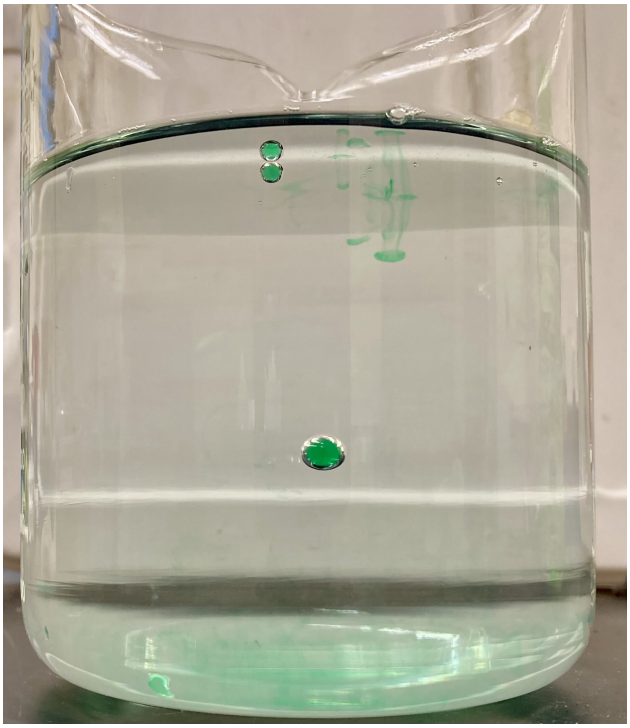
The Mathematical Language of Transport

A Primer on Mathematical Methods Important in Mastering Transport

David T. Leighton, Jr.

Department of Chemical & Biomolecular Engineering

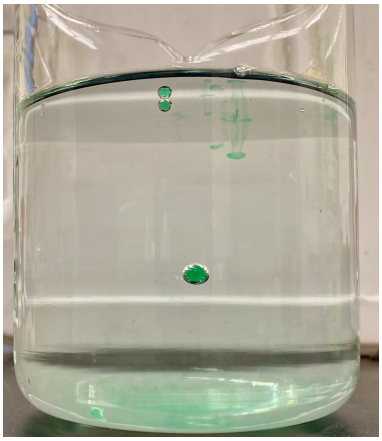
University of Notre Dame



Diffusive Press

https://www.diffusivepress.org

Published by Diffusive Press
https://www.diffusivepress.org



About the Cover: The front cover image shows an anti-bubble: a drop of dyed water surrounded by an air sheath. The density of the fluid inside the bubble is made to exactly balance the buoyancy of the air sheath, leaving the bubble neutrally buoyant. This image was captured during a demonstration from the companion volume, *A Teacher's Guide to Classroom Demonstrations in Transport Phenomena*, also available from Diffusive Press.

Introduction

Why Another Mathematics Text?

Transport phenomena is unusual among engineering courses in the breadth of mathematics it demands. A first course draws simultaneously on ordinary and vector calculus, ordinary and partial differential equations, and dimensional analysis. A second course expands on the application of all of these areas while considering more advanced techniques. Students arrive at these courses with widely varying preparation: some are taking differential equations for the first time alongside transport, others completed it a semester or a year earlier, and incoming graduate students range from those with strong mathematical backgrounds to those whose undergraduate programs emphasized process design and operations over rigorous mathematics. What nearly all of them share, regardless of background, is that the specific combination of tools transport requires — deriving a governing equation from a physical balance, scaling it to identify the dominant physics, imposing the right boundary conditions, and then solving it — is something they may not be adept at, even if the individual pieces are familiar. The difficulty is not usually that the mathematics is entirely new. It is that tools learned separately must suddenly be deployed together, in a specific sequence, on problems that begin as physical descriptions and end as differential equations.

This primer was written to help bridge that gap.

Mathematics as a Language

Transport phenomena is fundamentally concerned with rates of transport of momentum, heat, and mass. To describe these processes quantitatively, we require a mathematical language capable of expressing spatial variation, temporal change, and conservation principles.

That language consists largely of differential equations.

Whenever we perform a balance on a differential volume element and allow its dimensions to approach zero, derivatives inevitably appear. Conservation of mass leads to differential equations. Conservation of momentum leads to differential equations. Conservation of energy leads to differential equations. As a result, much of transport theory can be viewed as the formulation, interpretation, and solution of differential equations subject to appropriate boundary conditions. It is the detailed understanding of this mathematical language that allows the physics of transport to be expressed precisely.

Objectives of This Primer

The purpose of this primer is not to replace a mathematics textbook. Excellent references already exist for calculus, linear algebra, vector analysis, and differential equations. Nor is the goal to provide a rigorous development of the concepts of transport — excellent references exist for that as well.

Instead, the focus is on the specific mathematical techniques that arise repeatedly in transport phenomena and related engineering disciplines. The primer begins with material that all students will have encountered in their mathematics courses — differentiation, integration, the chain rule, integration by parts, and the basics of multivariable calculus — and proceeds through progressively more sophisticated techniques. While some are quite advanced (e.g., Morgan's Theorem and index notation), all are drawn from the mathematics employed in the Junior level Transport sequence taught at Notre Dame. The complete notes for these classes are available online at diffusivepress.org as are the notes which form the basis of this primer.

Several powerful mathematical techniques that appear in advanced transport analysis are intentionally omitted here. These include perturbation methods (both regular and singular), which are invaluable for problems with small parameters but require a separate treatment to use correctly; linear stability analysis, which examines when a base flow becomes unstable to small disturbances; and conformal mapping methods for potential flow in complex geometries. These are topics generally reserved for a graduate course in advanced transport or applied mathematics.

Topics covered in this primer include:

- Differential and integral calculus, including Taylor series expansions
- Analytical and transformative solution techniques for ordinary differential equations (ODEs)
- Computational methods for ODEs, including Runge-Kutta and shooting methods
- Vector calculus, index notation, and tensor symmetry arguments
- Generation of harmonic functions for 3D steady-state transport
- PDE simplification techniques, including dimensionless scaling and self-similarity
- Analytical PDE solutions via Sturm-Liouville theory and separation of variables
- Numerical and stochastic PDE methods, including forward marching and Monte Carlo simulations

The emphasis throughout is on practical application. Whenever possible, the mathematical methods are presented in a form that connects directly to the types of problems encountered in fluid mechanics, heat transfer, mass transfer, and reaction engineering. Each chapter includes worked examples and practice problems with full solutions.

Intended Audience

This primer was written primarily for students studying transport phenomena, whether at the advanced undergraduate or introductory graduate level.

Some readers may use the material as a refresher before taking a transport course. Others may use it as a companion reference while working through fluid mechanics, heat transfer, or mass transfer problems. Still others may simply wish to strengthen mathematical skills that have become rusty with time. This would be particularly true for incoming first-year graduate students seeking to prepare themselves for the mathematical rigor of a graduate curriculum.

Regardless of background, the objective is the same: to develop familiarity with the mathematical tools most commonly used in transport analysis, so that attention can be focused on connecting the mathematics to the underlying physical phenomena.

Mathematics is the language in which transport phenomena is written. Like any language, fluency comes through practice. The chapters that follow are intended to provide that practice and to serve as a useful reference whenever these tools are needed.

How to Use This Primer

There are many ways of using this primer. The simplest is to look through the description and examples on a particular topic of interest and then test your knowledge by solving the sample problems - without looking at the answers first, of course. If you have no difficulties, then simply move on to the next topic. If you find a particular topic challenging, however, more work is required. The simplest way of getting more information on a topic is to copy and paste the part of the PDF you find puzzling into an AI and ask for a more detailed explanation, describing what your problem with it is. A typical prompt might be:

> *The following is in a text. Can you please explain it in more detail, at a level appropriate to a chemical engineering student entering their junior year? The text is:*

The explanations you obtain in this manner are usually quite detailed: after all, advanced AI models are *trained* on the classic mathematics and transport texts, and thus have access to a wide variety of information! After making progress in understanding, copy and paste the problem that was giving you difficulty into an AI and ask it for another problem which covers the same technique - without solving it. Such a prompt might be:

> *I need practice with this technique. Please give me a problem which illustrates this method. Don't solve it yet, I'll ask you to do that after I work it out to compare answers.*

Once you have solved the problem yourself, ask the AI to solve it for you and compare the answers. In this manner all of these techniques should be rapidly absorbed.

If an AI is able to provide you with problems and solve them using the mathematical techniques described in this text (that it is also quite good at explaining), a straightforward question to ask is "Why should I devote the time necessary to learn the math at all?" This is, of course, a bit of a philosophical question, and there are a number of possible answers. For the student, an obvious reason for learning the math is surviving closed book exams. Research has shown that using an AI to solve problems actually results in "Cognitive Offloading" in which studying an AI's solution to a problem makes you *think* you understand a method, but then find yourself unable to use the technique when confronted with a problem even slightly different than the one solved - very frustrating both for the student and the instructor grading the exam! This is somewhat akin to the difference between

riding an e-bike vs. a regular one - both will get you to your destination, but if you haven't built up the muscles to ride a regular bike, you will find that it is very challenging when your e-bike battery runs down on an uphill grade. Solving the problems yourself and *only then* checking your answers with the AI is an effective (if sometimes painful) way of building up your "mental muscles".

For a practicing engineer whose goal is *productivity* rather than only learning, mastering the mathematical techniques still has value even if the AI is always there. That is because a fundamental understanding of how you can formulate, scale, simplify, and solve problems enables you to use the AI as an *assistant* much more effectively. You will be better able to guide the AI into using the right solution approach and identify the inevitable errors your assistant will make - which usually occur at the worst possible moment. At least for now, humans are better at executive function than is an AI, and thus you need to know enough to be able to guide the AI to the correct result.

The last part of this primer contains some numerical examples with coding. The codes are written in MATLAB, however any AI can translate them into Python or other languages, generally without error. With the availability of AI it is no longer necessary to be an expert at coding and syntax yourself - but it is very helpful to understand the basic concepts. Again, paste in one of the sample codes and ask the AI to explain the algorithm, suggest improvements, and apply it to a problem of interest to you. AI tools are *very* good at this, however your understanding of the logic of the algorithm generally enables your AI assistant to produce a superior result.

Acknowledgements

The material contained in this primer reflects the influence of the many teachers, colleagues, and students encountered over a career spanning four decades of teaching transport phenomena at the University of Notre Dame.

Among my own teachers, three had a particularly significant impact on my understanding of the subject. William Russel first taught me heat and mass transfer at Princeton. While I had already completed a course in fluid mechanics, it was through his presentation of transport phenomena that many of the underlying concepts finally came together in a coherent way. At Stanford, Milton Van Dyke introduced me to the power of scaling, similitude, and perturbation methods, demonstrating how seemingly complex problems can often be understood through careful physical reasoning. Most significant of all was Andreas Acrivos, my Ph.D. advisor, whose insight into fluid mechanics, asymptotic methods, and low Reynolds number hydrodynamics continues to shape much of my professional thinking.

I am also indebted to the many colleagues with whom I have had the privilege of interacting over the years, and to the generations of students who have taken my transport courses at Notre Dame. Their questions, confusions, and occasional skepticism continually revealed which explanations were effective and which required improvement. Many of the examples and perspectives presented here were refined through those interactions.

This primer began as handwritten course materials developed and modified over many years of teaching. The conversion of those notes into LaTeX was greatly accelerated through the use of modern AI-based transcription and editing tools. AI was also used for generating drafts of many of the extra practice problems and solutions, all of which were reviewed and verified before inclusion. Responsibility for any remaining errors is, of course, entirely my own.

Finally, I would like to thank all of the students who make the effort to learn this material. Transport phenomena has a reputation for being difficult, and often deservedly so. Yet it remains one of the most powerful, useful, and beautiful subjects in engineering. If this primer helps make that journey a little easier, then it has served its purpose.

Contents

Part I

Foundations of 1D Calculus & ODEs

1 Differentiation

We shall begin with the fundamental operations of calculus. Much of engineering (and transport) deals with how things change with position or time. This is captured by the derivative. Suppose we have a function:

$$y = f(x) \tag{1.1}$$

$$\frac{dy}{dx} = \frac{df}{dx}$$

This represents the local slope of the tangent to $f(x)$.

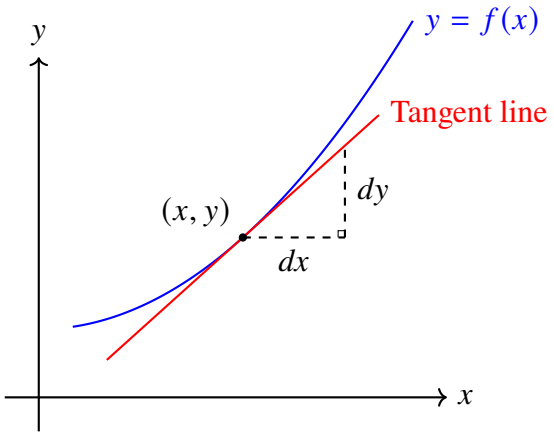


Figure 1.1: Geometric interpretation of the derivative as the local slope ($\frac{dy}{dx}$) of the tangent line to the curve $y = f(x)$.

We also need higher derivatives:

$$\frac{d^2 f}{dx^2} \equiv \frac{d}{dx}\left(\frac{df}{dx}\right)$$

Note the units: it is not $\frac{[f]^2}{[x]^2}$!!! (I have seen this error many times on exam papers...) The correct units are:

$$\left[\frac{d^2 f}{dx^2}\right] \equiv \frac{[f]}{[x]^2}$$

Also note:

- $f'' < 0$ = concave down
- $f'' > 0$ = concave up
- $f'' = 0$ = inflection point (usually - necessary but not sufficient)

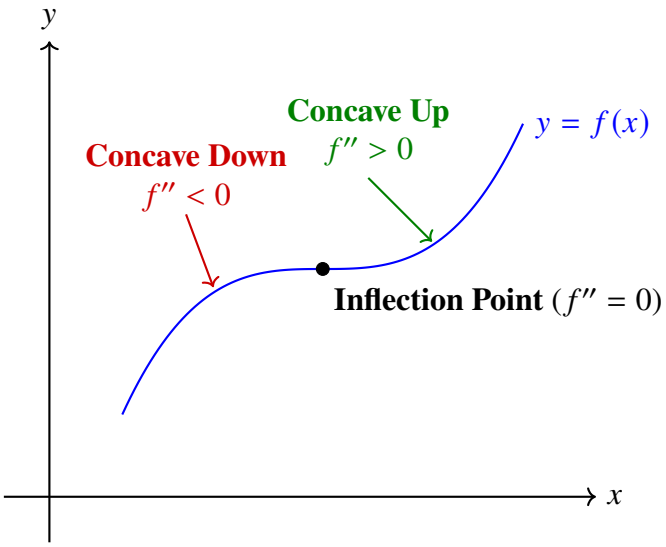


Figure 1.2: Visual representation of function concavity. The curve transitions from concave down ($f'' < 0$) to concave up ($f'' > 0$) at the inflection point ($f'' = 0$).

Common Derivatives

It is very convenient to know common derivatives without having to look them up. Probably the most often encountered ones are:

$$\frac{d}{dx}(x^a) = ax^{a-1}$$
$$\frac{d}{dx}(\ln x) = \frac{1}{x}$$
$$\frac{d}{dx}(e^{ax}) = a \cdot e^{ax}$$
$$\frac{d}{dx}(\sin x) = \cos x$$
$$\frac{d}{dx}(\cos x) = -\sin x$$

etc. . .

Chain Rule Differentiation

Often we need to do chain rule differentiation: Suppose $y = f(g(x))$ where f and g are functions.

$$\frac{dy}{dx} = \frac{df}{dg}\frac{dg}{dx}$$

Let's look at an example: $f = e^{ax^2}$ (which is $f = e^{ag}$, where $g = x^2$)

So,

$$\begin{aligned}\frac{df}{dx} &= (ae^{ax^2})(2x) \\ &= 2ax \cdot e^{ax^2}\end{aligned}$$

The Product Rule

This is the differentiation analogue of integration by parts.

If $y = f(x)g(x)$ we have the derivative:

$$\frac{dy}{dx} = g\frac{df}{dx} + f\frac{dg}{dx}$$

Example:

$$\frac{d}{dx}(\tan x) = \frac{d}{dx}\left(\frac{\sin x}{\cos x}\right)$$

Using the product rule on $\sin x \cdot (\cos x)^{-1}$:

$$= \frac{\cos x}{\cos x} + \sin x \left(\frac{-1}{\cos^2 x}\right)(-\sin x)$$

$$= 1 + \frac{\sin^2 x}{\cos^2 x} = \frac{\cos^2 x + \sin^2 x}{\cos^2 x}$$

$$= \frac{1}{\cos^2 x} = \sec^2 x$$

The Quotient Rule

The product rule can also be applied to quotients by rewriting the denominator with a negative exponent. Suppose:

$$y = \frac{f(x)}{g(x)} = f(x)\,(g(x))^{-1}$$

Applying the product rule and chain rule:

$$\frac{dy}{dx} = \frac{df}{dx}\frac{1}{g} + f\left(-\frac{1}{g^2}\right)\frac{dg}{dx}$$

Combining terms:

$$\frac{dy}{dx} = \frac{g\dfrac{df}{dx} - f\dfrac{dg}{dx}}{g^2}$$

So the quotient rule is:

$$\boxed{\frac{d}{dx}\left(\frac{f}{g}\right) = \frac{gf' - fg'}{g^2}}$$

Example:

$$\frac{d}{dx}\left(\frac{x^2+1}{x}\right) = \frac{x(2x) - (x^2+1)(1)}{x^2}$$

$$= \frac{x^2 - 1}{x^2}$$

Partial Derivatives

We also have partial derivatives for functions of more than one independent variable.

Let's take $h = f(x, y)$ (say, height as a function of position).

$$\frac{\partial h}{\partial x} \equiv \frac{\partial f}{\partial x}$$
$$\frac{\partial h}{\partial y} \equiv \frac{\partial f}{\partial y}$$

Again, these are slopes of a tangent - but in the x and y direction!

Suppose $h = x^2 + y^2$:

$$\frac{\partial h}{\partial x} = 2x, \quad \frac{\partial h}{\partial y} = 2y$$

We hold one variable constant while taking the derivative with respect to the other!

Suppose $h = x^2y^3$. Then:

$$\frac{\partial h}{\partial x} = 2xy^3$$
$$\frac{\partial h}{\partial y} = 3x^2y^2$$

Application to Transport

All this is often used together in transport. A classic problem is the temperature distribution in a fluid above a heated plate where the fluid flows past as a simple shear flow.

Solving the boundary layer problem you will find $T = x^{1/3} f(\eta)$ where the similarity variable $\eta = \frac{y}{x^{1/3}}$.

The heat flux boundary condition requires evaluating the derivative $\dfrac{\partial T}{\partial y}$ at the plate.

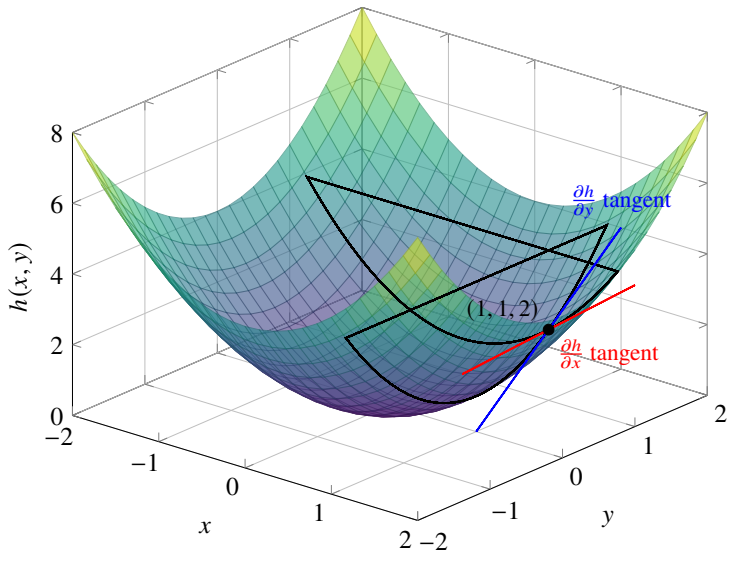


Figure 1.3: Visualizing partial derivatives on the surface $h(x, y) = x^2 + y^2$. The partial derivative $\frac{\partial h}{\partial x}$ is the slope of the tangent line in the x-direction (holding y constant), while $\frac{\partial h}{\partial y}$ is the slope of the tangent line in the y-direction (holding x constant).

Keeping x constant with respect to y (orthogonal coordinates):

$$\frac{\partial T}{\partial y} = \frac{\partial}{\partial y}(x^{1/3} f(\eta)) = x^{1/3} \frac{\partial f}{\partial y}$$

Using the chain rule:

$$= x^{1/3} \frac{df}{d\eta} \frac{\partial \eta}{\partial y} = x^{1/3} f' \left(\frac{1}{x^{1/3}} \right)$$

$$= f'$$

So the constant heat flux boundary condition at the plate $y = 0$ reduces to $f'(0) = C$.

We also need (for this problem) $\frac{\partial T}{\partial x}$:

$$\frac{\partial T}{\partial x} = \frac{\partial}{\partial x}(x^{1/3} f(\eta)) = \frac{1}{3} x^{-2/3} f + x^{1/3} \frac{\partial f}{\partial x}$$

Using the product rule and the chain rule:

$$= \frac{1}{3} x^{-2/3} f + x^{1/3} \frac{df}{d\eta} \frac{\partial \eta}{\partial x}$$

$$= \frac{1}{3} x^{-2/3} f + x^{1/3} f' \left(-\frac{1}{3} y x^{-4/3} \right)$$

$$= \frac{1}{3} x^{-2/3} f - \frac{1}{3} x^{-2/3} \left(\frac{y}{x^{1/3}} \right) f'$$

$$= \frac{1}{3}x^{-2/3}(f - \eta f')$$

These terms (and further derivatives) are combined to reduce the second order partial differential equation governing convective heat transfer in this problem into a simple ordinary differential equation that can be solved analytically - a dramatic simplification. That's just one example of why it is important to be able to use the rules of calculus to manipulate the terms in an equation!

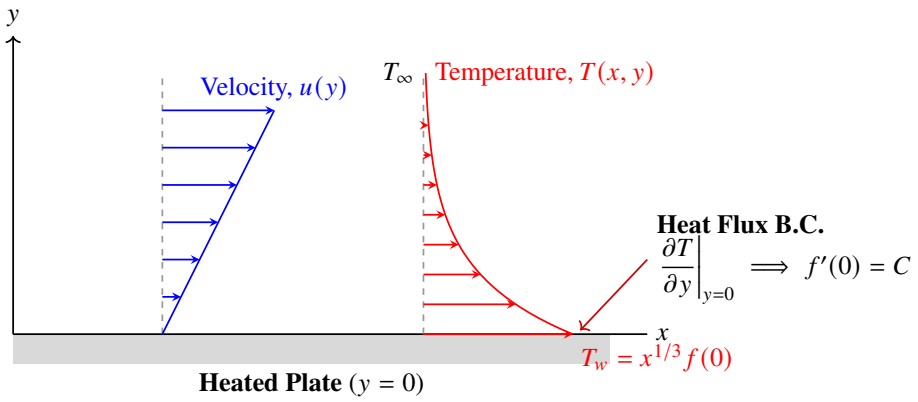


Figure 1.4: Schematic of convective heat transfer in a simple shear flow over a heated flat plate. The velocity profile $u(y)$ increases linearly, while the thermal boundary layer $T(x, y)$ decays from the wall temperature $T_w = x^{1/3} f(0)$ at the plate ($y = 0$) to the free-stream temperature $T_\infty = 0$.

Common Mistakes

- Forgetting the chain-rule factor
- Forgetting the negative sign when differentiating $1/x$
- Forgetting which variables are constant
- Confusing ∂ and d

Practice Problems

Test your skills on these problems before moving on. In transport and thermodynamics, recognizing which variables are constants is half the battle!

Problem 1: The Damped Oscillator

A classic first-year physics problem describes a mass on a spring losing energy over time. The position is given by:

$$x(t) = Ae^{-bt}\cos(\omega t)$$

Find the velocity $v(t) = \dfrac{dx}{dt}$. (Assume A, b, and ω are constants).

Problem 2: The Arrhenius Equation

The rate constant k for a chemical reaction depends on temperature T according to:

$$k = A \exp\left(-\frac{E_a}{RT}\right)$$

Find how the rate constant changes with temperature, $\frac{dk}{dT}$. (Assume A, E_a, and R are constants).

