r_s)^2) & -v_s f(r_s) & 0 & 0 \\ -v_s f(r_s) & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1481}$$

From there, we can do our necessary calculations.

York Time and Energy Density Here, we'll derive the the spacetime expansion/contraction and energy density associated with the (original) Alcubierre metric. The magnitude of the expansion and contraction of space resultant from the metric is called the **York Time**. We can derive the York Time from the extrinsic curvature tensor, which is given by:

$$K_{ij} = \frac{1}{2}\left(\partial_i X_j + \partial_j X_i\right) \tag{1482}$$

The York Time is the trace (another word for contraction contraction) of the extrinsic curvature tensor K_{ij} is given by:

$$K = K^i{}_i = \partial_i X^i \tag{1483}$$

For the Alcubierre metric, this would be:

$$K = v_s \frac{x_s}{r_s} \frac{df(r_s)}{dr_s} \tag{1484}$$

The above expression for the York Time gives the magnitude of the spacetime expansion and contraction, and we will refer to it with θ from this point on:

$$\theta = v_s \frac{x_s}{r_s} \frac{df(r_s)}{dr_s} \tag{1485}$$

Below, a plot of the York time with $v_s = c$, $\sigma = 8$ and a 2-meter radius warp shell is shown:

```
import numpy as np
import matplotlib.pyplot as plt

def f_rs(r_s, sigma=8, R=2):
    return (np.tanh(sigma * (r_s + R)) - np.tanh(sigma * (r_s - R)))/(2 * np.tanh(sigma * R))

def df_rs(r_s, sigma=8, R=2):
    return (sigma * (np.tanh(sigma * (R - r_s)) ** 2 - np.tanh(sigma * (R + r_s)) ** 2)) / (2

def d_rs(x, rho, x_s=2.5):
    # rho is y and z "squashed together"
    return ((x - x_s)**2 + rho**2)**(1/2)

def theta(x, rho, x_s=2.5, v_s=1, sigma=8, R=2):
    drs = d_rs(x, rho, x_s)
    dfrs = df_rs(drs, sigma, R)
    return v_s * ((x - x_s) / drs) * dfrs

def alcubierre_plt(width, height, samples=160):
    x = np.linspace(1.0, 8.0, num=samples)
    p = np.linspace( -4.0, 4.0, num=samples)

    # Generate coordinate matrices from coordinate vectors.
    X, P = np.meshgrid(x, p)

    # Get york time
    Z = theta(X, P, x_s = 5)

    # Create the Figure.
    fig = plt.figure(figsize=(width, height))
    ax = plt.axes(projection='3d')
```

```
    # Set the angle of the camera
    ax.view_init(25, -45)

    # Add latex math labels.
    ax.set_xlabel(r'$x$')
    ax.set_ylabel(r'$\rho$')
    ax.set_zlabel(r'$\theta$')

    # Set the axis limits
    ax.set_xlim(1.0, 8.0)
    ax.set_ylim( -4, 4)
    ax.set_zlim( -4.2, 4.2)

    # Plot the Surface.
    ax.plot_wireframe(X, P, Z, rstride=2, cstride=2, linewidth=0.5, antialiased=True, color='g
    plt.show()

alcubierre_plt(12, 6)
```

Here, we can see that the metric induces a contraction of spacetime in front and an expansion of spacetime behind the spacecraft.

Now, we can calculate the energy density. Returning to the metric, we know that we can calculate the stress-energy tensor from the metric. Taking the first component of the stress-energy tensor - that is, T_{00} - yields the energy density:

$$T_{00} = \frac{1}{8\pi} G_{00} \tag{1486}$$

> **Note**
> Here $G = c = 1$, which is the standard when using the ADM formalism. Thus $\frac{c^4}{8\pi G} \Rightarrow \frac{1}{8\pi}$.

Doing the tedious calculations of the metric to find the Einstein tensor, from which we can find the stress-energy tensor, will result in:

$$T_{00} = -\frac{1}{8\pi} \frac{(v_s)^2 (y^2 + z^2)}{4(r_s)^2} \left(\frac{df(r_s)}{dr_s} \right)^2 \tag{1487}$$

> **Note**
> Due to the verbosity of calculations such as this for complex metrics such as the Alcubierre metric, this is a task best left for a computer to do. Recommended software libraries for computing tensors in General Relativity include EinsteinPy and GraviPy