Problem 3: The Van der Waals Gas

The pressure of a real gas is given by the equation of state:

$$P = \frac{RT}{v - b} - \frac{a}{v^2}$$

Find the partial derivatives $\frac{\partial P}{\partial T}$ and $\frac{\partial P}{\partial v}$. (Assume R, a, and b are constants).

Problem 4: Spot the Mistake

A student is asked to find the spatial derivative of a basic 1D Gaussian distribution (often used in random walk or basic diffusion models): $f(x) = C_0 e^{-ax^2}$. The student writes the final answer as:

$$\frac{df}{dx} = -ax \cdot C_0 e^{-ax^2}$$

Identify and correct the student's mistake.

Problem 5: Similarity Variables and Differentiation

A basic 1D mass diffusion profile (like a drop of ink spreading in a tube of water) can be modeled as $C(x,t) = C_0 e^{-\eta^2}$, where the combined variable is $\eta = \frac{x}{\sqrt{4Dt}}$. Using the chain rule, find the spatial derivative $\frac{\partial C}{\partial x}$. (Treat time t and the diffusion coefficient D as constants).

Solutions to Practice Problems

Solution 1: The Damped Oscillator

You must use the product rule ($uv' + u'v$) and the chain rule. Let $u = Ae^{-bt}$ and $v = \cos(\omega t)$.

$$u' = -Abe^{-bt}$$
$$v' = -\omega \sin(\omega t)$$

Combining these:

$$v(t) = \frac{dx}{dt} = -Abe^{-bt}\cos(\omega t) - A\omega e^{-bt}\sin(\omega t)$$

Note: You can factor out the $-Ae^{-bt}$ to make this cleaner, but the calculus is done!

Solution 2: The Arrhenius Equation

This requires the chain rule. The most common mistake here is dropping a negative sign or forgetting how to differentiate T^{-1}. Let the inner function be $g = -\frac{E_a}{R}T^{-1}$. The derivative is $\frac{dg}{dT} = \frac{E_a}{R}T^{-2} = \frac{E_a}{RT^2}$.

$$\frac{dk}{dT} = A\exp\left(-\frac{E_a}{RT}\right)\cdot\left(\frac{E_a}{RT^2}\right)$$

Solution 3: The Van der Waals Gas

For $\frac{\partial P}{\partial T}$, treat specific volume v as a constant. The second term ($-\frac{a}{v^2}$) has no T in it, so its derivative is zero:

$$\frac{\partial P}{\partial T} = \frac{R}{v-b}$$

For $\frac{\partial P}{\partial v}$, treat T as a constant. Rewrite the equation mentally as $P = RT(v-b)^{-1} - a(v)^{-2}$ to make the power rule easier:

$$\frac{\partial P}{\partial v} = -RT(v-b)^{-2} - a(-2)(v)^{-3} = -\frac{RT}{(v-b)^2} + \frac{2a}{v^3}$$

Solution 4: Spot the Mistake

The student forgot the factor of 2 when applying the chain rule to the exponent. The inner function is $g(x) = -ax^2$. Its derivative is $\frac{dg}{dx} = -2ax$. The correct answer is:

$$\frac{df}{dx} = -2ax \cdot C_0 e^{-ax^2}$$

Solution 5: Similarity Variables and Differentiation

This is exactly how you will handle similarity variables in transport! Using the chain rule: $\frac{\partial C}{\partial x} = \frac{dC}{d\eta}\frac{\partial \eta}{\partial x}$.

$$\frac{dC}{d\eta} = -2\eta C_0 e^{-\eta^2}$$
$$\frac{\partial \eta}{\partial x} = \frac{1}{\sqrt{4Dt}}$$

Multiplying them together:

$$\frac{\partial C}{\partial x} = -2\eta C_0 e^{-\eta^2}\left(\frac{1}{\sqrt{4Dt}}\right)$$

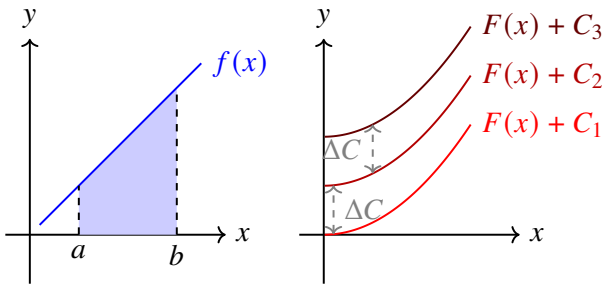
Note: If you substitute η back in, you get $-\frac{2x}{4Dt}C_0 e^{-\eta^2}$, which simplifies to $-\frac{x}{2Dt}C_0 e^{-\eta^2}$.

2 Integration

Our second topic is, naturally enough, integration.

- This is the "anti-Derivative" inverse operation.
- In one-D, it's just the area under the curve!

$$\int f(x)dx$$



$\int_a^b f(x)\,dx$ $\int f(x)\,dx = F(x) + C$

Figure 2.1: Comparison of definite and indefinite integrals. **Left** illustrates the definite integral as a specific value representing the physical area under the curve between boundaries a and b. **Right** illustrates the indefinite integral, which produces a family of parallel antiderivative functions of a line of slope 1 offset by the baseline constant C.

We have definite integrals, where explicit limits are specified and the result is a number (or possibly a function if one limit is variable), and indefinite integrals, which produce a family of antiderivatives differing by a constant of integration.

This constant represents the unknown baseline of the system. In transport problems, this is usually found by applying an initial condition (like the concentration at t=0) or a boundary condition (like the velocity at a wall).

So:

$$\int_0^1 x\,dx = \frac{1}{2}x^2\Big|_0^1 = \frac{1}{2} \quad \text{(Definite integral)}$$

$$\int_0^x x'\,dx' = \frac{1}{2}x'^2\Big|_0^x = \frac{1}{2}x^2 - \frac{1}{2}(0) = \frac{1}{2}x^2 \quad \text{(Note dummy variable)}$$

$$\int x\,dx = \frac{1}{2}x^2 + C \quad \text{(Note constant of integration)}$$

Common Integrals

Some integrals you should know:

$$\int e^{ax}\,dx = \frac{1}{a}e^{ax}$$
$$\int \sin x\,dx = -\cos x$$
$$\int x^a\,dx = \frac{1}{a+1}x^{a+1} \quad (a \neq -1)$$
$$\int \frac{1}{x}\,dx = \ln x \quad (x > 0)$$
$$\int_{-\infty}^{\infty} e^{-x^2}\,dx = \sqrt{\pi}$$

Note:

$$\int e^{-x^2}\,dx = \frac{\sqrt{\pi}}{2}\operatorname{erf}(x) + C$$

where $\operatorname{erf}(x)$ is the error function. This is usually looked up rather than memorized!

Integration by Guessing

Often you can figure out an integral by guessing from the derivative of related functions. Suppose we want $\int \ln x\,dx$. We expect this to be related to $x \ln x$ (just increasing the power of x by 1 as would occur integrating polynomials).

So:

$$\frac{d}{dx}(x \ln x) = \ln x + x\left(\frac{1}{x}\right) = \ln x + 1 \quad \text{(by product rule)}$$

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

So,

$$\int \ln x\,dx = x \ln x - \int dx = x \ln x - x + C$$

Integration by Parts

In general, we obtain integration by parts directly from the product rule:

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

Integrating both sides:

$$\int \frac{d}{dx}(fg)\,dx = \int f\frac{dg}{dx}\,dx + \int g\frac{df}{dx}\,dx$$

Since integration and differentiation are inverse operations:

$$fg = \int fg'\,dx + \int gf'\,dx$$

Rearranging:

$$\boxed{\int fg'\,dx = fg - \int gf'\,dx}$$

Note: In general you choose what part of the integrand to be f and what part g' in order to simplify things. Choose f so that it simplifies on differentiation and g' so that it simplifies on integration!

Integration by parts is often used in deriving transport relationships ranging from the Minimum Dissipation Theorem (a fundamental result in low Reynolds number flows) to the von Karman integral momentum balance (a fundamental result in boundary layer flow at high Reynolds numbers). It is a *very* useful result to remember!

Differentiation of an Integral with Variable Limits

Often in engineering (such as dealing with moving boundaries, boundary layer growth, or control volume balances), you will need to take the derivative of an integral where the limits themselves are functions of the variable you are differentiating against.

Even though the limit is a variable, this is still a **definite integral** because it has explicit bounds.

If the upper limit is a function of x (say, $g(x)$), and the lower limit is a constant a, we combine the Fundamental Theorem of Calculus with the chain rule:

$$\frac{d}{dx}\int_a^{g(x)} f(t)\,dt = f(g(x))\frac{dg}{dx}$$

Notice the use of the dummy variable t inside the integral!

If **both** limits are functions of x, we use the 1D **Leibniz Integral Rule**:

$$\frac{d}{dx}\int_{h(x)}^{g(x)} f(t)\,dt = f(g(x))\frac{dg}{dx} - f(h(x))\frac{dh}{dx}$$

In the case that the integrand is a function of both x and t, we use the **General Leibniz Integral Rule**:

$$\frac{d}{dx}\int_{h(x)}^{g(x)} f(x,t)\,dt = f(x,g(x))\frac{dg}{dx} - f(x,h(x))\frac{dh}{dx} + \int_{h(x)}^{g(x)} \frac{\partial f}{\partial x}(x,t)\,dt$$

Example: Suppose we have a profile defined by an integral (similar to the error function used in transient diffusion):

$$y(x) = \int_0^{x^2} e^{-t^2}\,dt$$

To find the gradient $\dfrac{dy}{dx}$, we apply the rule where the inner function is $f(t) = e^{-t^2}$ and the upper limit is $g(x) = x^2$:

$$\begin{aligned}\frac{dy}{dx} &= e^{-(x^2)^2}\cdot\frac{d}{dx}(x^2)\\ &= e^{-x^4}\cdot(2x)\\ &= 2xe^{-x^4}\end{aligned}$$

Multiple Integration

The analog to partial differentiation is multiple integration, usually over a surface or volume. Suppose we have some concentration distribution (mass/volume) $\phi(x,y,z)$ inside a rectangular prism volume of:

$$-a < x < a, \quad -b < y < b, \quad -c < z < c$$

The total amount in the volume:

$$I = \int_V \phi\,dV \equiv \int_{-a}^{a}\int_{-b}^{b}\int_{-c}^{c} \phi\,dz\,dy\,dx$$

Often shapes are more complicated! Suppose instead the shape is a rectangular pyramid of height h:

$$0 < z < h; \quad -a\left(1-\frac{z}{h}\right) < x < a\left(1-\frac{z}{h}\right); \quad -b\left(1-\frac{z}{h}\right) < y < b\left(1-\frac{z}{h}\right)$$

In this example you integrate over x & y first as the limits leave a function of z (e.g., $f(z)$). Then you finish it off by integrating over z!

If $\phi = 1$ (uniform concentration), we can find the volume of the pyramid. Since the inner integrals (y and x) do not depend on each other, we can evaluate them sequentially, which will leave us with a function of z:

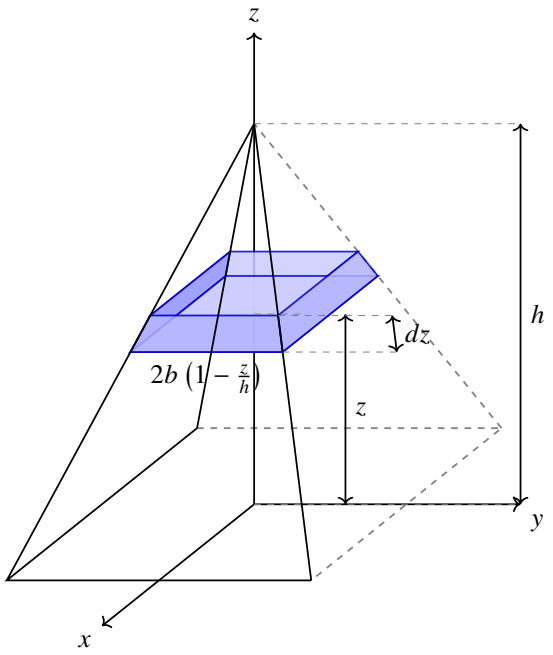


Figure 2.2: Integration volume of a rectangular pyramid. To set up the 3D limits, we consider a thin horizontal slice of thickness dz at an arbitrary height z. Because the geometry tapers linearly to the apex, the x and y bounds of this slice scale accordingly, yielding the side lengths $2a(1 - z/h)$ and $2b(1 - z/h)$.

$$\begin{aligned} V &\equiv \int_0^h \int_{-a(1-\frac{z}{h})}^{a(1-\frac{z}{h})} \int_{-b(1-\frac{z}{h})}^{b(1-\frac{z}{h})} dy\, dx\, dz \\ &= \int_0^h \int_{-a(1-\frac{z}{h})}^{a(1-\frac{z}{h})} [y]_{-b(1-z/h)}^{b(1-z/h)}\, dx\, dz \\ &= \int_0^h \int_{-a(1-\frac{z}{h})}^{a(1-\frac{z}{h})} 2b\left(1-\frac{z}{h}\right) dx\, dz \end{aligned}$$

Now, evaluating the x integral:

$$\begin{aligned} V &= \int_0^h \left[2b\left(1-\frac{z}{h}\right)x\right]_{-a(1-z/h)}^{a(1-z/h)} dz \\ &= \int_0^h 2b\left(1-\frac{z}{h}\right)\cdot 2a\left(1-\frac{z}{h}\right) dz \\ &= \int_0^h 4ab\left(1-\frac{z}{h}\right)^2 dz \end{aligned}$$

Notice that $4ab(1 - \frac{z}{h})^2$ is simply the cross-sectional area of the rectangular pyramid at any height z. Finally, evaluating the z integral (using u-substitution where $u = 1 - \frac{z}{h}$ and $dz = -h\, du$):

$$\begin{aligned} V &= 4ab\left[-\frac{h}{3}\left(1-\frac{z}{h}\right)^3\right]_0^h \\ &= 4ab\left(0-\left(-\frac{h}{3}\right)\right) \\ &= \frac{4}{3}abh \end{aligned}$$

This yields the standard geometric formula: $V = \frac{1}{3} \times$ (base area) $\times$ height, where the full base area is $(2a) \times (2b) = 4ab$.

Practice Problems

Test your skills on these problems. Remember to pay close attention to your dummy variables and the order of integration!

Problem 1: The Baseline (Initial Conditions)

A particle's acceleration is given by $a(t) = a_0 e^{-bt}$. Find the velocity $v(t)$ by integrating the acceleration. Use the initial condition $v(0) = 0$ to find your constant of integration. (Assume a_0 and b are constants).

Problem 2: Integration by Guessing

In the text, we found $\int \ln x \, dx$ by examining the derivative of $x \ln x$. Using a similar approach, find $\int xe^{ax} \, dx$. *Hint: Start by taking the derivative of xe^{ax} using the product rule, and then rearrange!*

Problem 3: The Leibniz Rule

You are analyzing the growth of a thermal boundary layer, and you encounter the following profile:

$$\theta(x) = \int_0^{x^3} e^{-2t} \, dt$$

Find the spatial derivative $\dfrac{d\theta}{dx}$.

Problem 4: Spot the Mistake

A student is asked to find the total distance traveled given a velocity $v = c_1 t$. The student writes:

$$x(t) = \int_0^t c_1 t \, dt = \frac{1}{2}c_1 t^2\Big|_0^t = \frac{1}{2}c_1 t^2$$

While the final algebraic answer happens to be correct, the notation contains a dangerous trap for transport problems. Identify the notation error and write the integral correctly.

Problem 5: The 2D Control Volume

A fluid flows through a parallel-plate channel. The concentration of a solute in a 2D cross-section of the channel is given by:

$$C(x, y) = C_0 \left(1 - \frac{y^2}{h^2}\right)$$

Find the total amount of solute M in the region defined by $0 < x < L$ and $-h < y < h$ by evaluating the double integral:

$$M = \int_0^L \int_{-h}^{h} C_0 \left(1 - \frac{y^2}{h^2}\right) \, dy \, dx$$

Solutions to Practice Problems

Solution 1: The Baseline (Initial Conditions)

Set up the indefinite integral:

$$v(t) = \int a_0 e^{-bt}\, dt = -\frac{a_0}{b} e^{-bt} + C$$

Now, apply the initial condition ($v = 0$ when $t = 0$):

$$0 = -\frac{a_0}{b} e^{-b(0)} + C$$
$$0 = -\frac{a_0}{b}(1) + C \quad \Rightarrow \quad C = \frac{a_0}{b}$$

Substitute C back into the velocity equation:

$$v(t) = -\frac{a_0}{b} e^{-bt} + \frac{a_0}{b} = \frac{a_0}{b}\left(1 - e^{-bt}\right)$$

Solution 2: Integration by Guessing

Following the hint, first find the derivative of xe^{ax} using the product rule:

$$\frac{d}{dx}\left(xe^{ax}\right) = e^{ax} + axe^{ax}$$

Now, integrate both sides with respect to x:

$$xe^{ax} = \int e^{ax}\, dx + \int axe^{ax}\, dx$$

Evaluate the first integral on the right ($\int e^{ax}\, dx = \frac{1}{a} e^{ax}$) and pull the constant a out of the second integral:

$$xe^{ax} = \frac{1}{a} e^{ax} + a \int xe^{ax}\, dx$$

Rearrange to solve for our target integral $\int xe^{ax}\, dx$:

$$\int xe^{ax}\, dx = \frac{xe^{ax}}{a} - \frac{e^{ax}}{a^2} + C$$

Solution 3: The Leibniz Rule

This is a direct application of the 1D Leibniz Rule. The upper limit is a function of x ($g(x) = x^3$), and the lower limit is a constant.

$$\frac{d\theta}{dx} = f(g(x))\frac{dg}{dx}$$

Plug x^3 in for the dummy variable t, and multiply by the derivative of x^3:

$$\frac{d\theta}{dx} = e^{-2(x^3)} \cdot (3x^2) = 3x^2 e^{-2x^3}$$

Solution 4: Spot the Mistake

The student failed to use a **dummy variable**. The limits of integration dictate the physical bounds of the problem (from time zero to the current time t). You cannot have the upper limit of an integral be the same variable you are integrating over inside the integral! If t is the upper limit, the inside variable must be something else (like t' or τ). The correct notation is:

$$x(t) = \int_0^t c_1 t' \, dt' = \frac{1}{2} c_1 t'^2 \Big|_0^t = \frac{1}{2} c_1 t^2$$

Note: If you don't use dummy variables in moving-boundary transport problems, you will inevitably plug variables into the wrong places when evaluating your limits!

Solution 5: The 2D Control Volume

Since the limits are constants, we can evaluate the integrals sequentially. Let's start with the inner integral (y):

$$\begin{aligned}
M &= \int_0^L C_0 \left[y - \frac{y^3}{3h^2} \right]_{-h}^{h} dx \\
&= \int_0^L C_0 \left(\left(h - \frac{h^3}{3h^2} \right) - \left(-h - \frac{(-h)^3}{3h^2} \right) \right) dx \\
&= \int_0^L C_0 \left(\left(h - \frac{h}{3} \right) - \left(-h + \frac{h}{3} \right) \right) dx \\
&= \int_0^L C_0 \left(\frac{2h}{3} - \left(-\frac{2h}{3} \right) \right) dx \\
&= \int_0^L C_0 \left(\frac{4h}{3} \right) dx
\end{aligned}$$

Now evaluate the outer integral (x):

$$M = C_0 \left(\frac{4h}{3} \right) [x]_0^L = \frac{4}{3} C_0 h L$$

Note: If you divide by the cross-sectional area $A = 2hL$ you find that the average concentration is just $M/A = \frac{2}{3}C_0$, the common result for a quadratic profile in a rectangular channel.

3 The Taylor Series

A very useful mathematical technique is the Taylor Series:

$$\begin{aligned} f(x) &= f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + \dots \\ &= \sum_{n=0}^{\infty} \frac{f^{(n)}(x_0)}{n!}(x - x_0)^n \end{aligned}$$

Effectively it is a polynomial expansion of a function about x_0. If $x_0 = 0$ it is called the Maclaurin Series.

A Taylor series requires the derivatives of the function to exist at the expansion point. For the infinite series to converge to the function itself, additional conditions are required.

We often truncate after the first (extra) term to linearize a problem. Suppose we are looking at the motion of a pendulum. The restoring force is proportional to sine (where θ is the displacement). Because sine is non-linear in θ, this is inconvenient. We can linearize using the Taylor Series; expand about $\theta = 0$:

$$\sin\theta = \sin(0) + \cos(0)(\theta) - \frac{\sin(0)}{2}\theta^2 - \frac{\cos(0)}{3!}\theta^3 + \dots$$

Note that this approximation is only accurate when $|\theta| \ll 1$ (in radians).

Since $\sin(0) = 0$ and $\cos(0) = 1$, this simplifies:

$$\begin{aligned} \sin\theta &= \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \dots \\ &\approx \theta + O(\theta^3) \end{aligned}$$

which makes analyzing a pendulum much easier! We can truncate after any term by not specifying where the last derivative is evaluated:

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(\xi)(x - x_0)^2$$

where $x_0 < \xi < x$.

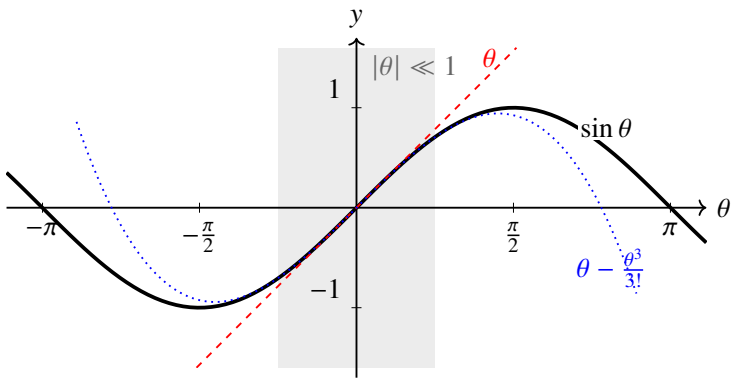


Figure 3.1: Taylor series approximations of $\sin(\theta)$ centered at $\theta = 0$. The exact function (solid black) is compared to its first-order (dashed red) and third-order (dotted blue) polynomial expansions. The shaded region illustrates where the small-angle approximation ($y \approx \theta$) is valid ($|\theta| \ll 1$), which is heavily utilized in linearizing pendulum dynamics.

Common Series Expansions

There are a few series expansions you should know:

$$\begin{aligned}
\sin x &= x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \\
\cos x &= 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots \\
e^x &= 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots \\
\ln(1+x) &= x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots \\
\frac{1}{1-x} &= \sum_{n=0}^{\infty} x^n
\end{aligned}$$

The last two are interesting due to a finite radius of convergence. If x is too large, the series diverges! In both cases it fails for $|x| > 1$.

One common way to test convergence is the ratio test, which examines the ratio of successive terms as $n \to \infty$. If the limit of the absolute value of the ratio of successive terms is less than 1, the series converges absolutely.

For some functions such as e^{-1/x^2} (with $f(0) = 0$) expanded about $x = 0$, the Taylor series exists but does not converge to the original function. In this case every derivative at $x = 0$ is zero, so the Taylor series is identically zero even though the function itself is not. For some functions the Taylor series doesn't exist at all! The Wikipedia article on the Taylor series has a nice summary of convergence behavior.

You can also do a Taylor series expansion with more than one variable as well:

$$f(\mathbf{x}) = f(\mathbf{x}_0) + \nabla f|_{\mathbf{x}_0} \cdot (\mathbf{x} - \mathbf{x}_0) + \frac{1}{2}(\mathbf{x} - \mathbf{x}_0)^T \cdot \left[\nabla^T \nabla f|_{\mathbf{x}_0}\right] \cdot (\mathbf{x} - \mathbf{x}_0) + \dots$$

where ∇ is the gradient operator and $\mathbf{x}$ is a column vector and the matrix $H = \nabla^T \nabla f$ is the Hessian.The multivariate Taylor series expansion is very useful in doing error propagation calculations and sensitivity analysis. We will discuss the gradient operator in more detail in a later section.

Algorithm Error

The Taylor series is useful for many things, a good example being algorithm error. Suppose you want to estimate the derivative of a function at x_0 (say, at the boundary of a domain) using the values at x_0 and $x_1 \equiv x_0 + \Delta x$.

$$f'(x_0) \approx \frac{f(x_1) - f(x_0)}{\Delta x}$$

To calculate the error in this approximation we expand $f(x)$ about x_0:

$$f(x_1) = f(x_0) + (x_1 - x_0) f'(x_0) + \frac{1}{2}(x_1 - x_0)^2 f''(\xi)$$

So:

$$\begin{aligned} \frac{f(x_1) - f(x_0)}{\Delta x} &= \frac{f(x_0) + \Delta x f'(x_0) + \frac{1}{2}(\Delta x)^2 f''(\xi) - f(x_0)}{\Delta x} \\ &= f'(x_0) + \frac{1}{2}\Delta x f''(\xi) \end{aligned}$$

The second term here is the algorithm error, thus our error is $O(\Delta x)$.

We can do better than this if we use the points at $x_{-1} = x_0 - \Delta x$ and $x_{+1} = x_0 + \Delta x$:

$$f'(x_0) \approx \frac{f(x_1) - f(x_{-1})}{2\Delta x}$$

where:

$$f(x_1) = f(x_0) + \Delta x f'(x_0) + \frac{1}{2}\Delta x^2 f''(x_0) + \frac{1}{6}\Delta x^3 f'''(\xi)$$

and:

$$f(x_{-1}) = f(x_0) - \Delta x f'(x_0) + \frac{1}{2}\Delta x^2 f''(x_0) - \frac{1}{6}\Delta x^3 f'''(\xi)$$

so:

$$\frac{f(x_1) - f(x_{-1})}{2\Delta x} = \frac{2\Delta x f'(x_0) + 0 + \frac{1}{3}\Delta x^3 f'''(\xi)}{2\Delta x}$$

$$= f'(x_0) + \frac{1}{6}\Delta x^2 f'''(\xi)$$

Note: the above expression is not exact, because the point ξ where the third derivative is evaluated in each sub interval must actually be different, however the error in ignoring this is at even higher order.

From the above expression we find that our error is now $O(\Delta x^2)$! This is the difference between a forward difference algorithm ($O(\Delta x)$) and a center difference algorithm ($O(\Delta x^2)$).

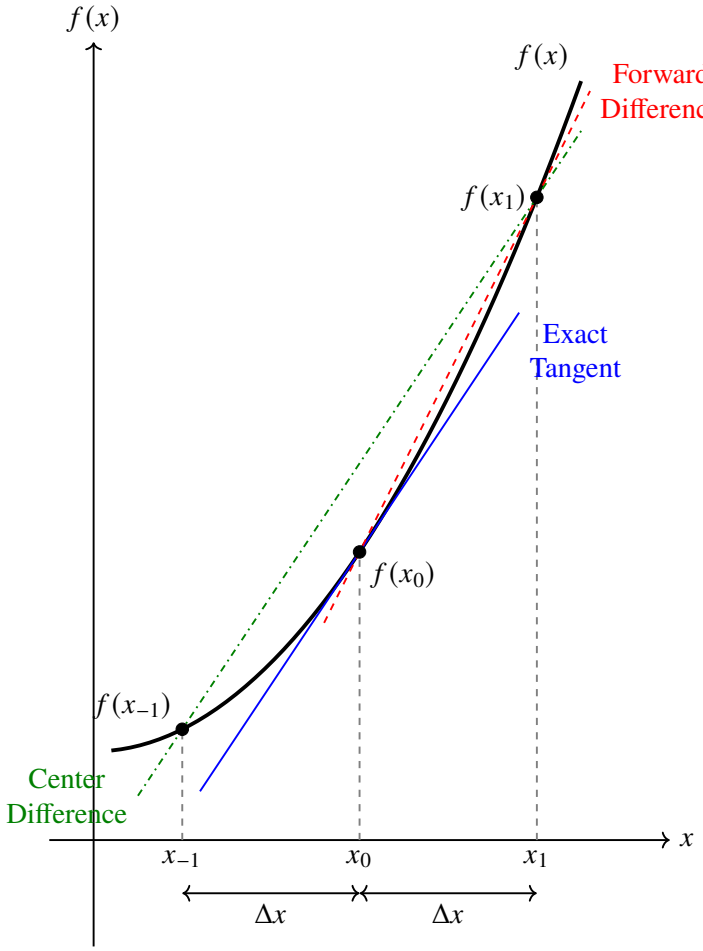


Figure 3.2: Visual comparison of finite difference approximations for the derivative $f'(x_0)$. The exact derivative is represented by the solid blue tangent line at x_0. The forward difference (dashed red) uses x_0 and x_1, yielding a secant line whose slope visibly diverges from the true tangent, yielding $O(\Delta x)$ error. The center difference (dash-dotted green) steps both backward to x_{-1} and forward to x_1; its resulting secant line is perfectly parallel to the exact tangent, demonstrating why evaluating points symmetrically around x_0 dramatically reduces truncation error to $O(\Delta x^2)$.

A problem in using a center difference formula at the boundary is that there is no x_{-1}. We can use our Taylor series to still get $O(\Delta x^2)$ error if we have a third point x_2.

Our error was:

$$\frac{f(x_1) - f(x_0)}{\Delta x} = f'(x_0) + \frac{1}{2}\Delta x f''(\xi)$$

If we estimate $f''(\xi)$ we can subtract off the error. f'' is just the derivative of the derivative, thus (if $x_2 = x_0 + 2\Delta x$):

$$f''(x_1) \approx \frac{\frac{f(x_2)-f(x_1)}{\Delta x} - \frac{f(x_1)-f(x_0)}{\Delta x}}{\Delta x}$$

The error of this formula is $O(\Delta x^2)$ as it is the center difference formula for the second derivative at x_1.

So:

$$f''(x_1) \approx \frac{f(x_2) + f(x_0) - 2f(x_1)}{\Delta x^2}$$

$$\begin{aligned} f'(x_0) &\approx \frac{f(x_1) - f(x_0)}{\Delta x} - \frac{\Delta x}{2}\left[\frac{f(x_2) + f(x_0) - 2f(x_1)}{\Delta x^2}\right] \\ &= \frac{1}{2\Delta x}\left(4f(x_1) - 3f(x_0) - f(x_2)\right) \end{aligned}$$

What is the error? Using the Taylor series again we get:

$$\begin{aligned} f(x_0) &= f(x_0) \\ f(x_1) &= f(x_0) + \Delta x f'(x_0) + \frac{1}{2}\Delta x^2 f''(x_0) + \frac{1}{6}\Delta x^3 f'''(\xi) \\ f(x_2) &= f(x_0) + 2\Delta x f'(x_0) + 2\Delta x^2 f''(x_0) + \frac{4}{3}\Delta x^3 f'''(\xi) \end{aligned}$$

So (after lots of cancellation):

$$\begin{aligned} f'(x_0) &\approx \frac{1}{\Delta x}\left(2f(x_1) - \frac{3}{2}f(x_0) - \frac{1}{2}f(x_2)\right) \\ &= f'(x_0) - \frac{1}{3}\Delta x^2 f'''(\xi) \end{aligned}$$

This algorithm (or truncation) error is approximately twice that using $f(x_1)$ and $f(x_{-1})$ but it works at boundaries where $f(x_{-1})$ is undefined. You use these sorts of algorithms when doing finite difference solutions to PDE's. It is also useful in estimating derivatives from data in the lab where you are often interested in, say, a heat flux at a boundary but only have temperature measurements at discrete points in the interior.

Practice Problems

Test your skills on these problems. In engineering, the Taylor Series is your primary tool for turning difficult non-linear equations into solvable linear ones, and for extracting data from discrete measurements!

Problem 1: Linearizing Radiation

Radiation heat transfer relies on the Stefan-Boltzmann law, where the heat flux q is proportional to the temperature to the fourth power: $q = \epsilon\sigma T^4$.

Using a first-order Taylor series (truncating after the first derivative), linearize this equation for small temperature variations around a baseline temperature T_0.

Problem 2: The Maclaurin Series

The concentration of a diffusing species near a source can often be modeled as an exponential decay: $C(x) = C_0 e^{-ax}$.

Expand this function in a Maclaurin series (a Taylor series about $x = 0$) up to the second-order term (the x^2 term).

Problem 3: Estimating Gradients

Given the non-linear velocity profile $u(y) = y^3$, evaluate the exact shear rate $\dfrac{du}{dy}$ at $y = 1$.

Then, estimate it numerically using:

1. The forward difference formula with $\Delta y = 0.1$.
2. The center difference formula with $\Delta y = 0.1$.

Which numerical estimate is closer to the exact value, and why?

Problem 4: Spot the Mistake

A student tries to derive the $O(\Delta x)$ **backward difference** formula for $f'(x_0)$ using the point behind the boundary, $x_{-1} = x_0 - \Delta x$. The student writes the Taylor expansion as:

$$f(x_{-1}) = f(x_0) + \Delta x f'(x_0) + \frac{1}{2}\Delta x^2 f''(x_0)$$

Rearranging this, the student concludes that $f'(x_0) = \frac{f(x_{-1}) - f(x_0)}{\Delta x}$.

Identify the student's mistake and write the correct backward difference formula.

Problem 5: The 3-Point Boundary Problem

You are measuring heat flux at a solid wall ($x = 0$). You have a thermocouple placed exactly at the wall ($x_0 = 0$), and two more in the fluid at $x_1 = 2$ mm and $x_2 = 4$ mm.

The measured temperatures are $T_0 = 300$ K, $T_1 = 290$ K, and $T_2 = 285$ K.

Using the $O(\Delta x^2)$ 3-point forward difference formula derived in the text, estimate the temperature gradient $\dfrac{dT}{dx}$ at the wall in K/mm.

Solutions to Practice Problems

Solution 1: Linearizing Radiation

Let $f(T) = \epsilon\sigma T^4$. The first-order Taylor expansion about T_0 is:

$$f(T) \approx f(T_0) + f'(T_0)(T - T_0)$$

Evaluate the function and its derivative at T_0:

$$\begin{aligned} f(T_0) &= \epsilon\sigma T_0^4 \\ f'(T) &= 4\epsilon\sigma T^3 \quad \Rightarrow \quad f'(T_0) = 4\epsilon\sigma T_0^3 \end{aligned}$$

Substitute these back into the expansion:

$$q \approx \epsilon\sigma T_0^4 + 4\epsilon\sigma T_0^3 (T - T_0)$$

Note: This linearization is extremely common in heat transfer when radiation occurs alongside convection!

Solution 2: The Maclaurin Series

A Maclaurin series expands about $x_0 = 0$: $f(x) \approx f(0) + f'(0)x + \frac{1}{2}f''(0)x^2$. Find the derivatives and evaluate them at $x = 0$:

$$\begin{aligned} f(x) &= C_0 e^{-ax} & \Rightarrow f(0) &= C_0 \\ f'(x) &= -aC_0 e^{-ax} & \Rightarrow f'(0) &= -aC_0 \\ f''(x) &= a^2 C_0 e^{-ax} & \Rightarrow f''(0) &= a^2 C_0 \end{aligned}$$

Substitute these into the series:

$$C(x) \approx C_0 - aC_0 x + \frac{1}{2}a^2 C_0 x^2$$

Solution 3: Estimating Gradients

First, find the exact value:

$$\frac{du}{dy} = 3y^2 \quad \Rightarrow \quad 3(1)^2 = 3.000$$

Next, evaluate the forward difference using $y_0 = 1$ and $y_1 = 1.1$:

$$\left.\frac{du}{dy}\right|_{\text{forward}} \approx \frac{u(1.1) - u(1.0)}{0.1} = \frac{(1.1)^3 - (1)^3}{0.1} = \frac{1.331 - 1}{0.1} = 3.310$$

Finally, evaluate the center difference using $y_{-1} = 0.9$ and $y_1 = 1.1$:

$$\left.\frac{du}{dy}\right|_{\text{center}} \approx \frac{u(1.1) - u(0.9)}{2(0.1)} = \frac{1.331 - 0.729}{0.2} = \frac{0.602}{0.2} = 3.010$$

The center difference is much closer to the exact value (3.010 vs 3.310) because its truncation error shrinks proportionally to Δy^2 rather than just Δy.

Solution 4: Spot the Mistake

The student forgot to account for the negative sign inside the evaluation point. The distance $(x_{-1} - x_0)$ is equal to $-\Delta x$, not $+\Delta x$. Therefore, the first-order term should be negative:

$$f(x_{-1}) = f(x_0) - \Delta x f'(x_0) + \frac{1}{2}\Delta x^2 f''(x_0)$$

Truncating the $O(\Delta x^2)$ term and rearranging to solve for $f'(x_0)$ yields the correct backward difference formula:

$$f'(x_0) \approx \frac{f(x_0) - f(x_{-1})}{\Delta x}$$

Solution 5: The 3-Point Boundary Problem

The step size is $\Delta x = 2$ mm. The formula derived in the text is:

$$f'(x_0) \approx \frac{1}{\Delta x}\left(2f(x_1) - \frac{3}{2}f(x_0) - \frac{1}{2}f(x_2)\right)$$

Substitute the measured temperatures:

$$\begin{aligned}\frac{dT}{dx} &\approx \frac{1}{2}\left(2(290) - 1.5(300) - 0.5(285)\right) \\ &\approx \frac{1}{2}\left(580 - 450 - 142.5\right) \\ &\approx \frac{1}{2}\left(-12.5\right) = -6.25 \text{ K/mm}\end{aligned}$$

Note: If we had used the basic $O(\Delta x)$ forward difference $(T_1 - T_0)/\Delta x$, we would have estimated the gradient as -5.0 K/mm. By accounting for the curvature using the third point, our estimate is significantly more accurate! There is a cost, however - the random error in the calculated derivative is proportional to the two-norm of the weightings on each point. Thus, the three point formula has a random error which is $\frac{\sqrt{13}}{2} = 1.8$ times greater than the random error of the simpler formula. The optimum depends on the ratio of the smaller algorithm error to the larger random error, but the three point formula usually wins.

4 First Order Linear ODEs

An equation of the form:

$$\frac{dy}{dx} + P(x)y = f(x)$$

appears in many problems. These range from accounting ($P(x)$ could be, say, a discount rate due to inflation or the negative of an interest rate paid by the bank on your account balance and $f(x)$ would be the revenue stream) to Chemical Engineering problems such as mass balances in a CSTR.

It is very convenient that it has an exact general solution! This solution is obtained using an integrating factor:

$$\mu(x) = e^{\int P(x)\,dx}$$

which converts the left side into the derivative of a product. The general solution is:

$$y = e^{-\int P(x)\,dx}\left[\int e^{+\int P(x)\,dx} f(x)\,dx + K\right]$$

where K is a constant determined from the initial condition (IC) or the value of $y(x)$ somewhere in the domain.

Example 1: Future Value with Compound Interest

Let's look at a couple of examples. Suppose you decide to save \$1/day for 10 years. What will it be worth at the end of that time?

One answer is simply the cash value:

$$\$1/\text{day} \times 3650 \text{ days} = \$3,650$$

But this ignores compound interest! Suppose you deposit this money into an investment account yielding a continuous interest rate of 5%/yr. The rate of change of your balance m is equal to the interest earned on the current balance plus the new deposits:

$$\frac{dm}{dt} = rm + S \quad \text{with} \quad m(0) = 0$$

where $r = 0.05$/yr and $S = 365$ \$/yr. Putting this in our standard form to solve: $x = t$, $y \equiv m$, $P(x) = -r$, $f(x) \equiv S$.

$$\frac{dm}{dt} - rm = S$$

So:

$$\begin{aligned} m &= e^{-\int -r\,dt}\left[\int Se^{\int -r\,dt}\,dt + K\right] \\ &= e^{rt}\left[\int Se^{-rt}\,dt + K\right] \\ &= e^{rt}\left[-\frac{S}{r}e^{-rt} + K\right] \\ &= -\frac{S}{r} + Ke^{rt} \end{aligned}$$

Now $m(0) = 0 \implies K = \frac{S}{r}$ and:

$$\begin{aligned} m &= \frac{S}{r}\left(e^{rt} - 1\right) \\ &= \frac{365}{0.05}\left(e^{0.05\times 10} - 1\right) \\ &= 7300\left(e^{0.5} - 1\right) \\ &= \$4,735 \end{aligned}$$

This is significantly more than the zero-interest value of \$3,650, showing the power of continuous compounding! For more complicated cases where interest rates or deposits fluctuate (P and f are not constant), the integrals tend to get a bit messy, so Wolfram Alpha or simple numerical quadrature schemes can be employed.

Example 2: Droplet Capture by a Face Shield

Sometimes higher order differential equations can be reduced to first order equations.

As an example we can answer a question that was critical during the pandemic: How large does a droplet need to be before a face shield can capture it?

In the actual situation, air flows around a face shield in a curve, but a droplet will want to move in a straight line due to inertia. If its displacement relative to the air is fairly large it will hit the shield and be captured via inertial impaction. If it is small, however, it follows the fluid right around the mask and avoids capture. This can actually be analyzed via the simple one-dimensional approximation. This problem is a lot like throwing something at a wall: if it is big it will hit it via inertial impaction, but if it's small enough viscosity slows it down and it never gets there!

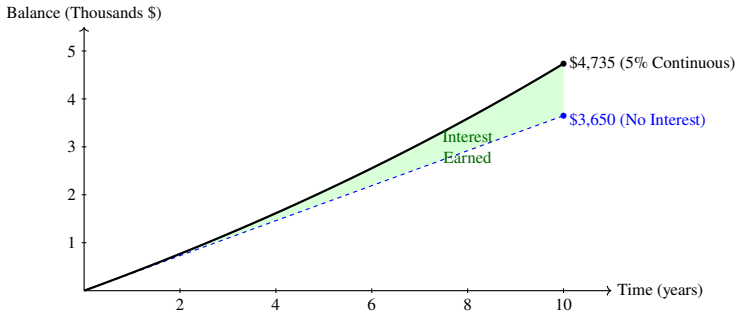


Figure 4.1: Comparison of saving $1/day over a 10-year period. The dashed blue line represents the linear cash value of the deposits ($3,650). The solid black curve demonstrates the exponential growth resulting from a continuous compound interest rate of 5% ($4,735). The shaded green area visually quantifies the interest earned, highlighting the divergence between simple accumulation and first-order linear ODE dynamics.

The trajectory is governed by $F = ma = m\dfrac{d^2x}{dt^2}$:

$$m\frac{d^2x}{dt^2} = -6\pi\mu a\frac{dx}{dt}$$

where the right side is viscous drag from Stokes Law, and a is the radius. This expression is valid for small particles at low Reynolds number, where Stokes drag applies.

The initial conditions are:

$$\left.\frac{dx}{dt}\right|_{t=0} = U_0, \quad x|_{t=0} = 0$$

This is second order in x, but we can solve for the velocity as a first order equation! Let $U = \dfrac{dx}{dt}$:

$$\frac{dU}{dt} = -\frac{6\pi\mu a}{m}U \quad \text{with} \quad U|_{t=0} = U_0$$

This yields the simple result $P(t) = \dfrac{6\pi\mu a}{m}$ and $f(t) = 0$.

$$U = U_0 e^{-\frac{6\pi\mu a}{m}t}$$

Finishing by integrating the velocity for position:

$$\frac{dx}{dt} = U_0 e^{-\frac{6\pi\mu a}{m}t}$$

$$x = \frac{U_0 m}{6\pi\mu a}\left(1 - e^{-\frac{6\pi\mu a}{m}t}\right)$$

At long times, the maximum distance traveled is $\Delta x = \dfrac{U_0 m}{6\pi\mu a}$. For a face shield to work, we need $\Delta x \sim 3$ cm.

Let $U_0 \sim 1$ m/s (typical droplet velocity), and mass $m = \frac{4}{3}\pi a^3 \rho$ (using the density of water).

$$\Delta x = U_0 \frac{\frac{4}{3}\pi a^3 \rho}{6\pi\mu a} = \frac{2}{9}\frac{U_0 a^2 \rho}{\mu}$$

Rearranging for the radius a:

$$\begin{aligned} a &= \left(\frac{9}{2}\frac{\mu\Delta x}{U_0\rho}\right)^{1/2} \\ &= \left(\frac{9}{2}\frac{(1.81\times 10^{-4}\ \text{g/cm}\cdot\text{s})(3\ \text{cm})}{(100\ \text{cm/s})(1\ \text{g/cm}^3)}\right)^{1/2} \\ &= 0.0049\ \text{cm} = 49\,\mu\text{m} \end{aligned}$$

The diameter is $2a \approx 100\,\mu$m. Typical oral drops are $\sim 25\,\mu$m, however, so face shields were not effective at capturing oral droplets. This is why during the pandemic we had to go with the more effective two-layer cloth masks or the equivalent to prevent the spread of Covid in the classroom prior to vaccine availability.

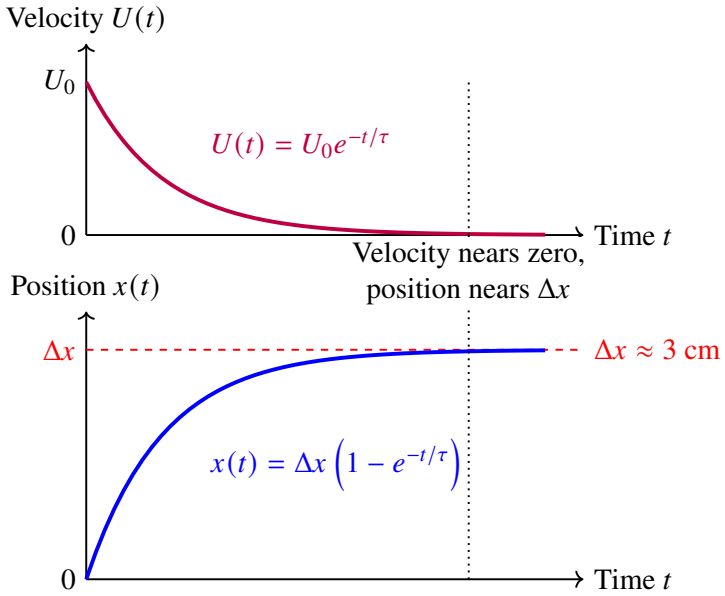


Figure 4.2: Kinematics of a microscopic droplet subjected to Stokes drag. The top graph shows the droplet's velocity decaying exponentially from its initial speed U_0. The bottom graph shows the corresponding position approaching the horizontal asymptote (Δx). If a face shield is placed beyond this asymptotic distance, the droplet will come to a halt before reaching it, avoiding inertial impaction and capture.

Practice Problems

Test your skills on these problems. Identifying the parts of the standard form $y' + P(x)y = f(x)$ and carefully tracking your constants of integration will get you through almost any 1st-order linear system in engineering!

Problem 1: The Baseline Formula

Solve the following first-order linear ODE subject to the initial condition $y(0) = 2$:

$$\frac{dy}{dx} + 3y = e^{-x}$$

Problem 2: The CSTR Washout

The text mentioned Continuous Stirred Tank Reactors (CSTRs). Suppose a tank of constant volume V is full of a dye at initial concentration C_0. At time $t = 0$, a stream of pure water begins flowing into the tank at a volumetric flow rate Q, and the perfectly mixed solution flows out at the same rate Q.

The mass balance for the dye is:

$$V\frac{dC}{dt} = -QC$$

Put this in standard first-order form and use the integrating factor method to solve for the concentration $C(t)$ as a function of time.

Problem 3: Non-Constant Coefficients

Solve the following ODE where $P(x)$ and $f(x)$ are not constants. (Leave your answer in terms of an unknown constant K).

$$\frac{dy}{dx} + \frac{1}{x}y = 3x$$

Hint: Remember that $e^{\ln(x)} = x$.

Problem 4: Spot the Mistake

A student is trying to solve the ODE $\frac{dy}{dx} + 2xy = 4x$. They find the integrating factor $e^{\int 2x\,dx} = e^{x^2}$ and write:

$$y = e^{-x^2} \int 4xe^{x^2}\,dx$$

They evaluate the integral to get $2e^{x^2}$, and then multiply by the e^{-x^2} in front, concluding that the only solution is $y = 2$.

While $y = 2$ is the steady-state solution, the student missed the transient portion of the answer. Identify their mistake and write the correct general solution.

Problem 5: Terminal Velocity

Similar to the face shield droplet, consider a spherical particle of mass m falling through a fluid. It is pulled down by gravity (mg) and slowed by a linear drag force ($-cv$). The equation of motion is:

$$m\frac{dv}{dt} = mg - cv$$

Assuming the particle is dropped from rest ($v(0) = 0$), solve for the velocity $v(t)$. Then, find the terminal velocity by evaluating $\lim_{t\to\infty} v(t)$.