The energy density distribution is orthogonal to the direction of the spacecraft's movement, as shown in the plot below:

```
def energy_density(x, rho, x_s=2.5, v_s=1, sigma=8, R=1):
    r_s = ((x - x_s)**2 + rho**2)**(1/2)
    drs = d_rs(x, rho, x_s)
    dfrs = df_rs(drs, sigma, R)
    return ( -1/(8 * np.pi)) * ((v_s ** 2 * rho ** 2)/(4 * r_s ** 2)) * ((dfrs / drs) ** 2)

def energy_density_plt(width, height, samples=160):
    x = np.linspace(1.0, 8.0, num=samples)
```

```
    p = np.linspace( -4.0, 4.0, num=samples)

    # Generate coordinate matrices from coordinate vectors.
    X, P = np.meshgrid(x, p)

    # Get york time
    Z = energy_density(X, P, x_s = 5)

    # Create the Figure.
    fig = plt.figure(figsize=(width, height))
    ax = plt.axes(projection='3d')

    # Set the angle of the camera
    ax.view_init(25, -45)

    # Add latex math labels.
    ax.set_xlabel(r'$x$')
    ax.set_ylabel(r'$\rho$')
    ax.set_zlabel(r'$T_{00}$')

    # Set the axis limits
    ax.set_xlim(1.0, 8.0)
    ax.set_ylim( -4, 4)
    ax.set_zlim( -4, 4)

    # Plot the Surface.
    ax.plot_surface(X, P, Z, alpha=1, cstride=2, rstride=2, linewidth=0.1, cmap=plt.cm.coolwar
    plt.show()

energy_density_plt(12, 6, samples=320)
```

The energy density graph, unfortunately, disguises perhaps *the* most important issue with the Alcubierre metric: energy requirements. I will spare a full calculation of the energy requirements, but past research has shown that a 100-meter radius warp shell would require a total negative energy of:

$$E \approx -6.2 \times 10^{62} \text{ kg} \tag{1488}$$

For perspective, let's consider an idealized version of the Casimir effect of quantum mechanics, which has been shown to produce negative energy densities in an experimental setting.

Given that a is the distance between the plates, we may calculate the force caused by the Casimir effect with:

$$F = -\frac{\hbar c \pi^2}{240} \frac{1}{a^4} \tag{1489}$$

While the Casimir effect is measured in N/m^2, this is equivalent to J/m^3, so the negative energy density $e_{\rho-}$ of two plates separated by a distance of 1 micrometer would be approximately equal to:

$$e_{\rho-} = -1.299 \times 10^{-3} \text{ kg} \tag{1490}$$

Thus, a 60+ order-of-magnitude reduction is necessary to allow a functioning Alcubierre warp shell to be built, even assuming a large number of Casimir cavities arrayed together on the spacecraft.

Vacuum energy extraction

Above the Schwinger limit, very intensive electric fields with field strengths 10^{18}V/m, the quantum electrodrodynamical vacuum breaks down, allowing matter to be produced directly from the vacuum - this is known as the Schwinger effect. While purely theoretical at the moment, very-high-power lasers currently being built may be able to exceed the Schwinger limit.

Any matter produced by the Schwinger effect is likely to be extremely miniscule in amount. However, any such matter would in theory gravitate, exerting what may be thought of as an "attractive pull" towards other matter, like a suction pump. This can be described as a negative gravitational pressure, and can be effectively modelled with a *negative* energy density in the stress-energy tensor. And by the Einstein equations $G_{\mu\nu} = \kappa T_{\mu\nu}$ this would mean a *negative* Einstein tensor, and thus a negative curvature of spactime. The physical effects of this scenario would be bizarre; instead of experiencing a attractive gravitational force, massive particles would experience a *repulsive* gravitational force. At the moment, this is purely theoretical; but exploiting this effect may make the Alcubierre Drive (previously discussed) possible, among other applications.

0.4 The administrator's guide

0.4.1 Guide to governance

Project Elara is intended to foster a free and open community, governed by open-source and democratic ideals. As part of this, Project Elara has a Charter, which can be read in full on the next chapter. The Charter is like a mini-constitution, and details the foundational principles and structure of the community.

To lead an organization with such lofty and ambitious goals, some form of self-governance is essential. Project Elara therefore has a leadership team, composed of an elected Project Head and Deputy Head, and appointed members, who serve under the authority of the elected leaders. The community, in the form of a General Assembly, makes policy; the leadership team implements it, and supervises the Project as a whole. Both have mechanisms to ensure their accountability to both the members of the community and the Charter itself.