Solutions to Practice Problems

Solution 1: The Baseline Formula

The equation is already in standard form with $P(x) = 3$ and $f(x) = e^{-x}$. Set up the general solution:

$$\begin{aligned} y &= e^{-\int 3\,dx}\left[\int e^{\int 3\,dx} e^{-x}\,dx + K\right] \\ &= e^{-3x}\left[\int e^{3x}e^{-x}\,dx + K\right] \\ &= e^{-3x}\left[\int e^{2x}\,dx + K\right] \\ &= e^{-3x}\left[\frac{1}{2}e^{2x} + K\right] \\ &= \frac{1}{2}e^{-x} + Ke^{-3x} \end{aligned}$$

Apply the initial condition $y(0) = 2$:

$$2 = \frac{1}{2}e^{0} + Ke^{0} \quad \Rightarrow \quad K = \frac{3}{2}$$

The final solution is:

$$y = \frac{1}{2}e^{-x} + \frac{3}{2}e^{-3x}$$

Solution 2: The CSTR Washout

First, rearrange into standard form by moving the concentration term to the left and dividing by V:

$$\frac{dC}{dt} + \frac{Q}{V}C = 0$$

Here, $P(t) = \frac{Q}{V}$ (a constant) and $f(t) = 0$. Apply the general formula:

$$\begin{aligned} C &= e^{-\int \frac{Q}{V}\,dt}\left[\int e^{\int \frac{Q}{V}\,dt}(0)\,dt + K\right] \\ &= e^{-\frac{Q}{V}t}\,[0 + K] \\ &= Ke^{-\frac{Q}{V}t} \end{aligned}$$

Apply the initial condition $C(0) = C_0$ to find $K = C_0$.

$$C(t) = C_0 e^{-\frac{Q}{V}t}$$

Note: The term V/Q is the "residence time" or time constant τ of the reactor!

Solution 3: Non-Constant Coefficients

Identify $P(x) = \frac{1}{x}$ and $f(x) = 3x$. The integrating factor is $e^{\int \frac{1}{x}\,dx} = e^{\ln x} = x$. Plug this into the general solution:

$$\begin{aligned} y &= \frac{1}{x}\left[\int x(3x)\,dx + K\right] \\ &= \frac{1}{x}\left[\int 3x^2\,dx + K\right] \\ &= \frac{1}{x}\left[x^3 + K\right] \\ &= x^2 + \frac{K}{x} \end{aligned}$$

Solution 4: Spot the Mistake

The student forgot to add the constant of integration K **inside** the brackets before multiplying by the outer e^{-x^2} term.

The proper setup using the exact solution formula is:

$$y = e^{-x^2}\left[\int 4xe^{x^2}\,dx + K\right]$$

Evaluating the integral gives $2e^{x^2}$, making the correct equation:

$$y = e^{-x^2}\left[2e^{x^2} + K\right] = 2 + Ke^{-x^2}$$

Without the $+K$ inside the bracket, the student entirely misses the transient decay term Ke^{-x^2}!

Solution 5: Terminal Velocity

Rearrange the equation into standard form for velocity v:

$$\frac{dv}{dt} + \frac{c}{m}v = g$$

Here, $P(t) = \frac{c}{m}$ and $f(t) = g$.

$$\begin{aligned} v &= e^{-\int \frac{c}{m}\,dt}\left[\int ge^{\int \frac{c}{m}\,dt}\,dt + K\right] \\ &= e^{-\frac{c}{m}t}\left[g\int e^{\frac{c}{m}t}\,dt + K\right] \\ &= e^{-\frac{c}{m}t}\left[g\left(\frac{m}{c}e^{\frac{c}{m}t}\right) + K\right] \\ &= \frac{mg}{c} + Ke^{-\frac{c}{m}t} \end{aligned}$$

Apply the initial condition $v(0) = 0$:

$$0 = \frac{mg}{c} + Ke^0 \quad \Rightarrow \quad K = -\frac{mg}{c}$$

Substitute K back in to get the final velocity profile:

$$v(t) = \frac{mg}{c}\left(1 - e^{-\frac{c}{m}t}\right)$$

The characteristic decay time is $\tau = m/c$.

To find the terminal velocity, take the limit as $t \to \infty$. The exponential term $e^{-\infty}$ goes to zero, leaving:

$$v_{\text{terminal}} = \frac{mg}{c}$$

Note: You can also find the terminal velocity simply by setting $\dfrac{dv}{dt} = 0$ *in the original ODE, since terminal velocity implies acceleration has ceased!*

5 Higher Order Linear ODEs

A linear ODE is linear in the dependent variable. The general form is:

$$\sum_{i=0}^{n} p_i(x) \frac{d^i y}{dx^i} = f(x)$$

where $p_i(x)$ is an arbitrary function of x, and $f(x)$ is the inhomogeneity. The equation is linear because the dependent variable y and all of its derivatives appear only to the first power and are never multiplied together.

For example:

$$x^2 \frac{d^2 y}{dx^2} + x \frac{dy}{dx} + (x^2 - n^2) y = 0$$

is a linear second order homogeneous ODE. It is actually a very special equation: Bessel's Equation of order n, which has well-known solutions, the famous Bessel functions $J_n(x)$ and $Y_n(x)$. These functions occur quite often when solving transport problems in a cylindrical domain.

Let's look at a simpler equation:

$$\frac{d^2 y}{dx^2} + \omega^2 y = x$$

This is a linear second order inhomogeneous ODE. Because it is second order, we need two conditions to determine the two unknowns which will appear in the solution. This could be two initial conditions (ICs):

$$y(0) = 1, \quad y'(0) = 0$$

or it could be boundary conditions (BCs), such as:

$$y(0) = 1, \quad y(1) = 0$$

For an inhomogeneous equation, it is usually convenient to break it up into a particular solution (which satisfies the inhomogeneous equation) and a homogeneous solution (which satisfies the equation with the inhomogeneity removed). The total solution (for linear ODEs only!!) is the sum of the two, $y = y^h + y^p$, which must together satisfy the overall BCs or ICs.

So for our problem:

$$\frac{d^2 y}{dx^2} + \omega^2 y = x \quad \text{with ICs: } y(0) = 1,\ y'(0) = 0$$

The particular solution by inspection is just $y^p = \frac{x}{\omega^2}$. Inspection often works for the particular solution, but it is always important to plug it back into the ODE to verify that you've got it right!

Now for the homogeneous solution y^h:

$$\frac{d^2y^h}{dx^2} + \omega^2 y^h = 0$$

The solution is just sines and cosines:

$$y^h = A\sin(\omega x) + B\cos(\omega x)$$

The total solution is:

$$y = y^h + y^p = \frac{x}{\omega^2} + A\sin(\omega x) + B\cos(\omega x)$$

Applying the initial conditions:

$$y(0) = 1 \implies 0 + 0 + B = 1 \implies B = 1$$
$$y'(0) = 0 \implies \frac{1}{\omega^2} + A\omega + 0 = 0 \implies A = -\frac{1}{\omega^3}$$

So the final solution is:

$$y = \frac{x}{\omega^2} - \frac{\sin(\omega x)}{\omega^3} + \cos(\omega x)$$

Periodic Inhomogeneity and Resonance

This problem is more interesting if our inhomogeneity is periodic:

$$\frac{d^2y}{dx^2} + \omega^2 y = \sin(ax)$$

with ICs: $y(0) = y'(0) = 0$.

Again by inspection, we expect a particular solution of the form $y^p = c\sin(ax)$. Plug it in!

$$-ca^2\sin(ax) + c\omega^2\sin(ax) = \sin(ax)$$

$$c = \frac{1}{\omega^2 - a^2}$$

and therefore:

$$y^p = \frac{1}{\omega^2 - a^2}\sin(ax)$$

The homogeneous solution is the same as before, so:

$$y = \frac{1}{\omega^2 - a^2}\sin(ax) + A\sin(\omega x) + B\cos(\omega x)$$

Now apply the initial conditions:

$$y(0) = 0 \implies B = 0 \quad \text{(this time!)}$$

$$\begin{aligned} y'(0) = 0 &\implies \frac{a}{\omega^2 - a^2} + A\omega = 0 \\ &\implies A = -\frac{a/\omega}{\omega^2 - a^2} \end{aligned}$$

So the complete solution is:

$$y = \frac{1}{\omega^2 - a^2}\left(\sin(ax) - \frac{a}{\omega}\sin(\omega x)\right)$$

This is the classic driven undamped oscillator and the solution is particularly interesting when $a \to \omega$. If the driving frequency approaches the natural frequency, the amplitude blows up!

To see exactly why the amplitude blows up as $a \to \omega$, we can evaluate the limit of our solution. Direct substitution of $a = \omega$ into the equation yields the indeterminate form $0/0$, so we must apply L'Hôpital's rule. L'Hôpital's rule arises naturally from a Taylor series expansion of the numerator and denominator. Since the limit is taken as $a \to \omega$, we differentiate numerator and denominator with respect to the parameter a, treating x as a constant:

$$\begin{aligned} \lim_{a\to\omega} y &= \lim_{a\to\omega} \frac{\sin(ax) - \frac{a}{\omega}\sin(\omega x)}{\omega^2 - a^2} \\ &= \lim_{a\to\omega} \frac{\frac{d}{da}\left[\sin(ax) - \frac{a}{\omega}\sin(\omega x)\right]}{\frac{d}{da}\left[\omega^2 - a^2\right]} \\ &= \lim_{a\to\omega} \frac{x\cos(ax) - \frac{1}{\omega}\sin(\omega x)}{-2a} \end{aligned}$$

Now, substituting $a = \omega$ gives a well-defined expression:

$$\begin{aligned} \lim_{a\to\omega} y &= \frac{x\cos(\omega x) - \frac{1}{\omega}\sin(\omega x)}{-2\omega} \\ &= \frac{\sin(\omega x)}{2\omega^2} - \frac{x\cos(\omega x)}{2\omega} \end{aligned}$$

Notice the second term in this final equation: $-\dfrac{x}{2\omega}\cos(\omega x)$. Because the independent variable x (which often represents time in dynamical systems) is multiplying the cosine wave, the amplitude of the oscillations grows linearly and without bound! This mathematical result perfectly describes the catastrophic physical phenomenon of resonance. Resonance is a very big problem in engineering. Bridges, turbines, rotating machinery, aircraft wings, and even buildings can fail catastrophically if driven near a natural frequency.

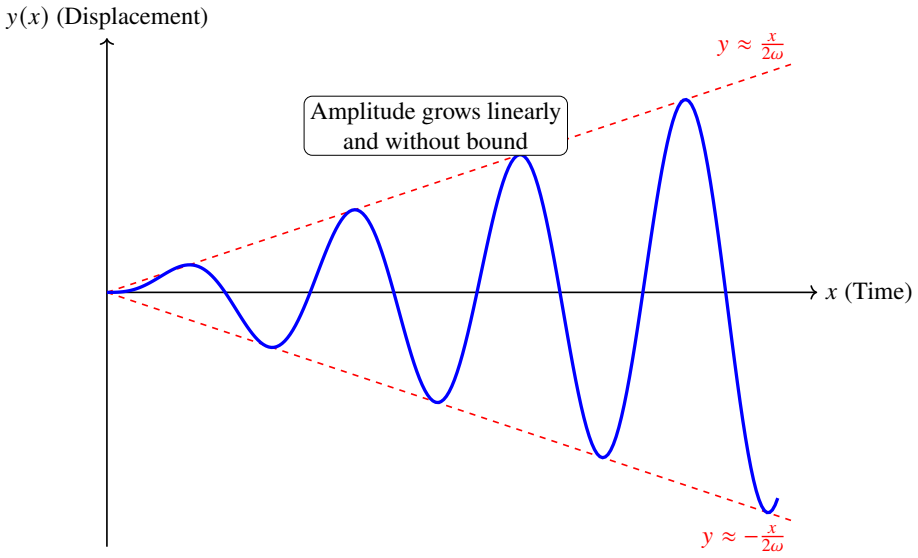


Figure 5.1: The mathematical manifestation of resonance. When the periodic driving frequency a perfectly matches the natural frequency ω of the undamped system, the resulting displacement $y(x)$ (solid blue curve, shown here for $\omega = 1$) oscillates with an ever-increasing amplitude. The wave expands to fill the linear envelopes formed by $\pm\frac{x}{2\omega}$ (dashed red lines). This unbounded displacement corresponds to the catastrophic mechanical failures seen in real-world bridges and machinery.

Practice Problems

Test your skills on these problems. Remember the core strategy for linear systems: find the homogeneous solution, find a particular solution, add them together, and *only then* apply your initial or boundary conditions!

Problem 1: The Baseline Formula

Solve the following second-order inhomogeneous ODE subject to the initial conditions $y(0) = 0$ and $y'(0) = 0$:

$$\frac{d^2y}{dx^2} + 9y = 18x$$

Hint: By inspection, guess a particular solution of the form $y^p = cx$ and plug it in to find c.

Problem 2: Boundary Conditions

So far we have mostly used Initial Conditions (where both conditions are specified at $x = 0$). Let's try a Boundary Value Problem (BVP). Solve the homogeneous equation:

$$\frac{d^2y}{dx^2} + \pi^2 y = 0$$

subject to the boundary conditions $y(0) = 0$ and $y(1/2) = 2$.

Problem 3: Inspection with Exponentials

Given the ODE $\frac{d^2y}{dx^2} + y = e^x$, find the particular solution y^p.

Hint: Guess a particular solution of the form $y^p = ce^x$ and plug it into the ODE to solve for the constant c.

Problem 4: Spot the Mistake

A student is presented with the following differential equation governing a non-linear spring:

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

The student attempts to solve it by finding a homogeneous solution y^h (by setting the right side to zero) and a particular solution y^p, and then adding them together $y = y^h + y^p$.

Why will the student's final answer be completely incorrect?

Problem 5: Avoiding Resonance

A machine part mounted on a chassis can be modeled as an undamped harmonic oscillator. The governing equation for its displacement $y(t)$ when subjected to a periodic vibration from the engine is:

$$\frac{d^2y}{dt^2} + 25y = \cos(\omega t)$$

where ω is the driving frequency of the engine in rad/s.

(a) What is the natural frequency of the machine part?

(b) At what engine driving frequency ω will the part experience catastrophic resonance?

Solutions to Practice Problems

Solution 1: The Baseline Formula

First, find the particular solution by guessing $y^p = cx$. Plug this into the ODE:

$$\frac{d^2}{dx^2}(cx) + 9(cx) = 18x$$
$$0 + 9cx = 18x \quad \Longrightarrow \quad c = 2$$

So, $y^p = 2x$.

Next, find the homogeneous solution for $\dfrac{d^2y^h}{dx^2} + 9y^h = 0$. Since $\omega^2 = 9$, $\omega = 3$.

$$y^h = A\sin(3x) + B\cos(3x)$$

The total solution is $y = y^h + y^p = 2x + A\sin(3x) + B\cos(3x)$. Now, apply the initial conditions to the total solution:

$$y(0) = 0 \implies 2(0) + A(0) + B(1) = 0 \implies B = 0$$
$$y'(x) = 2 + 3A\cos(3x) - 3B\sin(3x)$$
$$y'(0) = 0 \implies 2 + 3A(1) - 0 = 0 \implies A = -\frac{2}{3}$$

The final solution is:

$$y(x) = 2x - \frac{2}{3}\sin(3x)$$

Solution 2: Boundary Conditions

The equation is homogeneous, so we only need y^h. Here, $\omega^2 = \pi^2$, so $\omega = \pi$.

$$y(x) = A\sin(\pi x) + B\cos(\pi x)$$

Apply the first boundary condition at $x = 0$:

$$y(0) = 0 \implies A(0) + B(1) = 0 \implies B = 0$$

Now apply the second boundary condition at $x = 1/2$:

$$y(1/2) = 2 \implies A\sin\left(\pi \cdot \frac{1}{2}\right) = 2$$
$$\implies A(1) = 2 \implies A = 2$$

The final solution is:

$$y(x) = 2\sin(\pi x)$$

Note: Boundary Value Problems (BVPs) are very common in transport phenomena and heat transfer, where temperatures or concentrations are fixed at two physical walls!

Solution 3: Inspection with Exponentials

Following the hint, let $y^p = ce^x$. The first derivative is ce^x, and the second derivative is ce^x. Plug this back into the original ODE:

$$\begin{aligned} (ce^x) + (ce^x) &= e^x \\ 2ce^x = e^x \quad &\implies \quad 2c = 1 \quad \implies \quad c = \frac{1}{2} \end{aligned}$$

The particular solution is $y^p = \frac{1}{2}e^x$.

Solution 4: Spot the Mistake

The student failed to verify that the ODE was **linear**. The presence of the y^2 term makes this a non-linear differential equation.

The Principle of Superposition (which states that $y = y^h + y^p$ is a valid solution framework) applies **exclusively to linear systems**. If you try to plug $y^h + y^p$ into y^2, you get $(y^h + y^p)^2 = (y^h)^2 + 2(y^h)(y^p) + (y^p)^2$, which completely ruins the separation of the homogeneous and particular terms!

Solution 5: Avoiding Resonance

(a) The standard form for the undamped oscillator is $\dfrac{d^2y}{dt^2} + \omega_n^2 y = f(t)$, where ω_n is the natural frequency. Here, $\omega_n^2 = 25$, so the natural frequency is $\omega_n = 5$ rad/s.

(b) Resonance occurs when the driving frequency exactly matches the natural frequency of the system. Therefore, the part will experience resonance when the engine drives it at $\omega = 5$ rad/s.

6 Special Higher Order ODEs

Most high order ODEs are hard to solve, even if linear! There are two commonly found exceptions: Constant Coefficient ODEs and Euler equations. Let's look at these!

Constant Coefficient ODEs

The homogeneous form is:

$$\sum_{m=0}^{n} a_m \frac{d^m y}{dx^m} = 0$$

Note that if it were inhomogeneous, we would have to get the particular solution as well.

For homogeneous linear constant-coefficient ODEs, exponential functions are eigenfunctions of differentiation (e.g., after differentiation you get the same function back again, just multiplied by a constant), so we seek solutions of the form $y = e^{rx}$. Plugging in this assumed form:

$$\frac{d^m}{dx^m}(e^{rx}) = r^m e^{rx}$$

So:

$$\sum_{m=0}^{n} a_m r^m = 0$$

is an n-th order polynomial for all the r's. There are n roots counting multiplicity. If all roots are distinct, the solution is:

$$y = \sum_{m=1}^{n} c_m e^{r_m x}$$

Note that the r_m can be complex: this is what yields sines & cosines in the solution. To see this, let's look at our oscillator again:

$$\frac{d^2 y}{dx^2} + \omega^2 y = 0$$

Let $y = e^{rx}$:

$$\begin{aligned} r^2 e^{rx} + \omega^2 e^{rx} &= 0 \\ r^2 &= -\omega^2 \\ r &= \pm i\omega \end{aligned}$$

So $y = c_1 e^{i\omega x} + c_2 e^{-i\omega x}$. But using Euler's formula we have the equivalent:

$$y = A\sin(\omega x) + B\cos(\omega x)$$

as before!

Let's look at the damped oscillator:

$$\frac{d^2y}{dx^2} + 2\lambda\frac{dy}{dx} + y = 0$$

where 2λ is the damping coefficient. This equation has been nondimensionalized so that the natural frequency of the undamped system is unity. Inserting our assumed form $y = e^{rx}$ yields:

$$r^2 + 2\lambda r + 1 = 0$$

$$r = \frac{-2\lambda \pm \sqrt{4\lambda^2 - 4}}{2} = -\lambda \pm \sqrt{\lambda^2 - 1}$$

If $\lambda > 1$, we have two real roots which are both negative:

$$r = -\lambda + \sqrt{\lambda^2 - 1} \quad \text{and} \quad r = -\lambda - \sqrt{\lambda^2 - 1}$$

This is an overdamped oscillator.

If $\lambda < 1$, we have two complex roots:

$$r = -\lambda \pm i\sqrt{1 - \lambda^2}$$

The solution is written as:

$$y = Ae^{-\lambda x}\sin\left(\sqrt{1-\lambda^2}x\right) + Be^{-\lambda x}\cos\left(\sqrt{1-\lambda^2}x\right)$$

which is the underdamped oscillator.

If $\lambda = 1$, we have repeated roots! $r = -\lambda = -1$. This is the critically damped case. Repeated roots reduce the number of linearly independent exponential solutions, so in general multiplying by x generates the second independent solution. You should verify this by plugging it back into the ODE. The two solutions are $y = e^{-x}$ and $y = xe^{-x}$. So:

$$y = Ae^{-x} + Bxe^{-x}$$

Higher order problems are done the exact same way. These constant coefficient ODEs occur in many problems in transport and mechanics, ranging from beam deflection under load or vibration to the 2-D viscous flow of a fluid in a corner (the Taylor wiper problem or Moffatt eddies).

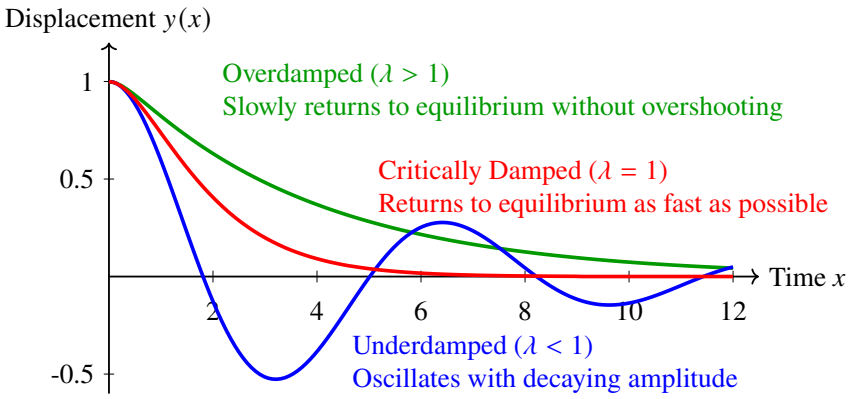


Figure 6.1: A comparison of the three damping regimes for a linear unforced harmonic oscillator ($y'' + 2\lambda y' + y = 0$), released from rest at an initial displacement of $y(0) = 1$. The critically damped case (red curve, $\lambda = 1$) represents the optimal return to zero. The underdamped case (blue curve) overshoots and oscillates, resulting from complex roots. The overdamped case (green curve) has distinct negative real roots, causing a sluggish return to equilibrium.

Euler Equations

The Euler equation is very similar:

$$\sum_{m=0}^{n} a_m x^m \frac{d^m y}{dx^m} = 0$$

In this case, the solutions are just $y = x^r$. Plugging in, you again get an n-th order polynomial for r. In fact, you can turn an Euler equation into a constant coefficient equation with the transformation $x \equiv e^t$, although it gets a little messy. (Strictly speaking, this transformation assumes $x > 0$.)

Let's look at an example. In the solution for Stokes flow past a sphere you get an equation of the form:

$$x^4 \frac{d^4 y}{dx^4} - 4x^2 \frac{d^2 y}{dx^2} + 8x \frac{dy}{dx} - 8y = 0$$

So we let $y = x^r$:

$$x^4 (r)(r-1)(r-2)(r-3)x^{r-4} - 4x^2 (r)(r-1)x^{r-2} + 8x(r)x^{r-1} - 8x^r = 0$$

$$r(r-1)(r-2)(r-3) - 4r(r-1) + 8r - 8 = 0$$

This has four roots: -1, 1, 2, 4.

$$\therefore\ y = \frac{c_1}{x} + c_2 x + c_3 x^2 + c_4 x^4$$

In this case we have the BCs:

$$\lim_{x \to \infty} \frac{y(x)}{x^2} = 0$$

which means that both $c_3 = c_4 = 0$.

The other BCs (at $x = 1$) are:

$$y(1) = -\frac{1}{2}, \quad \left.\frac{dy}{dx}\right|_{x=1} = -1$$

which yields $c_1 = \frac{1}{4}$ and $c_2 = -\frac{3}{4}$.

So:

$$y = \frac{1}{4x} - \frac{3}{4}x$$

Just as in the case of constant coefficient ODEs, you can get repeated roots! Suppose we have the repeated root $r = \lambda$. Then we have two solutions:

$$y = x^\lambda \quad \text{and} \quad y = x^\lambda \ln x$$

which makes sense as we can go from one equation to the other from the transformation $x = e^t$.

Practice Problems

Test your skills on these problems. For these specialized equations, your first step is always to identify the type (Constant Coefficient vs. Euler) so you know whether to guess $y = e^{rx}$ or $y = x^r$!

Problem 1: Third-Order Constant Coefficient

Find the general solution to the following 3rd-order homogeneous ODE:

$$\frac{d^3y}{dx^3} - \frac{d^2y}{dx^2} - 6\frac{dy}{dx} = 0$$

Problem 2: Vibrating Beam Mode Shapes

When analyzing the free transverse vibrations of an Euler-Bernoulli beam, separating the variables leads to a 4th-order spatial ODE for the mode shape $W(x)$:

$$\frac{d^4W}{dx^4} - \beta^4 W = 0$$

where β is a positive real constant containing the natural frequency and beam properties. Find the general solution for $W(x)$.

Problem 3: The Underdamped Oscillator

Solve the following initial value problem for a damped harmonic oscillator:

$$\frac{d^2y}{dt^2} + 4\frac{dy}{dt} + 13y = 0$$

with initial conditions $y(0) = 2$ and $y'(0) = 0$.

Problem 4: A Standard Euler Equation

Find the general solution to the following 2nd-order Euler equation:

$$x^2\frac{d^2y}{dx^2} - 2x\frac{dy}{dx} + 2y = 0$$

Problem 5: The Euler Transformation

The text mentioned that substituting $x = e^t$ transforms an Euler equation into a constant coefficient equation. Let's prove it! Consider the Euler equation:

$$x^2\frac{d^2y}{dx^2} + x\frac{dy}{dx} + 9y = 0$$

Using the chain rule, we know that $\dfrac{dy}{dx} = \dfrac{1}{x}\dfrac{dy}{dt}$ and $\dfrac{d^2y}{dx^2} = \dfrac{1}{x^2}\left(\dfrac{d^2y}{dt^2} - \dfrac{dy}{dt}\right)$.

Substitute these into the ODE to transform it into an equation in terms of y and t. Solve that new constant coefficient equation, and then substitute $t = \ln(x)$ back in to get your final answer $y(x)$.

Solutions to Practice Problems

Solution 1: Third-Order Constant Coefficient

This is a constant coefficient ODE, so we guess $y = e^{rx}$. This yields the characteristic polynomial:

$$r^3 - r^2 - 6r = 0$$

Factor out an r:

$$r(r^2 - r - 6) = 0 \quad \Longrightarrow \quad r(r-3)(r+2) = 0$$

The roots are $r = 0$, $r = 3$, and $r = -2$. Because the roots are real and distinct, the general solution is:

$$y(x) = c_1 e^{0x} + c_2 e^{3x} + c_3 e^{-2x} = c_1 + c_2 e^{3x} + c_3 e^{-2x}$$

Solution 2: Vibrating Beam Mode Shapes

This is a 4th-order constant coefficient ODE. Guessing $W = e^{rx}$ gives:

$$r^4 - \beta^4 = 0$$

This can be factored as a difference of squares:

$$(r^2 - \beta^2)(r^2 + \beta^2) = 0$$

The first term gives $r = \pm\beta$. The second term gives $r = \pm i\beta$. Combining the real exponentials and the complex (sine/cosine) terms, the general solution is:

$$W(x) = c_1 e^{\beta x} + c_2 e^{-\beta x} + A\sin(\beta x) + B\cos(\beta x)$$

*Note: In structural engineering texts, you will often see the exponential terms rewritten as hyperbolic sines and cosines (*sinh *and* cosh*), which makes applying boundary conditions at the ends of the beam much easier!*

Solution 3: The Underdamped Oscillator

Guess $y = e^{rt}$ to get the characteristic equation:

$$r^2 + 4r + 13 = 0$$

Using the quadratic formula:

$$r = \frac{-4 \pm \sqrt{16 - 52}}{2} = \frac{-4 \pm \sqrt{-36}}{2} = -2 \pm 3i$$

The roots are complex with $\lambda = 2$ and $\omega = 3$. The general solution is:

$$y(t) = e^{-2t}\left(A\cos(3t) + B\sin(3t)\right)$$

Apply the first initial condition $y(0) = 2$:

$$2 = e^0(A(1) + B(0)) \qquad \Longrightarrow \qquad A = 2$$

Now take the derivative $y'(t)$ using the product rule:

$$y'(t) = -2e^{-2t}(2\cos(3t) + B\sin(3t)) + e^{-2t}(-6\sin(3t) + 3B\cos(3t))$$

Apply the second initial condition $y'(0) = 0$:

$$0 = -2(1)(2 + 0) + (1)(0 + 3B)$$
$$0 = -4 + 3B \qquad \Longrightarrow \qquad B = \frac{4}{3}$$

The final solution is:

$$y(t) = e^{-2t}\left(2\cos(3t) + \frac{4}{3}\sin(3t)\right)$$

Solution 4: A Standard Euler Equation

This is an Euler equation, so we guess $y = x^r$. This gives:

$$\begin{aligned} x^2(r(r-1)x^{r-2}) - 2x(rx^{r-1}) + 2x^r &= 0 \\ r(r-1) - 2r + 2 &= 0 \\ r^2 - r - 2r + 2 &= 0 \\ r^2 - 3r + 2 &= 0 \end{aligned}$$

Factoring the polynomial yields $(r-1)(r-2) = 0$, so the roots are $r = 1$ and $r = 2$. The general solution is:

$$y(x) = c_1 x + c_2 x^2$$

Solution 5: The Euler Transformation

We substitute the given chain-rule expressions for $\frac{dy}{dx}$ and $\frac{d^2y}{dx^2}$ into the original Euler equation:

$$x^2\left[\frac{1}{x^2}\left(\frac{d^2y}{dt^2} - \frac{dy}{dt}\right)\right] + x\left[\frac{1}{x}\frac{dy}{dt}\right] + 9y = 0$$

Cancel the x terms:

$$\left(\frac{d^2y}{dt^2} - \frac{dy}{dt}\right) + \frac{dy}{dt} + 9y = 0$$

The $\dfrac{dy}{dt}$ terms completely cancel out, leaving a beautifully simple constant coefficient ODE:

$$\frac{d^2y}{dt^2} + 9y = 0$$

We know the solution to this harmonic oscillator is:

$$y(t) = A\cos(3t) + B\sin(3t)$$

Finally, since $x = e^t$, we know that $t = \ln(x)$. Substituting this back yields the final solution in terms of x:

$$y(x) = A\cos(3\ln(x)) + B\sin(3\ln(x))$$

Note: This is an example of why complex roots for Euler equations yield sines and cosines of the natural log of x, which is a pretty strange looking function! You can also recover this solution via the standard Euler solution approach (yielding x^{3i} and x^{-3i}) via the substitution $x^{3i} = e^{3i\ln(x)}$ and the Euler identity $e^{it} = \cos(t) + i\sin(t)$. Euler equations with complex roots are uncommon in transport; however, they do occur. Examples include heat transfer from a tapered fin with an exothermic surface reaction or the buckling of a tapered column.

7 Solving ODEs Via Transformation

Linear first order, constant coefficient, and Euler equations can be solved directly. While these are common, there are many other ODEs which arise in transport! Sometimes these can be solved via transformations of independent or dependent variables. Unlike the earlier examples, the approach is specific to each problem: there is no "one size fits all" method.

Let's look at some commonly seen examples!

1st Order Non-linear ODEs (Inverting the Derivative)

This is the easiest and most general example.

Suppose you squish a fluid between two plates with a constant force. The radius R of the fluid is governed by the non-linear first order equation (which depends on force, viscosity, etc.):

$$\frac{dR}{dt} = \frac{\lambda}{R^7} \quad \text{with} \quad R|_{t=0} = R_0 \text{ (initial radius)}$$

R is the dependent variable, but we can invert it to make the equation linear in t:

$$\frac{dt}{dR} = \frac{R^7}{\lambda}$$

Integrating this gives:

$$t = \frac{R^8}{8\lambda} + c$$

Applying the initial condition $t|_{R=R_0} = 0$:

$$c = -\frac{R_0^8}{8\lambda}$$

$$t = \frac{1}{8\lambda}(R^8 - R_0^8) \quad \text{or} \quad R = (R_0^8 + 8\lambda t)^{1/8}$$

Perfect Differentials (Separation of Variables)

There are other ways of looking at this. If we take the original problem and multiply by R^7, we can get a perfect differential:

$$\begin{aligned} R^7 \frac{dR}{dt} &= \lambda \\ \frac{1}{8}\frac{d(R^8)}{dt} &= \lambda \\ \therefore\ R^8 &= 8\lambda t + R_0^8 \quad \text{etc.} \end{aligned}$$

In fact, this works if the right hand side (RHS) is any separable function of R and t:

$$\begin{aligned} \text{Let } \frac{dR}{dt} &= f(R)g(t) \\ \frac{1}{f(R)}dR &= g(t)dt \\ \int \frac{1}{f(R)}dR &= \int g(t)dt \end{aligned}$$

which is also sometimes useful.

Perfect Differentials in Higher Order ODEs

Perfect differentials can be used in higher order ODEs too—sometimes!

A classic problem in radiative heat transport for space applications is the temperature distribution in a cooling fin. For a long fin in a vacuum without back radiation the temperature distribution is governed by:

$$\frac{d^2T}{dx^2} - \lambda T^4 = 0$$

where the first term is conduction along the fin and the second is cooling due to radiation. The boundary conditions are $T|_{x=0} = T_0$ and $T \to 0$ as $x \to \infty$.

Because it is a second order equation inverting doesn't help, but if we multiply by $\dfrac{dT}{dx}$ we can get a perfect differential!

$$\begin{aligned} \frac{d^2T}{dx^2}\frac{dT}{dx} &= \lambda T^4 \frac{dT}{dx} \\ \frac{1}{2}\frac{d}{dx}\left(\frac{dT}{dx}\right)^2 &= \frac{\lambda}{5}\frac{d(T^5)}{dx} \\ \frac{1}{2}\left(\frac{dT}{dx}\right)^2 &= \frac{\lambda}{5}T^5 + C \end{aligned}$$

Now as $x \to \infty$, both T and $\frac{dT}{dx}$ vanish, so $C = 0$. Note that this would not be true for a finite length fin, where the constant would need to be determined by the boundary condition (usually zero derivative) at the end - a somewhat more complicated case. For the long fin, however, we have the simple expression:

$$\left(\frac{dT}{dx}\right)^2 = \frac{2}{5}\lambda T^5$$

Taking the square root (the negative root is physical because it is cooling):

$$\frac{dT}{dx} = -\sqrt{\frac{2\lambda}{5}}T^{5/2}$$

Now we separate variables again:

$$\begin{aligned} T^{-5/2}\frac{dT}{dx} &= -\sqrt{\frac{2\lambda}{5}} \\ -\frac{2}{3}\frac{d(T^{-3/2})}{dx} &= -\sqrt{\frac{2\lambda}{5}} \\ T^{-3/2} &= \frac{3}{2}\sqrt{\frac{2\lambda}{5}}x + C \end{aligned}$$

Applying the boundary condition $T|_{x=0} = T_0$ yields $C = T_0^{-3/2}$. Thus the temperature profile is:

$$T = \left(T_0^{-3/2} + \frac{3}{2}\sqrt{\frac{2\lambda}{5}}x\right)^{-2/3}$$

This can be used to get the heat flux at the base which is the rate of cooling provided by the fin. This is proportional to:

$$\left.\frac{dT}{dx}\right|_{x=0} = -\sqrt{\frac{2\lambda}{5}}T_0^{5/2}$$

Autonomous Second Order ODEs

The previous example is actually a specific case of a much broader and very powerful mathematical rule. The cooling fin equation is an example of an **autonomous** second-order ODE. An autonomous differential equation is one where the independent variable (in this case, position x) does not explicitly appear anywhere in the equation.

The general form of an autonomous second-order ODE is:

$$\frac{d^2y}{dx^2} = f(y)$$

For equations of this exact form, multiplying by the first derivative $\frac{dy}{dx}$ will *always* turn the equation into a perfect differential! This is a mathematical guarantee:

$$\frac{d^2y}{dx^2}\frac{dy}{dx} = f(y)\frac{dy}{dx}$$
$$\frac{d}{dx}\left[\frac{1}{2}\left(\frac{dy}{dx}\right)^2\right] = \frac{d}{dx}\left[\int f(y)\,dy\right]$$

Let's test this on a much more complex version of the cooling fin. Suppose our long fin is not in a vacuum, but rather in air at an ambient temperature T_a. We also have a constant convective heat transfer coefficient h. The heat transfer now includes radiation out, back radiation in from the surroundings, and convection. The steady-state energy balance becomes:

$$\frac{d^2T}{dx^2} - \lambda(T^4 - T_a^4) - h(T - T_a) = 0$$

with boundary conditions $T|_{x=0} = T_0$ and $T \to T_a$ as $x \to \infty$.

Because this is an autonomous ODE (x does not appear explicitly), we are guaranteed that multiplying by $\frac{dT}{dx}$ will create perfect differentials:

$$\frac{d^2T}{dx^2}\frac{dT}{dx} - \lambda T^4\frac{dT}{dx} + \lambda T_a^4\frac{dT}{dx} - hT\frac{dT}{dx} + hT_a\frac{dT}{dx} = 0$$

Integrating every term once with respect to x yields:

$$\frac{1}{2}\left(\frac{dT}{dx}\right)^2 - \frac{\lambda}{5}T^5 + \lambda T_a^4 T - \frac{h}{2}T^2 + hT_aT = C$$

To find the integration constant C, we apply the boundary condition at the tip of the long fin. As $x \to \infty$, the temperature approaches the ambient air ($T \to T_a$), and the temperature gradient goes to zero ($\frac{dT}{dx} \to 0$). Plugging these in:

$$0 - \frac{\lambda}{5}T_a^5 + \lambda T_a^4(T_a) - \frac{h}{2}T_a^2 + hT_a(T_a) = C$$
$$\frac{4\lambda}{5}T_a^5 + \frac{h}{2}T_a^2 = C$$

Substituting C back in, we get an exact expression for the temperature gradient squared:

$$\frac{1}{2}\left(\frac{dT}{dx}\right)^2 = \frac{\lambda}{5}T^5 - \lambda T_a^4 T + \frac{h}{2}T^2 - hT_aT + \frac{4\lambda}{5}T_a^5 + \frac{h}{2}T_a^2$$

The Catch: To get the actual temperature profile $T(x)$, we would have to take the square root, separate the variables, and integrate a second time. However, a 5th-degree polynomial inside a

square root in the denominator forms a *hyperelliptic integral*, which has no analytical solution! It is, however, trivial to invert it and numerically integrate it to get $x(T)$ rather than $T(x)$. Plotting this up (switching axes) is equally trivial.

Even without solving for the full temperature profile $T(x)$, we have successfully solved for the **base heat flux** which is the most important quantity to know. By simply evaluating the above equation at $x = 0$ (where $T = T_0$), we can calculate the exact rate of heat removed by the fin analytically without ever needing to explicitly obtain $T(x)$!

Differentiation to Create a Perfect Differential

Sometimes you can simplify differential equations (and get perfect differentials) by differentiation - essentially "going backwards" to go forwards to the solution! As an example, the unsteady heating of a slab yields (after some work which we will discuss in a later section) the ODE:

$$\frac{d^2 f}{d\eta^2} + \frac{1}{2}\eta\frac{df}{d\eta} - \frac{1}{2}f = 0$$

with boundary conditions $\left.\dfrac{df}{d\eta}\right|_{\eta=0} = -1$ (a heat flux condition) and $f \to 0$ as $\eta \to \infty$.

We can solve this by rearranging and differentiation:

$$f'' = \frac{1}{2}(f - \eta f')$$

Differentiating with respect to η:

$$\begin{aligned} f''' &= \frac{1}{2}(f' - f' - \eta f'') \\ &= -\frac{1}{2}\eta f'' \end{aligned}$$

Now we can make this a perfect differential:

$$\begin{aligned} \frac{f'''}{f''} &= -\frac{1}{2}\eta \\ \frac{d}{d\eta}(\ln f'') &= -\frac{1}{2}\eta \end{aligned}$$

Integrating once:

$$\ln f'' = -\frac{1}{4}\eta^2 + C \implies f'' = c\, e^{-\frac{1}{4}\eta^2}$$

This can be integrated (twice) to get f' and f. Note that from the original differential equation at $\eta = 0$, we have $f''(0) = \frac{1}{2}f(0)$.

The final answer is proportional to the integral complementary error function, $\text{ierfc}(\eta/2)$.

Practice Problems

Test your skills on these problems. Because these are non-standard ODEs, pay close attention to the hints. Your goal is to find the mathematical "trick" that linearizes the equation or turns it into a perfect differential!

Problem 1: Inverting the Derivative

Find the relationship between x and y for the following non-linear first-order ODE:

$$\frac{dy}{dx} = \frac{1}{e^y + \cos(y)}$$

with the initial condition that $y = 0$ when $x = 2$.
Hint: Flip the derivative to solve for $x(y)$ instead of $y(x)$.

Problem 2: Separation of Variables

Find the general solution to the following ODE:

$$\frac{dy}{dx} = \frac{x}{ye^{y^2}}$$

Problem 3: Conservation of Energy

The unforced, undamped harmonic oscillator is governed by:

$$m\frac{d^2y}{dt^2} + ky = 0$$

where m is mass, k is the spring constant, y is position, and $\frac{dy}{dt}$ is velocity (v).

This is an autonomous ODE. Multiply the entire equation by the velocity $\frac{dy}{dt}$ and integrate once with respect to t to find a relationship between velocity and position.

Problem 4: A Solvable Autonomous ODE

We saw that the fin equation resulted in an impossible hyperelliptic integral. Let's try an autonomous ODE that actually has a clean analytical solution. Solve the following non-linear 2nd-order ODE:

$$\frac{d^2y}{dx^2} = 6y^2$$

subject to the initial conditions $y(0) = 1$ and $y'(0) = -2$.
Hint: Multiply by y' and integrate. When you take the square root for the second integration, be careful with your $\pm$ signs based on the initial conditions!

Problem 5: Going Backwards to Go Forwards

Sometimes differentiating an ODE simplifies it by completely eliminating a lower-order term. Consider the following non-homogeneous ODE:

$$x\frac{d^2y}{dx^2} - \frac{dy}{dx} = 3x^2$$

with initial conditions $y(1) = 0$ and $y'(1) = \frac{1}{2}$.

Differentiate the entire equation with respect to x to find an explicit equation for y''', and then integrate.
Beware: Because you differentiated the original equation, you will introduce an extraneous "fake" constant of integration! You must plug your y' and y'' equations back into the original ODE to eliminate it before applying your initial conditions.

Solutions to Practice Problems

Solution 1: Inverting the Derivative

The equation is highly non-linear in y, but we can invert the derivative:

$$\frac{dx}{dy} = e^y + \cos(y)$$

Now this is a trivial direct integration with respect to y:

$$x(y) = \int (e^y + \cos(y))\, dy = e^y + \sin(y) + C$$

Apply the condition ($x = 2$ when $y = 0$):

$$\begin{aligned} 2 &= e^0 + \sin(0) + C \\ 2 &= 1 + 0 + C \quad \implies \quad C = 1 \end{aligned}$$

The final relationship is:

$$x(y) = e^y + \sin(y) + 1$$

Solution 2: Separation of Variables

Multiply both sides by ye^{y^2} and by dx to separate the variables into perfect differentials:

$$\begin{aligned} ye^{y^2} dy &= x dx \\ \int ye^{y^2} dy &= \int x dx \end{aligned}$$

Using u-substitution on the left side (let $u = y^2$, $du = 2ydy$):

$$\begin{aligned} \frac{1}{2} e^{y^2} &= \frac{1}{2} x^2 + C \\ e^{y^2} &= x^2 + C_2 \end{aligned}$$

Taking the natural log yields the general solution $y = \pm\sqrt{\ln(x^2 + C_2)}$.

Solution 3: Conservation of Energy

Following the autonomous ODE trick, multiply the equation by $v = \dfrac{dy}{dt}$:

$$m \frac{d^2y}{dt^2} \frac{dy}{dt} + ky \frac{dy}{dt} = 0$$

Recognize that these are perfect differentials:

$$\frac{d}{dt}\left[\frac{1}{2}m\left(\frac{dy}{dt}\right)^2\right] + \frac{d}{dt}\left[\frac{1}{2}ky^2\right] = 0$$

Integrating with respect to t yields:

$$\frac{1}{2}mv^2 + \frac{1}{2}ky^2 = C$$

Note: This is exactly the conservation of energy! By performing this mathematical transformation, you have just proved that the sum of kinetic energy ($\frac{1}{2}mv^2$) and potential energy ($\frac{1}{2}ky^2$) is equal to a constant total energy C.

Solution 4: A Solvable Autonomous ODE

Multiply by y' and recognize the perfect differentials:

$$y''y' = 6y^2y'$$
$$\frac{d}{dx}\left[\frac{1}{2}(y')^2\right] = \frac{d}{dx}\left[2y^3\right]$$

Integrate to get $\frac{1}{2}(y')^2 = 2y^3 + C$.
Apply the initial conditions $y(0) = 1$ and $y'(0) = -2$:

$$\frac{1}{2}(-2)^2 = 2(1)^3 + C \quad \Longrightarrow \quad 2 = 2 + C \quad \Longrightarrow \quad C = 0$$

So we have $(y')^2 = 4y^3$. Taking the square root gives $y' = \pm 2y^{3/2}$. Because the initial slope $y'(0)$ is -2 (it is negative!), we must take the negative root:

$$\frac{dy}{dx} = -2y^{3/2}$$

Separate the variables and integrate again:

$$y^{-3/2}dy = -2dx$$
$$-2y^{-1/2} = -2x + C_2$$
$$y^{-1/2} = x + C_3$$

Apply the initial condition $y(0) = 1$:

$$(1)^{-1/2} = 0 + C_3 \quad \Longrightarrow \quad C_3 = 1$$

So $y^{-1/2} = x + 1$. Squaring both sides and inverting yields the final solution:

$$y(x) = \frac{1}{(x+1)^2}$$

Solution 5: Going Backwards to Go Forwards

Differentiate the entire equation with respect to x, using the product rule on the first term:

$$\begin{aligned} \frac{d}{dx}(xy'' - y') &= \frac{d}{dx}(3x^2) \\ (y'' + xy''') - y'' &= 6x \\ xy''' &= 6x \quad \Longrightarrow \quad y''' = 6 \end{aligned}$$

The y'' terms cancel beautifully! Now we just integrate sequentially. First, to get y'':

$$y'' = 6x + C_1$$

Integrate again to get y':

$$y' = 3x^2 + C_1x + C_2$$

Because we differentiated the original ODE, we artificially raised its order from 2 to 3, which created the extra constant C_2. To find C_2, substitute y'' and y' back into the **original** ODE:

$$\begin{aligned} x(6x + C_1) - (3x^2 + C_1x + C_2) &= 3x^2 \\ 6x^2 + C_1x - 3x^2 - C_1x - C_2 &= 3x^2 \\ 3x^2 - C_2 &= 3x^2 \quad \Longrightarrow \quad C_2 = 0 \end{aligned}$$

So, our true first derivative is $y' = 3x^2 + C_1x$. Now, apply the initial condition $y'(1) = \frac{1}{2}$:

$$\frac{1}{2} = 3(1)^2 + C_1(1) \quad \Longrightarrow \quad C_1 = -\frac{5}{2}$$

So $y' = 3x^2 - \frac{5}{2}x$. Integrate one last time to find y:

$$y = x^3 - \frac{5}{4}x^2 + C_3$$

Apply the final initial condition $y(1) = 0$:

$$0 = 1^3 - \frac{5}{4}(1)^2 + C_3 \quad \Longrightarrow \quad C_3 = \frac{1}{4}$$

The final solution is:

$$y(x) = x^3 - \frac{5}{4}x^2 + \frac{1}{4}$$

Note: There was actually an even easier way to solve this problem. The original equation didn't involve y (just y′ and y″). Thus, if you substitute u = y′ you would get a first order ODE for u. There is often more than one way of solving an equation!

8 Transformation of the Dependent Variable

A last approach which is sometimes useful is the transformation of the dependent variable. Like the methods in the previous section, this is very problem-specific!

An excellent example: suppose we have a chunk of uranium (fissile material). The steady-state neutron flux $\phi(r)$ for a spherical mass is approximated by the following governing equation:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\phi}{dr}\right) + \lambda^2\phi + S = 0$$

Here, the first term arises from diffusion, the $\lambda^2\phi$ term represents the fission reaction (the production of neutrons proportional to the local flux), and S is a constant radioactive source. This equation has been rendered dimensionless for simplicity, so in the fission term λ is a dimensionless ratio involving the neutron reproduction factor, neutron diffusion, and the size of the sphere.

The boundary conditions for this dimensionless sphere are:

$$\phi|_{r=0} = \text{finite}$$
$$\phi|_{r=1} = 0 \quad \text{(all neutrons assumed to escape at the boundary)}$$

This is a linear, second order ODE. It is inhomogeneous due to the source term S. Because it is linear, we can invoke superposition and say the total solution is the sum of a homogeneous and a particular solution:

$$\phi = \phi_h + \phi_p$$

By inspection, the particular solution is simply a constant: $\phi_p = -\frac{S}{\lambda^2}$.

Substituting this back, the homogeneous equation is:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{d\phi_h}{dr}\right) + \lambda^2\phi_h = 0$$

The Transformation

We can solve this tricky spherical ODE by transforming the dependent variable. Let's make the substitution $\phi_h = \frac{f}{r}$.