The entire governance system is constructed such as to make abuses of power as unlikely as possible, and to ensure that the project remains free, independent, and democratic long into the future.

Amendment process for the Project Elara Charter

We know that the Project Elara Charter isn't perfect: so we've made sure that it can be amended (albeit not *too* easily, since it serves as the bedrock of the Project and shouldn't be changed lightly). The steps to amend the Charter are as follows:

- Make a branch of the website with `git checkout -b <amendment-branch-name>`
- Edit `content/charter.md` with your proposed amendment
- Submit a PR on the website repository with a title in the format `[AMENDMENT] name_of_your_amendment`, and add the `charter` role to it
- Members will discuss in the comments section on the PR
- The amendment may be passed either by consensus or by holding a plurality vote (again, on the comments section)
- Members can also decide to reject the amendment if there is a general consensus to not proceed with it or if the vote fails to reach a plurality
  - Please **do not** close any PRs with amendments otherwise!

> **Note:** This process does not apply retroactively for practical purposes.

0.4.2 Charter of Project Elara

> **Note**
> This Charter is *provisional* at the moment and will only come into force after ratification. In the meantime, changes *are* likely to be made to the Charter.

The Provisional Charter of Project Elara

The members of Project Elara, dedicated to the mission of advancing peaceful space-based energy, institute this Charter among ourselves, to guide us in our pursuit of this mission, until such a time comes when it may come into force.

1.1 All members of Project Elara (henceforth referred to simply as "the community") are free, equal, and have inviolable dignity. They have the responsibility to treat each other as equals and to act with conscience towards one another. There shall be no disparity in the treatment of any member in the community on any grounds. Any attempts to discriminate against any member of the community on the basis of identity, show favoritism, or deny equal protection under the Charter is unacceptable and will be subject to consequences. This article is **not to be removed, amended, or abridged for any reason**, and cannot be changed by a consensus or majority decision.

1.2 Members have the freedom to hold and express their opinions, or to openly express a disagreement with another's opinions, in a **respectful** manner. They also have the freedom to criticize, in public or in private, and to do so without retaliation, when this is done in **good faith**.

1.3 Members are to be treated with fairness and are guaranteed protection from harassment, intimidation, abuse, and assaults on themselves and their well-being. Every member of the community has an obligation to come to the defense of another when a member has been subject to such attacks, and to render reasonable support and solidarity. To fail to do so is subject to immediate review of wrongdoing and possible repercussions.

1.4 If a member has been suspected of wrongdoing, the member will be temporarily **stripped of their position** and **placed in quarantine** from the rest of the community. An open board of review will be established to examine the wrongdoing. A member cannot be subject to arbitrary accusations or baseless attacks on their honor and reputation, that attempts to circumvent this review process, or exact retribution. If a member is found *responsible* of wrongdoing, the quarantine may be **extended indefinitely** while the member's behavior is monitored and continuously reviewed. In the most serious of cases, the member may be permanently and irrevocably expelled.

1.5 Any board of review that does not abide by the standards and procedure set out in the Charter, or is found to be biased, corrupt, or inept, will result in the accused member absolved and an immediate review of the accuser and board of review. If found of complicity or miscarriage of due process, all members involved will be immediately stripped of their positions. In addition, any member of the community may request a re-review on the behalf of another, should they disagree with the outcome of a review of wrongdoing. If such a re-review finds that the accused member did not commit the wrongdoing in question, the accused member will be absolved.

1.6 The Charter takes precedence over all acts and decisions made in the Project, and may not be violated for **any** reason. If any member of the Project suspects a violation of the Charter, a report and investigation is to be started immediately, and a remedy to the violation promptly issued. As a last resort, members of the Project have an obligation to resist violation of the Charter and restore its provisions by any methods necessary.

1.7 Membership may be acquired by taking a pledge to protect, defend, and abide by the Charter, in the individual's own words, in oral or written form. This must be followed by a public statement to the same within a reasonable timeframe. After attaining membership, the individual is granted an official role within the Project. Full members are bound to their pledge and have full responsibility for their actions.

1.8 Non-members of the Project may work alongside the Project and receive **participant status**, without being required to join the Project, unless a consensus in the Project decides against their admittance on fair grounds. Non-members must be treated with respect within the Project and

receive rightful credit for their work. Non-members, however, do not have the same obligations as members do, and may choose to leave the Project at any time.

1.9 Positions of leadership within the Project are given only to volunteers who have taken the pledge to the Charter. Before attaining a position of leadership, volunteers must spend a period of no less than five years carrying out duties delegated to them by the incumbent leadership and by the community at large. A member may serve for a maximum of ten years in a leadership role, and **may not serve again**.