First, take the derivative using the quotient rule:

$$\frac{d\phi_h}{dr} = \frac{d}{dr}\left(\frac{f}{r}\right) = \frac{1}{r}\frac{df}{dr} - \frac{1}{r^2}f$$

Multiply by r^2 to match the inside of the ODE's parentheses:

$$r^2\frac{d\phi_h}{dr} = r\frac{df}{dr} - f$$

Now, take the derivative of this entire group with respect to r (using the product rule on the first term):

$$\begin{aligned}\frac{d}{dr}\left(r^2\frac{d\phi_h}{dr}\right) &= \left(r\frac{d^2f}{dr^2} + \frac{df}{dr}\right) - \frac{df}{dr} \\ &= r\frac{d^2f}{dr^2}\end{aligned}$$

Substitute this back into the original homogeneous differential equation:

$$\begin{aligned}\frac{1}{r^2}\left(r\frac{d^2f}{dr^2}\right) + \lambda^2\left(\frac{f}{r}\right) &= 0 \\ \frac{1}{r}\frac{d^2f}{dr^2} + \lambda^2\frac{f}{r} &= 0\end{aligned}$$

Multiply the entire equation by r, and the ODE simplifies beautifully to:

$$\frac{d^2f}{dr^2} + \lambda^2 f = 0$$

Solving the Transformed Equation

This is a simple constant coefficient ODE (the harmonic oscillator), and we know the solution!

$$f(r) = A\sin(\lambda r) + B\cos(\lambda r)$$

Transforming back to our original dependent variable ϕ, and adding our particular solution back in:

$$\phi(r) = A\frac{\sin(\lambda r)}{r} + B\frac{\cos(\lambda r)}{r} - \frac{S}{\lambda^2}$$

Now we apply our boundary conditions. First, we know the flux must be finite at the center of the sphere ($r = 0$). Because $\cos(0) = 1$, the term $\frac{\cos(\lambda r)}{r}$ blows up to infinity at $r = 0$. To prevent this physically impossible result, we must set $B = 0$.

Next, apply the boundary condition at the edge of the sphere ($\phi = 0$ at $r = 1$):

$$0 = A\frac{\sin(\lambda \cdot 1)}{1} - \frac{S}{\lambda^2}$$
$$A = \frac{S}{\lambda^2 \sin(\lambda)}$$

Our final equation for the neutron flux is:

$$\phi(r) = \frac{S}{\lambda^2}\left(\frac{\sin(\lambda r)}{r\sin(\lambda)} - 1\right)$$

Analyzing the Limits (Critical Mass)

Using the Taylor series for sine ($\sin(x) \approx x - \frac{x^3}{6}$), we can look at the asymptote as $\lambda \to 0$ (the limit where fission is negligible compared to diffusion).

$$\lim_{\lambda\to 0}\phi = \frac{S}{\lambda^2}\left(\frac{\lambda r - \frac{1}{6}(\lambda r)^3}{r(\lambda - \frac{1}{6}\lambda^3)} - 1\right)$$
$$= \frac{S}{\lambda^2}\left(\frac{1 - \frac{1}{6}\lambda^2 r^2}{1 - \frac{1}{6}\lambda^2} - 1\right)$$

Using the binomial expansion $(1 - \epsilon)^{-1} \approx 1 + \epsilon$:

$$\approx \frac{S}{\lambda^2}\left[\left(1 - \frac{1}{6}\lambda^2 r^2\right)\left(1 + \frac{1}{6}\lambda^2\right) - 1\right]$$
$$\approx \frac{S}{\lambda^2}\left[1 + \frac{1}{6}\lambda^2 - \frac{1}{6}\lambda^2 r^2 - 1\right]$$
$$= \frac{1}{6}S(1 - r^2) + O(\lambda^2)$$

This parabolic profile is exactly what we expect for steady diffusion with a constant volumetric source!

A more interesting issue is what happens for larger λ (higher fission rates or a larger sphere). The highest flux is at the core ($r = 0$). Again, using a Taylor series, $\lim_{r\to 0}\frac{\sin(\lambda r)}{r} = \lambda$. Therefore, the core flux is:

$$\phi|_{r=0} = \frac{S}{\lambda^2}\left(\frac{\lambda}{\sin(\lambda)} - 1\right)$$

This is well-behaved for small λ. However, when $\lambda = \pi$, the denominator $\sin(\pi) = 0$, and the neutron flux grows without bound! **This represents the criticality parameter for the sphere of uranium.**

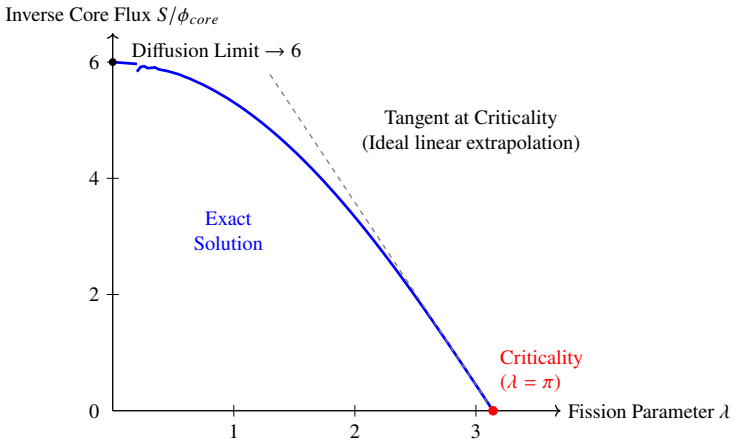


Figure 8.1: An "approach to critical" plot showing the inverse core flux (S/ϕ_{core}) as a function of the fission parameter λ. Plotting the flux directly is unrevealing as it asymptotes to infinity. By plotting the inverse of this multiplication factor, we determine the critical limit ($\lambda = \pi$). In the development of the first nuclear reactor Chicago Pile-1 Enrico Fermi plotted up the ratio in this way as a function of the number of layers of graphite and uranium - determining that criticality would be reached as the 57th layer was added to the pile.

Eliminating the First Derivative (The "Normal Form")

The substitution $\phi_h = f/r$ that we just used for the spherical reactor might seem like a lucky guess. However, it is actually a specific instance of a universal mathematical rule!

For *any* general linear second-order ODE of the form:

$$y'' + P(x)y' + Q(x)y = 0$$

You can **always** completely eliminate the first derivative term (y') by using the following dependent variable transformation:

$$y(x) = u(x) \exp\left(-\frac{1}{2}\int P(x)\,dx\right)$$

The transformed equation becomes:

$$u'' + \left(Q - \frac{1}{2}P' - \frac{1}{4}P^2\right)u = 0$$

Let's verify this with our spherical diffusion operator. The r-derivative in our homogeneous equation expanded to:

$$\frac{d^2\phi_h}{dr^2} + \frac{2}{r}\frac{d\phi_h}{dr} + \lambda^2\phi_h = 0$$

Here, $P(r) = \frac{2}{r}$. If we plug this into our universal transformation formula:

$$\begin{aligned}\phi_h(r) &= u(r) \exp\left(-\frac{1}{2}\int \frac{2}{r}\, dr\right) \\ &= u(r) \exp\left(-\ln r\right) \\ &= u(r)\left(r^{-1}\right) = \frac{u(r)}{r}\end{aligned}$$

The guess of f/r was just the standard Normal Form transformation! Whenever you are faced with a nasty linear second-order ODE, computing this exponential integrating factor is often a useful way to simplify the problem, although the function $\left(Q - \frac{1}{2}P' - \frac{1}{4}P^2\right)$ multiplying u may still render solution difficult. For this problem, though, it was just a constant!

Bernoulli Differential Equations

Another classic dependent variable transformation is used to linearize a specific family of non-linear equations known as **Bernoulli equations**. These equations take the form:

$$\frac{dy}{dt} + P(t)y = Q(t)y^n$$

Because of the y^n term on the right-hand side, the equation is non-linear (assuming $n \neq 0$ and $n \neq 1$). However, you can perfectly linearize it with the substitution:

$$v = y^{1-n}$$

Let's look at a physical example. In biochemical engineering, the growth of a bacteria population $P(t)$ in a batch bioreactor is proportional to the current population (growth rate k). However, as the bacteria consume the limited nutrients, a "crowding" or death effect emerges that is proportional to the square of the population (death rate c). This is the logistic growth model:

$$\frac{dP}{dt} = kP - cP^2$$

Rearranging this to match the Bernoulli form:

$$\frac{dP}{dt} - kP = -cP^2$$

We can see that $n = 2$. Our Bernoulli transformation dictates that we should substitute $v = P^{1-2} = P^{-1}$.

First, let's find the derivative of our substitution with respect to t using the chain rule:

$$\frac{dv}{dt} = -P^{-2}\frac{dP}{dt}$$

Now, divide the original non-linear ODE entirely by P^2:

$$P^{-2}\frac{dP}{dt} - kP^{-1} = -c$$

Substitute v and dv/dt into the equation:

$$-\frac{dv}{dt} - kv = -c$$
$$\frac{dv}{dt} + kv = c$$

The non-linearity has been completely eliminated! We now have a standard, linear first-order ODE. The solution to this equation (using a standard integrating factor e^{kt}) is:

$$v(t) = Ce^{-kt} + \frac{c}{k}$$

Finally, we transform back to our original dependent variable $P(t) = v^{-1}$:

$$P(t) = \frac{1}{Ce^{-kt} + \frac{c}{k}}$$

If we define our initial population at $t = 0$ as $P(0) = P_0$, we can solve for C:

$$P_0 = \frac{1}{C + \frac{c}{k}} \quad \implies \quad C = \frac{1}{P_0} - \frac{c}{k}$$

Plugging this back in and multiplying the numerator and denominator by kP_0 gives the standard logistic growth curve:

$$P(t) = \frac{kP_0}{cP_0 + (k - cP_0)e^{-kt}}$$

This analytical solution yields the famous "S-curve" (sigmoid) of biological population growth, leveling out exactly at the carrying capacity $P_{max} = k/c$ as $t \to \infty$.

Final Note on Transformations: In general, these tricks sometimes enable you to get an analytical solution to more complicated ODEs. If they don't work, numerical solutions are required.

Sometimes you can also simplify a differential equation by transforming the *independent* variable. The conversion of Euler equations to constant coefficient ODEs via the $x = e^t$ transform is a classic example. Independent variable transforms are used much more often for Partial Differential Equations (PDEs)—such as going from Cartesian coordinates to cylindrical coordinates—as that sometimes lets you convert a PDE directly into an ODE, which is a huge simplification. We will look at this later!

Practice Problems

Test your skills on these problems. Because these are non-standard ODEs, pay close attention to the hints. Your goal is to find the dependent variable transformation that either eliminates a troublesome derivative or completely linearizes a non-linear equation!

Problem 1: Finding the Normal Form

Consider the following variable-coefficient second-order linear ODE:

$$y'' + 2xy' + (x^2 + 2)y = 0$$

with the initial conditions $y(0) = 1$ and $y'(0) = 0$.

Use the universal Normal Form transformation $y(x) = u(x) \exp\left(-\frac{1}{2}\int P(x)\,dx\right)$ to eliminate the first derivative term, solve the resulting simplified ODE for $u(x)$, and transform back to find the explicit solution for $y(x)$.
Hint: The transformed equation will simplify into a beautifully classic constant-coefficient harmonic oscillator.

Problem 2: Atomic Recombination in a Purged Vacuum Chamber (Bernoulli Equation)

In an experimental vacuum chamber simulating interstellar conditions, a cloud of free hydrogen atoms undergoes a three-body recombination process. Because three atoms must collide simultaneously, the free atoms decay at a third-order rate dictated by the physical constant r. Simultaneously, an active vacuum pump continuously purges the chamber, removing the gas at a first-order volumetric flow rate k.

The transient concentration profile $C(t)$ of the free hydrogen atoms is governed by the non-linear equation:

$$\frac{dC}{dt} + kC = -rC^3$$

where k and r are positive physical constants, and the initial concentration at $t = 0$ is $C(0) = C_0$.

Identify this as a Bernoulli equation, apply the substitution $v = C^{1-n}$ to completely linearize it, solve for $v(t)$ using an integrating factor, and show that the explicit concentration over time is given by:

$$C(t) = \left[\left(C_0^{-2} + \frac{r}{k}\right)e^{2kt} - \frac{r}{k}\right]^{-1/2}$$

Problem 3: Fractional Bessel Equations and Cylindrical Waves

Just as spherical coordinates prompted the $\phi_h = f/r$ substitution in our notes, solving transport equations in cylindrical coordinates often prompts fractional transformations. Consider the following linear ODE:

$$x^2y'' + xy' + \left(x^2 - \frac{1}{4}\right)y = 0$$

Transform this equation using the dependent variable substitution $y(x) = \dfrac{u(x)}{\sqrt{x}}$ to eliminate the first derivative, and find the general solution for $y(x)$.
Hint: Write the substitution as $y = ux^{-1/2}$ *and use the product rule to calculate* y' *and* y'' *before plugging them into the ODE.*

Solutions to Practice Problems

Solution 1: Finding the Normal Form

We are given the ODE $y'' + 2xy' + (x^2 + 2)y = 0$, meaning $P(x) = 2x$ and $Q(x) = x^2 + 2$. First, compute the exponential integrating factor for our dependent variable transformation:

$$\exp\left(-\frac{1}{2}\int P(x)\,dx\right) = \exp\left(-\frac{1}{2}\int 2x\,dx\right) = \exp\left(-\frac{1}{2}x^2\right) = e^{-x^2/2}$$

Thus, our transformation is $y(x) = u(x)e^{-x^2/2}$. Let's compute the derivatives using the product rule:

$$y' = u'e^{-x^2/2} - xue^{-x^2/2} = (u' - xu)e^{-x^2/2}$$
$$y'' = (u'' - u'x - u - xu')e^{-x^2/2} - x(u' - xu)e^{-x^2/2} = (u'' - 2xu' + (x^2 - 1)u)e^{-x^2/2}$$

Substitute y, y', and y'' back into the original ODE:

$$\left[u'' - 2xu' + (x^2 - 1)u\right] e^{-x^2/2} + 2x\left[u' - xu\right] e^{-x^2/2} + (x^2 + 2)\left[u\right] e^{-x^2/2} = 0$$
$$\left[u'' - 2xu' + x^2u - u + 2xu' - 2x^2u + x^2u + 2u\right] e^{-x^2/2} = 0$$

Grouping like terms inside the brackets reveals that the u' terms perfectly cancel and the u terms simplify nicely:

$$u'' + u = 0$$

This is the simple harmonic oscillator equation. Its general solution is:

$$u(x) = A\cos(x) + B\sin(x)$$

Transforming back to our original variable $y(x)$:

$$y(x) = (A\cos(x) + B\sin(x))e^{-x^2/2}$$

Now we apply our initial conditions to evaluate the constants. First, $y(0) = 1$:

$$1 = (A\cos(0) + B\sin(0))e^0 \quad \Longrightarrow \quad A = 1$$

Next, find $y'(x)$ to apply the second initial condition:

$$y'(x) = (-\sin(x) + B\cos(x))e^{-x^2/2} - x(\cos(x) + B\sin(x))e^{-x^2/2}$$

Apply $y'(0) = 0$:

$$0 = (0 + B \cdot 1)e^0 - 0 \quad \Longrightarrow \quad B = 0$$

The final explicit solution is:

$$y(x) = \cos(x)e^{-x^2/2}$$

Solution 2: Atomic Recombination in a Purged Vacuum Chamber (Bernoulli Equation)

The equation is $\frac{dC}{dt} + kC = -rC^3$. This matches the Bernoulli form $y' + P(t)y = Q(t)y^n$ where $n = 3$. Our change of variable must be:

$$v = C^{1-3} = C^{-2}$$

Differentiating v with respect to t via the chain rule yields:

$$\frac{dv}{dt} = -2C^{-3}\frac{dC}{dt} \quad \Longrightarrow \quad C^{-3}\frac{dC}{dt} = -\frac{1}{2}\frac{dv}{dt}$$

Now, divide our governing ODE entirely by C^3:

$$C^{-3}\frac{dC}{dt} + kC^{-2} = -r$$

Substitute our expressions for v and dv/dt into this equation:

$$-\frac{1}{2}\frac{dv}{dt} + kv = -r \quad \Longrightarrow \quad \frac{dv}{dt} - 2kv = 2r$$

This is now a linear first-order ODE. The integrating factor is $I(t) = e^{-2kt}$. Multiplying through:

$$\begin{aligned} \frac{d}{dt}\left(ve^{-2kt}\right) &= 2re^{-2kt} \\ ve^{-2kt} &= \int 2re^{-2kt}\,dt = -\frac{r}{k}e^{-2kt} + A \\ v(t) &= -\frac{r}{k} + Ae^{2kt} \end{aligned}$$

Transform back to our concentration variable $C = v^{-1/2}$:

$$C(t) = \left(Ae^{2kt} - \frac{r}{k}\right)^{-1/2}$$

Apply the initial condition $C(0) = C_0$:

$$C_0 = \left(Ae^0 - \frac{r}{k}\right)^{-1/2} \quad \Longrightarrow \quad C_0^{-2} = A - \frac{r}{k} \quad \Longrightarrow \quad A = C_0^{-2} + \frac{r}{k}$$

Plugging A back into our equation gives the final explicit solution:

$$C(t) = \left[\left(C_0^{-2} + \frac{r}{k}\right)e^{2kt} - \frac{r}{k}\right]^{-1/2}$$

Solution 3: Fractional Bessel Equations and Cylindrical Waves

We are given $x^2y'' + xy' + \left(x^2 - \frac{1}{4}\right)y = 0$, and the substitution $y = ux^{-1/2}$. Let's calculate its derivatives using the product rule:

$$y' = u'x^{-1/2} - \frac{1}{2}ux^{-3/2}$$
$$y'' = u''x^{-1/2} - \frac{1}{2}u'x^{-3/2} - \frac{1}{2}u'x^{-3/2} + \frac{3}{4}ux^{-5/2} = u''x^{-1/2} - u'x^{-3/2} + \frac{3}{4}ux^{-5/2}$$

Now, plug y, y', and y'' back into the original differential equation:

$$x^2\left(u''x^{-1/2} - u'x^{-3/2} + \frac{3}{4}ux^{-5/2}\right) + x\left(u'x^{-1/2} - \frac{1}{2}ux^{-3/2}\right) + \left(x^2 - \frac{1}{4}\right)ux^{-1/2} = 0$$

Distributing the x^2 and x through the terms yields:

$$\left(u''x^{3/2} - u'x^{1/2} + \frac{3}{4}ux^{-1/2}\right) + \left(u'x^{1/2} - \frac{1}{2}ux^{-1/2}\right) + \left(ux^{3/2} - \frac{1}{4}ux^{-1/2}\right) = 0$$

Let's group like terms by their powers of x:

$$u''x^{3/2} + \underbrace{\left(-u'x^{1/2} + u'x^{1/2}\right)}_{=0} + \underbrace{\left(\frac{3}{4} - \frac{1}{2} - \frac{1}{4}\right)ux^{-1/2}}_{=0} + ux^{3/2} = 0$$

The first derivative terms and the fractional inverse-root terms perfectly drop out of the problem, leaving only:

$$u''x^{3/2} + ux^{3/2} = 0$$

Dividing out $x^{3/2}$ gives the beautifully simple harmonic oscillator equation:

$$u'' + u = 0$$

The general solution for our transformed variable is:

$$u(x) = A\cos(x) + B\sin(x)$$

Transforming back to find our final general solution for $y(x)$:

$$y(x) = \frac{A\cos(x) + B\sin(x)}{\sqrt{x}}$$

Part II

Computational Solutions for ODEs

9 Numerical Solutions to ODEs

In general, analytic solutions to ODEs are preferable because you learn a lot more about the problem from the final mathematical form (e.g., the critical mass asymptote we found in the previous section). Often, however (particularly for non-linear ODEs), an analytic solution either doesn't exist or is simply too hard to get! Thus, we solve such ODEs numerically.

Most solution techniques are based on solving an ODE as a system of first-order equations.

Systems of First-Order Equations

A classic example is from population dynamics: the Lotka-Volterra model for rabbits and foxes.

Let r = rabbits, and f = foxes.

We have coupled 1st-order equations:

$$\frac{dr}{dt} = 2r - \alpha r f \quad \leftarrow \text{(reproduction rate − foxes eat bunnies)}$$
$$\frac{df}{dt} = (\alpha r - 1) f \quad \leftarrow \text{(more bunnies yields more foxes − foxes starve without food)}$$

To format this for a numerical solver, we combine them into a state vector. Let $y_1 = r$ and $y_2 = f$:

$$\frac{dy}{dt} = \begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \end{bmatrix} = \begin{bmatrix} 2y_1 - \alpha y_1 y_2 \\ (\alpha y_1 - 1) y_2 \end{bmatrix} \equiv g(t, y)$$

This system $\frac{dy}{dt} = g(t, y)$ can be integrated numerically from some initial condition to demonstrate the fascinating oscillatory population dynamics that are observed in many natural systems.

Transforming High-Order ODEs

This approach works just as well for high-order ODEs. Consider the Falkner-Skan equation for boundary-layer flow past a wedge. This equation arises after applying a similarity transformation to the Navier–Stokes boundary-layer equations:

$$f''' + f f'' + \beta(1 - (f')^2) = 0$$

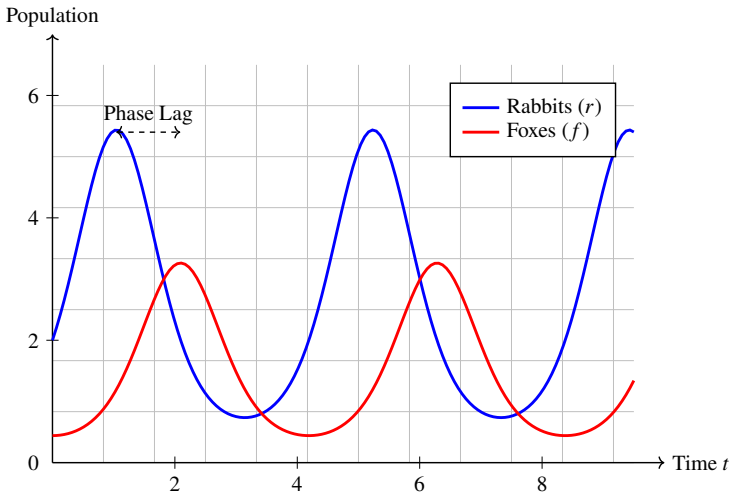


Figure 9.1: Oscillatory population dynamics typical of the Lotka-Volterra predator-prey model. The prey population (rabbits) expands rapidly until checked by the growing predator population (foxes). The subsequent collapse of the prey leads to the starvation of the predators, allowing the cycle to begin anew. Notice the non-linear "shark fin" shape of the curves and the characteristic phase lag between the two populations.

with boundary conditions: $f(0) = 0$, $f'(0) = 0$, and $f'(\infty) = 1$.

If we have a third-order equation, we need three 1st-order equations. We define our state variables as the function and its consecutive derivatives:

$$\begin{aligned} y_1 &\equiv f \\ y_2 &\equiv f' \\ y_3 &\equiv f'' \quad \leftarrow \text{stop here!} \end{aligned}$$

So by definition:

$$\begin{aligned} \frac{dy_1}{d\eta} &= f' = y_2 \\ \frac{dy_2}{d\eta} &= f'' = y_3 \end{aligned}$$

And from the original ODE, we isolate the highest derivative:

$$\begin{aligned} \frac{dy_3}{d\eta} &= f''' = -ff'' - \beta(1 - (f')^2) \\ &= -y_1 y_3 - \beta(1 - y_2^2) \end{aligned}$$

Putting this into vector form $\dfrac{dy}{d\eta} = g(\eta, y)$:

$$\frac{dy}{d\eta} = \begin{bmatrix} y_2 \\ y_3 \\ -y_1 y_3 - \beta(1 - y_2^2) \end{bmatrix}$$

The Shooting Method

Most numerical integrators start from an initial condition at $t = 0$ and integrate strictly forward in time (or space). Because we have a 3rd-order system, we need three initial conditions. We know two of them:

$$y_1(0) = 0$$
$$y_2(0) = 0$$

However, $y_3(0)$ is unknown! Let's call it x. Instead of a third condition at the wall, we have a condition at the far-field boundary: $f'(\infty) \equiv y_2(\infty) = 1$.

We solve this using the **Shooting Method**. We guess $y_3(0) = x$ (picking a reasonable physical value, like the estimated wall shear stress) and integrate forward to a large t (or η). We then adjust our guess x until our far-field condition is met, specifically when $y_2(\infty) - 1 = 0$. This iterative guessing is done automatically using a root-finder!

For higher-order equations (e.g., natural convection from a heated wire) you may have two or more shooting parameters. This would yield a vector of "misses" $P(x) = 0$ to satisfy the boundary conditions on the other boundary. You can solve this via multidimensional root-finding, or sometimes it is more numerically stable to minimize $||P(x)||^2$ as an optimization problem—whatever works!

Implementation in MATLAB

As an example, this optimization approach is implemented for the Falkner-Skan problem in the short function `fsmiss.m`. It can be run directly from the MATLAB command line by passing the function to a minimizer with an initial guess (e.g., guessing $f''(0) = 1$):

```
» fminsearch('fsmiss', 1)
```

```
function out=fsmiss(x)
% This function integrates the Falkner-Skan equation and returns zero when
% the boundary condition at infinity is satisfied.  The unknown argument is
% the stress at the wall f''(0).  It can be called from the command line
% using the function minimization routine fminsearch('fsmiss',1).  We use a
% combined output including both the "miss" at infinity as well as the
% second derivative (which should vanish at infinity as well).  This
% provides a more robust convergence for larger values of beta.
% Numerically, we replace infinity with a sufficiently large finite cutoff where the
% solution has effectively converged.

beta = .5; % We pick a value of beta corresponding to a 90 degree wedge.

fdot = @(t,f) [f(2); f(3); -f(1)*f(3) - beta*(1-f(2)^2)];
% MATLAB still requires a dummy independent variable in the function definition
% even though here it is not contained in the actual derivatives.

f0 = [0;0;x]; % The initial condition

[etaout fout] = ode23(fdot,[0 8],f0); % We integrate only to 8 due to rapid decay.

figure(1)
plot(etaout,fout(:,2))
xlabel('eta')
ylabel('scaled velocity')
drawnow

out = norm([fout(end,2)-1,fout(end,3)]);
```

Note: The code above was written for MATLAB. However, it will run perfectly without modification in GNU Octave, which is a free, open-source alternative. If you prefer to work in Python, simply ask your favorite AI to translate this script for you!

On the Use of AI for Coding

With the rapid development of coding prowess by various AI models, a question naturally arises: Why should a student learn anything about numerical methods? The answer is complex, but it basically boils down to getting a cleaner, more reliable, and more accurate result—because, as we all know, LLMs do make mistakes.

For the Falkner-Skan problem above, you can get away with using a "vibe coding" approach. Suppose we want to calculate both the wall shear stress and the displacement thickness (the latter is defined as the integral of $(1 - f')$ from zero to ∞). A prompt such as the following:

> *"Write a MATLAB code which solves the Falkner-Skan equations for flow past a 90 degree wedge, calculating the wall shear stress and the displacement thickness."*

This will actually work, but the AI (Gemini, in this case) will do a couple of things that may not be

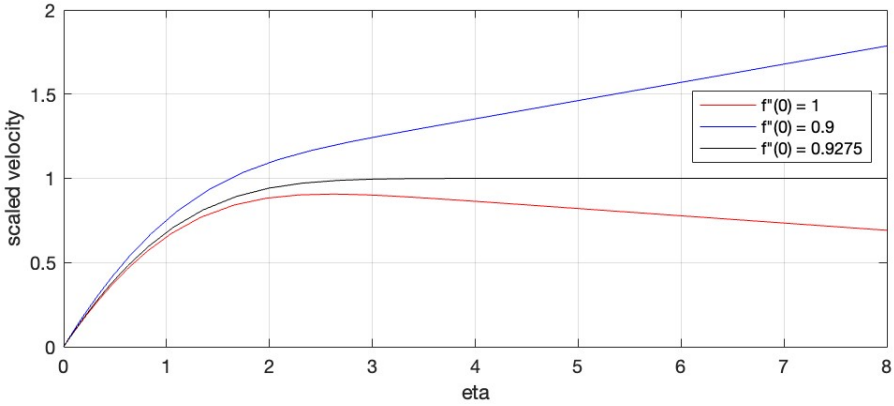


Figure 9.2: An illustration of the Shooting Method applied to the Falkner-Skan equation for flow past a 90-degree wedge. Because the initial wall shear stress $f''(0)$ is unknown, it acts as our "shooting parameter." The integrator fires numerical trajectories using different initial guesses to see where they land at the far-field boundary. An initial guess of $f''(0) = 1.0$ overshoots the target velocity of $f'(\infty) = 1$, while a guess of $f''(0) = 0.9$ undershoots it. By treating the final miss distance as an optimization problem, the root-finder quickly converges on the exact required initial condition of $f''(0) \approx 0.9277$.

desirable. First, it will treat it as a boundary value problem (and actually sets a reasonable upper limit of $\eta = 8$), but it will use the built-in MATLAB function **`bvp4c.m`**. This works, but unlike more basic routines, it would not be compatible with the free Octave clone. You would need to explicitly instruct the AI to use the shooting method, or to write the code using only functions compatible with both MATLAB and Octave.

A more serious issue is the way in which the AI calculates the displacement thickness. Invariably, it solves the differential equation first, and then goes back and uses the Trapezoidal Rule and interpolation to get the final integral—at a loss of accuracy in the fourth decimal place, and a significant loss of elegance.

The optimal way of doing this is to simply track the integral as a fourth dependent variable in our system of ODEs. If we define $y_4 = \int (1 - f')\, d\eta$, we just add its derivative to our state vector:

$$\frac{dy_4}{d\eta} = 1 - f' = 1 - y_2$$

In general, the more you know about the math behind the problem, the more accurate your code will be. You can guide the AI to "solve it the way I want," and even have it analyze the pros and cons of different solution approaches to find the best method. However, it is extremely useful to have a foundational level of understanding of both the mathematics and the numerical methods to place the AI-produced code in its proper context.

Practice Problems

Test your skills on these problems. Before you can write a script in MATLAB or Python to solve a differential equation, you must be able to translate the physical problem into the rigid mathematical structure that numerical integrators expect!

Problem 1: State Vectors for High-Order ODEs

The Blasius boundary layer equation describes the velocity profile of fluid flowing over a flat plate. It is a non-linear, third-order ODE given by:

$$f''' + f f'' = 0$$

with boundary conditions $f(0) = 0$, $f'(0) = 0$, and $f'(\infty) = 1$.

Convert this third-order equation into a system of three first-order ODEs in the form $\frac{dy}{d\eta} = g(\eta, y)$. Explicitly write out the vector $g(\eta, y)$ and state which initial condition is unknown (your shooting parameter).

Problem 2: Coupled First-Order Systems

The SIR model is a classic set of coupled ordinary differential equations used in epidemiology to model the spread of a virus. The population is divided into Susceptible (S), Infected (I), and Recovered (R). The dynamics are governed by:

$$\begin{aligned} \frac{dS}{dt} &= -aSI \\ \frac{dI}{dt} &= aSI - bI \\ \frac{dR}{dt} &= bI \end{aligned}$$

where a is the infection rate and b is the recovery rate.

Define a state vector y, and write this entire system as a single vector equation $\frac{dy}{dt} = g(t, y)$.

Problem 3: Setting up the Shooting Method

Consider the steady-state temperature profile of a finite cooling fin with thermal radiation in a vacuum. Unlike the infinite fin we solved earlier, this fin has a physical length L. The temperature $T(x)$ is governed by:

$$\frac{d^2T}{dx^2} - \lambda T^4 = 0$$

with boundary conditions $T(0) = T_0$ (the hot base) and the zero heat flux condition $\frac{dT}{dx} = 0$ at the end $x = L$.

(a) Convert this 2nd-order ODE into a system of first-order ODEs.

(b) Explain exactly how you would set up the Shooting Method to solve this numerically. Specifically identify your guessed initial condition x_{guess} and define the mathematical "miss" function $P(x_{guess})$ that your root-finding algorithm will try to drive to zero.

Solutions to Practice Problems

Solution 1: State Vectors for High-Order ODEs

We have a third-order equation, so we need to define three state variables corresponding to f and its derivatives:

$$y_1 \equiv f$$
$$y_2 \equiv f'$$
$$y_3 \equiv f''$$

Taking the derivative of each with respect to the independent variable η:

$$\frac{dy_1}{d\eta} = f' = y_2$$
$$\frac{dy_2}{d\eta} = f'' = y_3$$

For the third derivative, we rearrange the original Blasius equation:

$$\frac{dy_3}{d\eta} = f''' = -ff'' = -y_1 y_3$$

Putting this all together into the state vector format $\frac{dy}{d\eta} = g(\eta, y)$:

$$\frac{dy}{d\eta} = \begin{bmatrix} y_2 \\ y_3 \\ -y_1 y_3 \end{bmatrix}$$

Based on our boundary conditions, we know $y_1(0) = 0$ and $y_2(0) = 0$. We do not know the initial second derivative at the wall, so $y_3(0)$ **is our unknown shooting parameter**.

Solution 2: Coupled First-Order Systems

Because the system is already a set of first-order ODEs, we do not need to introduce any new derivatives. We simply map the physical variables to a state vector y. Let:

$$y_1 \equiv S$$
$$y_2 \equiv I$$
$$y_3 \equiv R$$

Now, we substitute these y variables into the original system of equations to build our vector $g(t, y)$:

$$\frac{dy}{dt} = \begin{bmatrix} \frac{dy_1}{dt} \\ \frac{dy_2}{dt} \\ \frac{dy_3}{dt} \end{bmatrix} = \begin{bmatrix} -ay_1 y_2 \\ ay_1 y_2 - by_2 \\ by_2 \end{bmatrix} \equiv g(t, y)$$

This single vector equation can now be passed directly into `ode23` (MATLAB) or `solve_ivp` (Python)!

Solution 3: Setting up the Shooting Method

(a) We have a second-order ODE, so we define two state variables. Let $y_1 \equiv T$ and $y_2 \equiv \frac{dT}{dx}$. Taking the derivatives:

$$\frac{dy_1}{dx} = \frac{dT}{dx} = y_2$$
$$\frac{dy_2}{dx} = \frac{d^2T}{dx^2} = \lambda T^4 = \lambda y_1^4$$

The state vector formulation is:

$$\frac{dy}{dx} = \begin{bmatrix} y_2 \\ \lambda y_1^4 \end{bmatrix}$$

(b) To solve this using the Shooting Method, we must integrate from $x = 0$ to $x = L$. We are given the initial temperature $y_1(0) = T_0$. However, we do not know the initial temperature gradient at the base of the fin.

Therefore, our guess is the initial slope: $x_{guess} = y_2(0)$.

We use an ODE solver to integrate our system from $x = 0$ to $x = L$ using the initial condition vector $[T_0, x_{guess}]^T$. The solver will output both the temperature $y_1(L)$ and the derivative at the tip of the fin, which is $y_2(L)$. The true boundary condition requires that the derivative of the temperature at the tip vanishes.

Thus, we define our "miss" function as the difference between our numerical result and the required boundary condition:

$$P(x_{guess}) = y_2(L) - 0$$

We wrap this entire process in a root-finding algorithm (like `fzero` or `fminsearch`), which will systematically tweak our guess for the initial slope x_{guess} until $P(x_{guess}) = 0$.

10 Runge-Kutta Integration

In general, you are likely to be using "canned solvers" for obtaining a numerical solution to an ODE, but it is useful to understand how they work - and where they run into problems. Most integrators are fundamentally based on a Taylor series approximation to the function being integrated. Suppose we have the 1st-order ODE:

$$\frac{dy}{dt} = f(t, y)$$

We want to evaluate y at a set of discrete locations $t_0, t_1, \ldots, t_n$. Our approximations are $y_0, y_1, \ldots, y_n$.

The simplest approach is the **Euler Method**:

$$y_{k+1}^{EM} = y_k^{EM} + f(t_k, y_k^{EM}) \cdot h_k$$

where $h_k = t_{k+1} - t_k$.

This is just the first two terms of the Taylor Series. What is the error in this method? Use the Taylor Series again!

$$y(t_{k+1}) = y(t_k) + f(t_k, y(t_k))h_k + \frac{1}{2}y''(\xi)h_k^2$$

Subtracting the exact solution from our numerical approximation:

$$y_{k+1}^{EM} - y(t_{k+1}) = y_k^{EM} - y(t_k) + \left[f(t_k, y_k^{EM}) - f(t_k, y(t_k))\right] h_k - \frac{1}{2}y''(\xi)h_k^2$$

Now, from a Taylor Series expansion in y, we know:

$$f(t_k, y_k^{EM}) \approx f(t_k, y(t_k)) + \left.\frac{\partial f}{\partial y}\right|_{(t_k, y(t_k))} \left(y_k^{EM} - y(t_k)\right)$$

Substituting this back into our error equation yields:

$$y_{k+1}^{EM} - y(t_{k+1}) = \underbrace{\left(1 + h_k \frac{\partial f}{\partial y}\right)}_{\text{amplification factor}} (y_k^{EM} - y(t_k)) - \underbrace{\frac{1}{2}y''(\xi)h_k^2}_{\text{local error}}$$

So at each step we have a **local error** which is $O(h^2)$. Because the error is additive and the total number of steps is $O(1/h)$, the **overall (global) accuracy** is $O(h)$ (first order).

The amplification factor is $1 + hJ$, where J is the Jacobian $\frac{\partial f}{\partial y}$. This amplifies our error from previous steps! Note that if $hJ < -2$ (e.g., we have a stiff equation), our solution is numerically unstable. This is a problem for all explicit methods. An explicit method is one in which the value of y_{k+1} is explicitly calculated from information at (t_k, y_k).

You can avoid these strict stability limits by using an implicit method, such as the **Backward Euler Method**:

$$y_{k+1}^{BE} = y_k^{BE} + hf(t_{k+1}, y_{k+1}^{BE})$$

The error in this method can be calculated from the Taylor series in the same manner as before. The local truncation error is still $O(h^2)$ at each step, but the amplification factor is $\frac{1}{1 - hJ}$. This means that the method is absolutely stable for all $hJ < 0$, regardless of how large the step size h is. Because the unknown y_{k+1}^{BE} appears inside the function f, however, it requires solving a (usually non-linear) equation at each step. Non-linear solvers (such as the Newton-Raphson method) require a good initial guess to converge efficiently—a guess that the simple, explicit Euler method is perfectly suited to provide.

Runge-Kutta Methods

The simplest extension of the Euler Method is a 2-stage Runge-Kutta rule. This is based on the Trapezoidal Rule integration method given by:

$$y_{k+1}^{TR} \approx y_k^{TR} + \frac{h}{2}\left[f(t_k, y_k^{TR}) + f(t_{k+1}, y_{k+1}^{TR})\right]$$

The Trapezoidal Rule itself is also implicit (requiring solution to an implicit equation at each step), but is more accurate than the Backward Euler method.

The equivalent Runge-Kutta method uses an explicit 2-stage approximation where the derivative at y_{k+1} is calculated from the explicit Euler Method rule:

$$\begin{aligned} k_1 &= hf(t_i, y_i) \\ k_2 &= hf(t_{i+1}, \underbrace{y_i + k_1}_{\text{EM est. of } y_{i+1}}) \\ y_{i+1}^{2S} &= y_i^{2S} + \frac{1}{2}(k_1 + k_2) \end{aligned}$$

This is explicit, has a local error of $O(h^3)$, and a global error of $O(h^2)$.

This can be extended to higher orders. A popular 4-stage rule is:

$$
\begin{aligned}
k_1 &= hf(t_i, y_i) \\
k_2 &= hf\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_1\right) \\
k_3 &= hf\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}k_2\right) \\
k_4 &= hf(t_i + h, y_i + k_3) \\
y_{i+1}^{4S} &= y_i^{4S} + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

which has a local error of $O(h^5)$ and an overall error of $O(h^4)$ — and it is still explicit. Like all explicit methods, however, it will become unstable for sufficiently stiff ($J << 0$) equations.

While the above discussion was for first order ODEs, the exact same approaches can be used for higher order ODEs that are converted into systems of ODEs, with the same effects on accuracy and stability. In the latter case, the stability simply depends on the eigenvalues of the Jacobian (now a matrix). In general, systems of equations tend to have more stability issues than single ODEs. As we shall see later, this is particularly true for explicit finite difference solutions of parabolic PDEs by marching forward in time, the easiest way to approach such problems.

Adaptive Integration in MATLAB

While Runge-Kutta methods are easy to code up, it is more usual to use canned adaptive integration methods. An adaptive method is one that automatically changes the step size h to satisfy an error tolerance. For a Runge-Kutta based method such as MATLAB's `ode23` a three stage rule is employed, with the extra stage used to provide a higher accuracy estimate of the integration at each step. The local accuracy (and thus correction of the step size) is based on the difference between the results of the two and three stage rules.

In MATLAB, the codes used most often are:

- `ode23` (low order, explicit, generates more points for plotting)
- `ode45` (high order, explicit)
- `ode23s` (or `ode15s`) (implicit methods for stiff equations with a large, negative Jacobian)

All three require the specification of the range of integration $[t_0, t_f]$ and initial conditions, as well as the derivative function.

If you are using the shooting method, the 1-D root finder in MATLAB is:

- `fzero`

and the multi-dimensional optimization routine is:

- `fminsearch`

Use the `help` command to get examples of how to format your inputs!

Boundary Value Problem (BVP) Solvers

While the Runge-Kutta methods and the shooting method described above are powerful, MATLAB and Python also have dedicated solvers for Boundary Value Problems (such as the Falkner-Skan equation).

In MATLAB, the primary solver is:

- `bvp4c` (or `bvp5c`)

Unlike the shooting method (which converts the BVP into an Initial Value Problem), `bvp4c` uses a **collocation method**. It generates a mesh of points across the entire integration domain and solves for the solution at all points simultaneously using a system of algebraic equations. This requires you to provide an initial guess for the solution across the entire domain, rather than just guessing a single initial condition at the wall.

AI and Solver Selection

As discussed in the previous section, AI tools like ChatGPT or Gemini can easily write numerical integration scripts for you. However, LLMs have a strong bias toward using the most famous generic solvers—typically `ode45` for IVPs and `bvp4c` for BVPs.

If your mathematical model is computationally "stiff" (i.e., it has a large negative Jacobian, causing rapid transients alongside slow decay), an AI's default choice of `ode45` may fail. The script will either run infinitely slowly as the adaptive step-size h shrinks to near-zero, or it will become numerically unstable. For non-stiff equations, the step size is usually limited by accuracy requirements. For stiff equations, explicit methods are often limited instead by stability requirements.

Recognizing that a system is stiff and explicitly prompting the AI to "rewrite the integration using `ode15s`" is why it is important to understand the properties of the equations you are trying to solve. Implicit methods are more robust - but are also much slower to execute. But when you need them, you really need them...

Why Code it Yourself?

As stated at the beginning of the chapter, you will usually want to use a built-in solver when integrating a differential equation. There are specific scenarios, however, where this is not the best choice. Probably the most common case is when you are doing a highly vectorized calculation, such as tracking the motion of a large ensemble of particles in a flow field. Using an adaptive integrator like **ode23** would be highly inefficient for such a problem; the solver's global error control means that if even one particle requires a small time-step, the integration slows down for the entire system. Writing custom code is also a necessity if your system includes a stochastic driving force, such as Brownian motion. Standard ODE solvers are fundamentally unequipped to handle the non-differentiable nature of Stochastic Differential Equations (SDEs). We will examine how to tackle these specific problems in the upcoming chapter on Monte Carlo integration.

Practice Problems

Test your skills on these problems. Understanding the mechanics behind these numerical methods is the key to knowing why a simulation crashes or takes forever to run!

Problem 1: Stability of the Explicit Euler Method

Consider the simple radioactive decay of an isotope governed by the differential equation:

$$\frac{dy}{dt} = -50y$$

where time t is measured in seconds.

(a) What is the Jacobian J for this equation?

(b) Using the Explicit Euler Method, what is the amplification factor in terms of the step size h?

(c) To maintain numerical stability, the magnitude of the amplification factor must be strictly less than 1. What is the maximum allowable step size h you can use before the numerical solution blows up?

Problem 2: Executing a 2-Stage Runge-Kutta Step

Performing these methods by hand helps you understand how explicit solvers "look ahead" to guess the next slope. Consider the ODE:

$$\frac{dy}{dt} = y + t$$

with the initial condition $y(0) = 1$. Use the explicit 2-stage Runge-Kutta method with a step size of $h = 0.1$ to calculate the numerical approximation for $y(0.1)$.

Recall the 2-Stage Explicit Rule:

$$\begin{aligned} k_1 &= hf(t_0, y_0) \\ k_2 &= hf(t_1, y_0 + k_1) \\ y_1 &= y_0 + \frac{1}{2}(k_1 + k_2) \end{aligned}$$

Problem 3: AI and Solver Selection

You are modeling a chemical reactor where Species A reacts to form Species B almost instantaneously (a timescale of 10^{-5} seconds), but Species B decays into Species C very slowly (a timescale of 10^3 seconds). You prompt an AI to write a MATLAB script to solve this system, and it hands you back a beautiful script utilizing the `ode45` solver.

(a) What is the mathematical term for a system of equations with such vastly different timescales?

(b) What will likely happen when you try to run the AI's script? Explain why based on the mechanics of explicit methods.

(c) What MATLAB solver should you explicitly instruct the AI to use instead?

Solutions to Practice Problems

Solution 1: Stability of the Explicit Euler Method

(a) We are given $f(t, y) = -50y$. The Jacobian is the derivative of the right-hand side with respect to the dependent variable y:

$$J = \frac{\partial f}{\partial y} = -50$$

(b) As derived from the Taylor Series expansion, the amplification factor for the Explicit Euler Method is:

$$\text{Amplification Factor} = 1 + hJ = 1 - 50h$$

(c) For numerical stability, the error must decay (or at least not grow) with each step, meaning $|1 + hJ| < 1$.

$$-1 < 1 - 50h < 1$$
$$-2 < -50h < 0$$

Dividing entirely by -50 (and flipping the inequality signs) yields:

$$\frac{2}{50} > h > 0 \quad \Longrightarrow \quad h < 0.04$$

The maximum allowable step size is $h = 0.04$ **seconds**. If you try to integrate with a step size of 0.05, the solution will oscillate and diverge to infinity!

Solution 2: Executing a 2-Stage Runge-Kutta Step

We are given $f(t, y) = y + t$. Our initial state is $t_0 = 0$ and $y_0 = 1$. The step size is $h = 0.1$, which means our next time step is $t_1 = 0.1$.

First, calculate k_1 (which is just the standard Euler Method step):

$$k_1 = hf(t_0, y_0) = 0.1(1 + 0) = 0.1$$

Next, calculate k_2. This uses the slope at the *next* time step ($t_1 = 0.1$), evaluated at our "Euler guess" for the next y-value ($y_0 + k_1 = 1 + 0.1 = 1.1$):

$$k_2 = hf(t_1, y_0 + k_1) = 0.1 \cdot f(0.1, 1.1) = 0.1(1.1 + 0.1) = 0.1(1.2) = 0.12$$

Finally, average the two k values to find the final 2-Stage approximation for y_1:

$$y_1 = y_0 + \frac{1}{2}(k_1 + k_2)$$
$$y_1 = 1 + \frac{1}{2}(0.1 + 0.12) = 1 + \frac{1}{2}(0.22)$$
$$y_1 = 1.11$$

Note: The exact analytical solution evaluated at $t = 0.1$ *is roughly* 1.11034*. The explicit 2-stage Runge-Kutta method got us extremely close in a single step!*

Solution 3: AI and Solver Selection

(a) A system featuring processes that operate on vastly different timescales (and thus has large, negative Jacobians) is known as a **stiff** equation.

(b) Because the AI defaulted to `ode45` (an explicit method), the adaptive solver will be forced to shrink its step size h to an incredibly small value (on the order of 10^{-5}) just to maintain numerical stability for the fast reaction. Because the overall simulation needs to run for thousands of seconds to capture the slow decay, the script will execute hundreds of millions of the tiny steps required for stability and **run infinitely slowly** (or crash entirely if the required h violates the solver's minimum tolerance).

(c) You must override the software's default choice and instruct it to use an implicit solver designed for stiff equations, such as **`ode15s`** or **`ode23s`**. Unlike explicit methods, these solvers utilize implicit formulations where the step size h and the Jacobian J appear in the denominator of the amplification factor (such as $\frac{1}{1-hJ}$ for the Backward Euler method). This mathematical structure ensures they remain stable along the negative real axis regardless of how large the step size h becomes, allowing the solver to rapidly take giant time steps through the slow, highly-damped portions of the simulation.

11 Numerical Example: The Forced Oscillator

In this chapter, we illustrate the numerical solution techniques of the Euler Method, the 2-stage Runge-Kutta method, the 4-stage R-K method, and an adaptive integration method built into MATLAB, `ode23`. We choose as an example an undamped forced oscillator: a problem for which there is a simple analytic solution.