1.10 The Project may never be dissolved. Any attempt to rescind the Charter, dismantle the Project, or take away its independence will be treated as an attack on the Project, and all members of the Project are obligated to oppose any such action.

1.11 The enactment of this Charter is sufficient by a community consensus. From that point on, it will be considered in full force.

1.12 We declare upon ratification that the Charter binds us in perpetuity and take it upon ourselves to realize it to the fullest measure.

0.4.3 Vision of the community

Project Elara is not the work of one individual, but an entire community, and we want this community to be one we can be proud of. We want to create a community that is:

- Welcoming
- Kind
- Understanding
- Supportive
- Safe

We will explore each of these facets of a good community in detail in this chapter.

Welcoming

We want a community that reflects the diversity of all humanity, and allows everyone to have a voice and feel valued. Thus, we must stress that this community must be an **entirely** non-judgemental, non-competitive, and non-elitist place, where everyone is free to share their ideas without intimidation. We will treat everyone as a human being and we will do everything we can to give everyone a sense of belonging here. This means we will, in accordance with the Charter, not accept any harrassment and hate, nor tolerate any individuals or groups whose aim is to harrass, abuse, or harm anyone in any way.

Kind

We want a community that brings out the best in people. We ask that everyone act courteously, gently, and compassionately to others. Always try to act in good faith, and be considerate before saying or doing anything. Be sensitive to others, especially those with a different point of view. Try to help others and lift others up, instead of being angry, harsh, rash, or mean. Celebrate each other's accomplishments, and be happy for each other in the community. Be forgiving of accidental mistakes that happen. The world can always use a little more kindness.

And we ask that our members - especially our leaders - to value others as much as themselves. Yes, we all have our own wants and needs, but the community as a whole will benefit if we also make sure we consider the wants and needs of others. We ask that everyone go the extra mile in making someone else's lives better, and be dedicated to helping others. Our concern for each other and the world around us is the spirit that binds and inspires all of us.

Understanding

Everyone is different, and everyone is unique and special in their own way. We celebrate our differences, and we ask everyone to respect the differences. Remember that there may be different ways to do the same thing, and no one way is the "best" way. Don't try to force one way of life upon someone else, and try to see someone else's point of view. Incorporate others and don't exclude others just because they are different in some way.

Supportive

We want everyone to encourage and inspire others. It doesn't have to be much, but everyone can use a helping hand at some point or another. Be here for one another and care for each other. Let one member's support of another inspire another, who inspires another, who inspires countless more.

Safe

No community can be a true community without its members feeling safe, and thus, we will take every measure to protect the community. It goes without saying that threatening, hostile, or intentionally hurtful acts will not be tolerated in the community. We ask that if any members have disagreements, they settle their differences through the Charter Council rather than taking direct action themselves.

If anyone feels unsafe in the community, they are free to anonymously speak to any of the members of the leadership team, and we will make sure to secure the community.

0.5 Appendix

0.5.1 Contributions

At Project Elara, we have a policy that all contributions *must* be given proper credit for, no matter how small, and that any results that were independently found by two (or more) researchers will be given equal and shared credit. We therefore maintain this list of contributions and update it as frequently as possible.

Pre-2024

> **Note**
> As these contributions were prior to December 2024 (when this list was started) there may some inaccuracies in reporting contributions. Please report any missing or erroneous contributions to the Project.

2024

December *Contributor: Jacky Song.*

0.5.2 Acknowledgements

Countless authors, physicists, mentors, and online resources made this book possible, and this book was the product of collaboration between researchers and students around the world. The contributors to this book include:

- Jacky Song

The incredible book A Most Incomprehensible Thing provided great insights into General Relativity and gave the Elara Handbook its approach to detailed, step-by-step teaching. Khan Academy's excellent calculus and multivariable calculus courses helped tremendously. Grant Sanderson's YouTube channel 3blue1brown provided a lot of the guidance behind the linear algebra series. Eigenchris, Andrew Dotson and Physics with Elliot motivated a lot of the examples and explanatory derivations. Finally, we would like to thank Dr. Harold White of Limitless Space Institute and Andrew Bramante of Greenwich High School for their incredible wisdom and support. A partial list of other resources consulted can be found on the project website at https://elaraproject.github.io/acknowledgements.

The importance of openness and sharing knowledge in science cannot be overstated. This book is a testament to this fact.