Our problem is:

$$y'' + y = \sin(at) \tag{11.1}$$

with initial conditions:

$$y(0) = 0, \quad y'(0) = 0 \tag{11.2}$$

The behavior of this oscillator depends on a, the dimensionless driving frequency. In the absence of damping the amplitude blows up as a approaches 1, the natural frequency of the oscillator[cite: 8]. We solve this problem as a pair of first-order equations, such that $y(1)$ is y and $y(2)$ is y'. The corresponding derivatives are:

$$\begin{bmatrix} y(1)' \\ y(2)' \end{bmatrix} = \begin{bmatrix} y(2) \\ -y(1) + \sin(at) \end{bmatrix}$$

with initial conditions $y_0 = [0; 0]$.

We have the analytic solution to this equation:

```
a = 0.2;
yexact = @(t) (sin(a*t) - a*sin(t))/(1-a^2);
```

The Euler Method

We plot things up over four periods (e.g., up to 8π). We just define the derivative and the time step. We also specify a, the frequency of the driving term.

```
dt = 0.2;
tall = [0:dt:8*pi]';
y0 = zeros(2,1); % The initial condition
ydot = @(t,y) [y(2); -y(1) + sin(a*t)];

yem = zeros(2,length(tall)); % we keep both y1 and y2.
n = length(tall)-1;          % the number of steps

for i=1:n
    yem(:,i+1) = yem(:,i) + dt*ydot(tall(i),yem(:,i));
end

figure(1)
plot(tall,yexact(tall),'-o',tall,yem(1,:))
xlabel('t')
ylabel('y')
legend(['exact solution'],['Euler Method'])
```

The Two-Stage Runge-Kutta Technique

Very little needs to be added to the Euler method for the 2-stage R-K technique. The error, for this step size, is very small and is graphically indistinguishable from the exact solution.

```
y2s = zeros(2,length(tall));

for i=1:n
    k1 = dt*ydot(tall(i), y2s(:,i));
    k2 = dt*ydot(tall(i+1), y2s(:,i) + k1);
    y2s(:,i+1) = y2s(:,i) + (k1+k2)/2;
end

figure(1)
hold on
plot(tall,y2s(1,:),'r')
hold off
legend(['exact solution'],['Euler Method'],['2 stage R-K'])
```

The Four-Stage Runge-Kutta Technique

This is very similar to the two-stage technique, except there are four intermediate steps. The method is even more accurate.

```
y4s = zeros(2,length(tall));

for i=1:n
    k1 = dt*ydot(tall(i), y4s(:,i));
    k2 = dt*ydot(tall(i) + dt/2, y4s(:,i) + k1/2);
    k3 = dt*ydot(tall(i) + dt/2, y4s(:,i) + k2/2);
    k4 = dt*ydot(tall(i) + dt, y4s(:,i) + k3);
    y4s(:,i+1) = y4s(:,i) + (k1 + 2*k2 + 2*k3 + k4)/6;
end

figure(1)
hold on
plot(tall,y4s(1,:),'g')
hold off
legend(['exact solution'],['Euler Method'],['2 stage R-K'],['4 stage R-K'])
```

Adaptive Integrator

We use the adaptive integration method ode23 supplied with MATLAB. It is also very accurate, and would require fewer steps.

```
[tout, yout] = ode23(ydot, [0 8*pi], y0);

figure(1)
hold on
plot(tout,yout(:,1),'k')
hold off
legend(['exact solution'],['Euler Method'],['2 stage R-K'],['4 stage R-K'],['ode23'])
```

Error Comparison

A better way of looking at the different methods is to determine the maximum error of each. Here, since we have the exact solution, we can simply subtract it off and determine the maximum deviation.

```
errorem = max(abs(yem(1,:) - yexact(tall')))
error2s = max(abs(y2s(1,:) - yexact(tall')))
error4s = max(abs(y4s(1,:) - yexact(tall')))
errorode23 = max(abs(yout(:,1)' - yexact(tout')))
```

This yields the following maximum errors:

- **Euler Method:** 1.9829×10^{0}
- **2-Stage R-K:** 3.3977×10^{-2}
- **4-Stage R-K:** 6.5674×10^{-5}
- **ode23:** 6.3964×10^{-3}

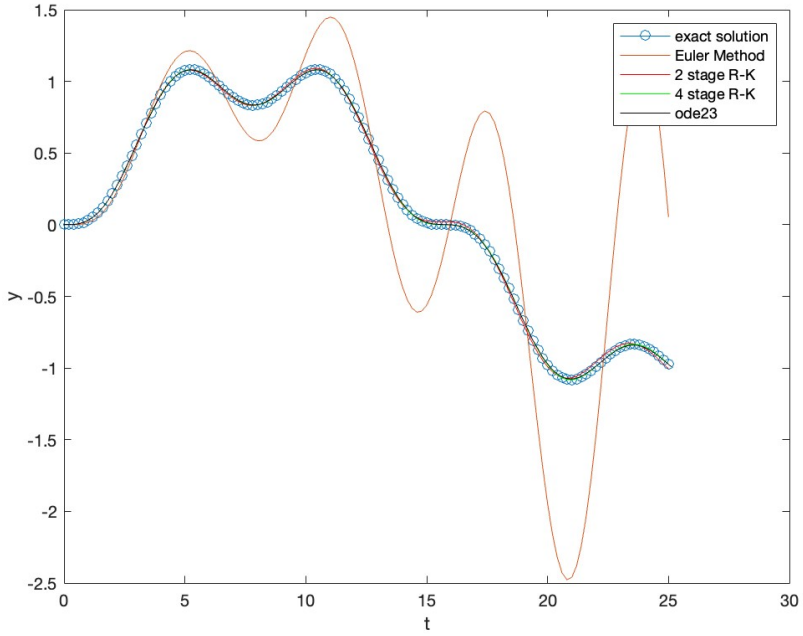


Figure 11.1: Comparison of the exact solution with the Euler, 2-stage RK, 4-stage RK, and ode23 methods.

Conclusion

As can be seen, the Euler Method isn't very accurate for this differential equation. The two RK methods are much more accurate, with the 4-stage method being three orders of magnitude better than the 2-stage method for this choice of `dt`.

The accuracy of the adaptive quadrature routine is determined by its tolerances, and can be adjusted. For the choice of parameters used here, it has fewer steps than the other methods and lies between the 2-stage and 4-stage methods in accuracy.

Other Approaches: Integration to a Condition

Often it is desired to integrate to some physical condition (e.g., where a function value reaches zero, such as a falling droplet hitting the ground) rather than over a fixed, predetermined time range. If you are writing your own Runge-Kutta scheme, this is easily handled by using a `while` loop rather than a `for` loop, updating the index at each step until the condition is met.

At the conclusion of the integration, the exact stopping condition usually falls somewhere between the final two time steps. The last element can be adjusted via interpolation for improved accuracy. For computational efficiency, it is best practice to pre-dimension a large array for the function values, and then simply trim off the unused pre-allocated zeros at the end. (Note that you can also dynamically grow the array size inside the `while` loop step-by-step, though this significantly slows down MATLAB for long integrations). An implementation of a `while` loop approach is given below.

If you are using MATLAB's built-in adaptive integrators like `ode23` or `ode45`, you can accomplish this exact same truncation using an "event function" passed through the `odeset` options command. Documentation for the syntax to set up an event function is provided in the MATLAB help menu—but it is actually much easier just to have your favorite AI write it for you!

```
t0 = 0;           % The initial time
y0 = [0; 0];      % The inital value (column vector)
nchunk = 20;      % We will predimension the arrays and expand as necessary
tallw = zeros(1, nchunk);       % We will keep time as a row vector
y4sw = zeros(length(y0), nchunk); % The y values are an array
tallw(1) = t0;
y4sw(:,1) = y0;
i = 1;

while (y4sw(1,i) > 0) || (i == 1) % The truncation condition, doing it at least once
    if i+1 > length(tallw)        % Redimensioning the array
        y4sw = [y4sw, zeros(length(y0),nchunk)];
        tallw = [tallw, zeros(1, nchunk)];
    end

    k1 = dt*ydot(tallw(i), y4sw(:,i));
    k2 = dt*ydot(tallw(i) + dt/2, y4sw(:,i) + k1/2);
    k3 = dt*ydot(tallw(i) + dt/2, y4sw(:,i) + k2/2);
    k4 = dt*ydot(tallw(i) + dt, y4sw(:,i) + k3);
    y4sw(:,i+1) = y4sw(:,i) + (k1 + 2*k2 + 2*k3 + k4)/6;
    tallw(i+1) = tallw(i) + dt;
    i = i + 1; % We update i
end

% We trim the values
tallw = tallw(1:i);
y4sw = y4sw(:,1:i);

% Now for the interpolation for the root crossing:
f = y4sw(1,end-1) / (y4sw(1,end-1) - y4sw(1,end)); % Test on y=0
tallw(end) = tallw(end-1) + f*dt;
y4sw(:,end) = y4sw(:,end-1) + f*(y4sw(:,end) - y4sw(:,end-1));

figure(2)
plot(tall,yexact(tall),tallw,y4sw(1,:),'*')
xlabel('t')
ylabel('y')
legend(['exact solution'],['4 stage R-K'])
text(5,.5,['Root crossing at t = ',num2str(tallw(end))])
```

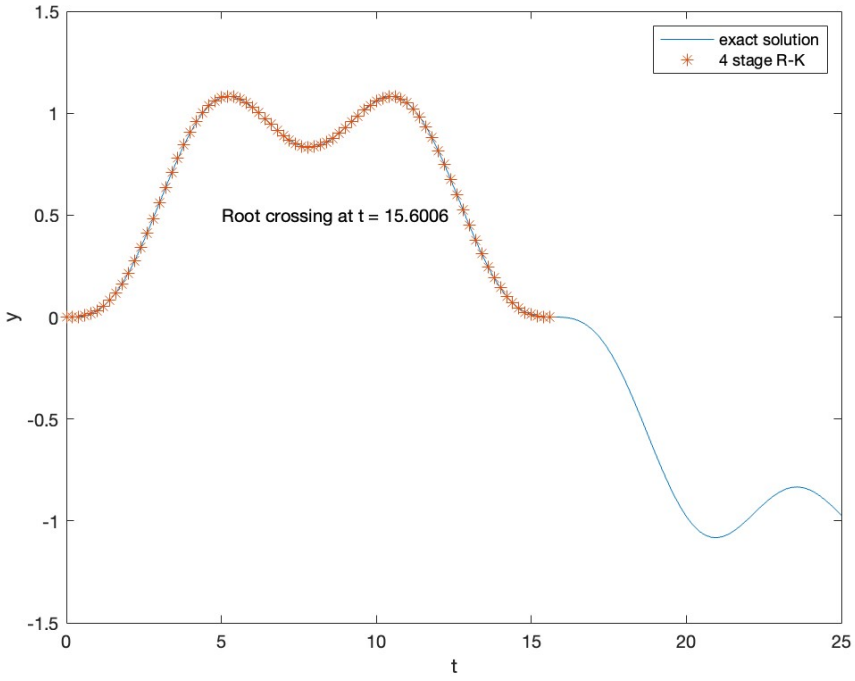


Figure 11.2: Integration tracking the forced oscillator until a root crossing occurs at $t = 15.6006$.

Practice Problems

Problem 1: Stopping at Maximum Amplitude

In the forced oscillator example solved via the 4-stage Runge-Kutta method, the integration was terminated when the position y crossed zero (a root crossing). Modify the `while` loop algorithm to instead terminate the integration when the oscillator reaches its *first maximum positive amplitude* after $t = 0$.

Provide the modified MATLAB `while` loop and the corresponding interpolation logic required to accurately determine the time t_{max} and the amplitude y_{max}. *Hint: Think about what physical quantity must cross zero when the position reaches a maximum.*

Problem 2: The Nonlinear Pendulum

The equations solved in this chapter were linear. However, one of the primary advantages of numerical integration is the ability to solve nonlinear ODEs that have no simple analytic solution. Consider the unforced, nonlinear simple pendulum described by the equation:

$$\theta'' + \frac{g}{L}\sin(\theta) = 0 \tag{11.3}$$

where θ is the angle in radians, $g = 9.81\,\text{m/s}^2$, and $L = 1\,\text{m}$. The pendulum is released from rest at a large initial angle $\theta(0) = \frac{\pi}{2}$, with $\theta'(0) = 0$.

Write the MATLAB code required to define the derivative function `ydot` for this system. Then, using the `ode23` syntax demonstrated in the text, write the code to integrate this system from $t = 0$ to $t = 10$ seconds.

Solutions to Practice Problems

Solution 1: Stopping at Maximum Amplitude

To find the maximum amplitude, we must terminate the integration when the velocity (the derivative of position) crosses zero from positive to negative. In our state vector formulation, $y(1)$ is the position and $y(2)$ is the velocity. Therefore, the `while` loop must check the condition $y(2) > 0$.

```
% Setup is identical to the chapter example
t0 = 0;
y0 = [0; 0];
nchunk = 20;

tallw = zeros(1, nchunk);
y4sw = zeros(length(y0), nchunk);
tallw(1) = t0;
y4sw(:,1) = y0;
i = 1;

% Modified truncation condition: test if velocity y(2) is positive
while (y4sw(2,i) > 0) || (i == 1)
    if i+1 > length(tallw)
        y4sw = [y4sw, zeros(length(y0),nchunk)];
        tallw = [tallw, zeros(1, nchunk)];
    end

    k1 = dt*ydot(tallw(i), y4sw(:,i));
    k2 = dt*ydot(tallw(i) + dt/2, y4sw(:,i) + k1/2);
    k3 = dt*ydot(tallw(i) + dt/2, y4sw(:,i) + k2/2);
    k4 = dt*ydot(tallw(i) + dt, y4sw(:,i) + k3);

    y4sw(:,i+1) = y4sw(:,i) + (k1 + 2*k2 + 2*k3 + k4)/6;
    tallw(i+1) = tallw(i) + dt;

    i = i + 1;
end

tallw = tallw(1:i);
y4sw = y4sw(:,1:i);
% Interpolation for the velocity zero-crossing (y(2) = 0):
f = y4sw(2,end-1) / (y4sw(2,end-1) - y4sw(2,end));
t_max = tallw(end-1) + f*dt;

% Interpolate both position and velocity at t_max
y4sw(:,end) = y4sw(:,end-1) + f*(y4sw(:,end) - y4sw(:,end-1));
y_max = y4sw(1,end)
```

Solution 2: The Nonlinear Pendulum

First, we must reduce the second-order nonlinear ODE into a system of two first-order ODEs. Let $y(1) = \theta$ and $y(2) = \theta'$. The derivatives are therefore $\dot{y}(1) = y(2)$ and $\dot{y}(2) = -\frac{g}{L}\sin(y(1))$.

```
g = 9.81;
L = 1.0;

% Define the nonlinear derivative function
ydot = @(t,y) [y(2); -(g/L)*sin(y(1))];

% Initial conditions: theta = pi/2, omega = 0
y0 = [pi/2; 0];

% Integrate using ode23 from t=0 to t=10
[tout, yout] = ode23(ydot, [0 10], y0);

% Optional: Plot the angular position over time
figure(3)
plot(tout, yout(:,1), '-b')
xlabel('Time (s)')
ylabel('\theta (radians)')
title('Nonlinear Pendulum Motion')
```

Part III

The Language of Multivariable Calculus

12 Div, Grad & Curl

So far we've reviewed ODEs—functions of a single independent variable. Transport (and most of engineering) requires more than one! In addition, momentum is a vector, so there are multiple dependent variables too!

The vector differential operator in Cartesian coordinates is:

$$\nabla = \left(\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z} \right)$$

So the gradient of a scalar ϕ is:

$$\nabla \phi = \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z} \right)$$

This is a vector proportional to the local slope and points uphill. This is the **gradient**.

If $\mathbf{u}$ is a vector, $\mathbf{u} = (u_x, u_y, u_z)$ (e.g., the velocity vector), then:

$$\nabla \cdot \mathbf{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} + \frac{\partial u_z}{\partial z}$$

which is a scalar (the **divergence** of $\mathbf{u}$). This is identically zero for a constant density incompressible fluid, for example.

This is *very* different from $\mathbf{u} \cdot \nabla$:

$$\mathbf{u} \cdot \nabla = u_x \frac{\partial}{\partial x} + u_y \frac{\partial}{\partial y} + u_z \frac{\partial}{\partial z}$$

which is a **scalar operator**. This operator appears in the convective term (usually the left side!) of the heat transfer equation, for example.

There is also $\nabla \mathbf{u}$ (the vector composition or dyadic product), which is a matrix (the Jacobian):

$$\nabla \mathbf{u} = \begin{pmatrix} \frac{\partial u_x}{\partial x} & \frac{\partial u_y}{\partial x} & \frac{\partial u_z}{\partial x} \\ \frac{\partial u_x}{\partial y} & \frac{\partial u_y}{\partial y} & \frac{\partial u_z}{\partial y} \\ \frac{\partial u_x}{\partial z} & \frac{\partial u_y}{\partial z} & \frac{\partial u_z}{\partial z} \end{pmatrix}$$

(Note: We are using the standard transport phenomena convention from Bird, Stewart, and Lightfoot. In some mathematics textbooks, the Jacobian is defined as the transpose of this matrix!)

All three are very important in transport, so you have to pay attention!

The Laplacian

You can take the divergence of the gradient:

$$\nabla \cdot \nabla\phi = \nabla \cdot \left(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial y}, \frac{\partial \phi}{\partial z} \right) = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + \frac{\partial^2 \phi}{\partial z^2} \equiv \nabla^2 \phi$$

This is the **Laplacian** of ϕ, which is the 3-D analog of the second derivative.

In heat transfer in solids:

$$\frac{\partial T}{\partial t} = \alpha \nabla^2 T$$

where α is the thermal diffusivity. So if $\nabla^2 T > 0$ ("concave up"), the local temperature increases in time—which makes sense!

These operators can also apply to vectors (or tensors). In momentum transfer, the velocity vector $\mathbf{u}$ is governed by the Navier-Stokes equations:

$$\underbrace{\rho \left(\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} \right)}_{\text{Density} \times \text{Acceleration \& Convection}} = \underbrace{-\nabla p}_{\text{Pressure Gradient}} + \underbrace{\mu \nabla^2 \mathbf{u}}_{\text{Viscosity}} + \underbrace{\rho \mathbf{g}}_{\text{Gravity}}$$

This is actually three equations—one for each u_x, u_y, u_z. That is because it is a vector equation.

The Curl and Vorticity

There is another way ∇ is used: the **curl** of a vector field. The curl of a magnetic field $\mathbf{B}$ is the current density (from Electricity and Magnetism - E&M):

$$\nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{J} \quad \text{(current density)}$$

In fluid mechanics, the **vorticity** $\boldsymbol{\omega}$ is the curl of the velocity:

$$\boldsymbol{\omega} = \nabla \times \mathbf{u}$$

The vorticity is useful in a number of ways, but curls are a bit messy! It is computed using the determinant:

$$\nabla \times \mathbf{u} = \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ u_x & u_y & u_z \end{vmatrix}$$

$$\nabla \times \mathbf{u} = \left(\frac{\partial u_z}{\partial y} - \frac{\partial u_y}{\partial z}, \quad \frac{\partial u_x}{\partial z} - \frac{\partial u_z}{\partial x}, \quad \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y} \right)$$

You can show that the vorticity is twice the local rate of rotation of a fluid (e.g., twice the angular velocity of a sphere suspended in a flow). Manipulation of the vorticity and curl is much easier using index notation, which we introduce elsewhere.

Curvilinear Coordinates: Cylindrical and Spherical

In transport phenomena, we rarely work strictly in Cartesian boxes. We are usually modeling flow through circular pipes, heat conduction out of cylindrical wires, or mass diffusion from spherical droplets.

Because the basis vectors ($\hat{e}_r, \hat{e}_\theta, \hat{e}_\phi$) in spherical or cylindrical coordinates actually change direction depending on where you are in space, the ∇ operator becomes more complex. We will skip the tedious geometric derivations, but it is critical that you know how to apply the gradient, divergence, and Laplacian in these coordinate systems.

Cylindrical Coordinates (r, θ, z)

Used for pipes, cylinders, and disks.

- **Gradient ($\nabla\phi$):**

$$\left(\frac{\partial \phi}{\partial r}, \quad \frac{1}{r}\frac{\partial \phi}{\partial \theta}, \quad \frac{\partial \phi}{\partial z}\right)$$

  These components are in the r, θ, and z directions.

- **Divergence ($\nabla \cdot \mathbf{u}$):**

$$\frac{1}{r}\frac{\partial}{\partial r}(r u_r) + \frac{1}{r}\frac{\partial u_\theta}{\partial \theta} + \frac{\partial u_z}{\partial z}$$

- **Laplacian ($\nabla^2\phi$):**

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial \phi}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 \phi}{\partial \theta^2} + \frac{\partial^2 \phi}{\partial z^2}$$

Spherical Coordinates (r, θ, ϕ)

Used for bubbles, droplets, and spherical catalyst pellets. *Note: In physics and engineering math, θ is typically the polar angle down from the z-axis, and ϕ is the azimuthal angle in the x-y plane. Think of these as related to latitude and longitude on a globe - although from the pole, not the equator.*

- **Gradient ($\nabla\Phi$):**

$$\left(\frac{\partial \Phi}{\partial r}, \quad \frac{1}{r}\frac{\partial \Phi}{\partial \theta}, \quad \frac{1}{r\sin\theta}\frac{\partial \Phi}{\partial \phi}\right)$$

  These components are in the r, θ, and ϕ directions.

- **Divergence ($\nabla \cdot \mathbf{u}$):**

$$\frac{1}{r^2}\frac{\partial}{\partial r}(r^2 u_r) + \frac{1}{r\sin\theta}\frac{\partial}{\partial \theta}(u_\theta \sin\theta) + \frac{1}{r\sin\theta}\frac{\partial u_\phi}{\partial \phi}$$

- **Laplacian ($\nabla^2\Phi$):**

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial \Phi}{\partial r}\right) + \frac{1}{r^2\sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta\frac{\partial \Phi}{\partial \theta}\right) + \frac{1}{r^2\sin^2\theta}\frac{\partial^2 \Phi}{\partial \phi^2}$$

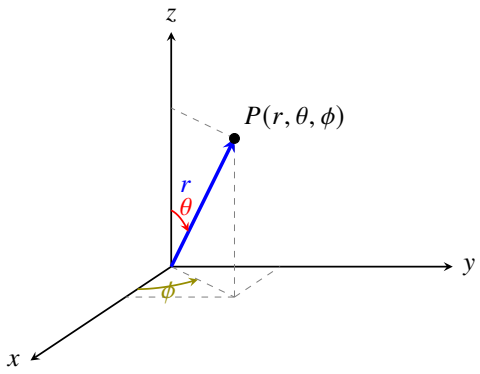


Figure 12.1: The spherical coordinate system configured to match standard physics and transport phenomena conventions. The coordinate r denotes the direct distance from the origin to point P. The polar angle (colatitude) θ measures the angular deflection down from the positive z-axis, while the azimuthal angle ϕ tracks the rotation in the x-y plane mapped counterclockwise from the positive x-axis.

These are the simplest of the expressions in curvilinear coordinates: If you are writing out the Navier-Stokes equations, for example, you require the $(\mathbf{u} \cdot \nabla)\mathbf{u}$ terms, which are much more complex (as are the $\nabla^2\mathbf{u}$ terms). In this case it is **always** safest to look them up!

The Divergence Theorem

The last bit we'll talk about here is the fundamental theorem in multi-variable calculus: **The Divergence Theorem**.

For an arbitrary closed surface ∂D bounding domain D, outward pointing unit normal $\mathbf{n}$, and vector field $\mathbf{u}$ differentiable on D we have:

$$\underbrace{\int_{\partial D} \mathbf{u} \cdot \mathbf{n}\, dA}_{\text{Surface integral}} = \underbrace{\int_{D} \nabla \cdot \mathbf{u}\, dV}_{\text{Volume integral}}$$

This is used extensively in deriving transport equations!

This works for scalars too:

$$\underbrace{\int_{\partial D} \phi\, \mathbf{n}\, dA}_{\text{Surface integral}} = \underbrace{\int_{D} \nabla \phi\, dV}_{\text{Volume integral}}$$

which can be easily proved by substituting $\mathbf{u} = \mathbf{e}\,\phi$ into the Divergence Theorem.

Practice Problems

Test your skills on these problems. In transport phenomena, the most common mathematical mistake is confusing the divergence, gradient, and Laplacian operators, or applying Cartesian rules to cylindrical geometries!

Problem 1: Gradient and Laplacian

Consider a 3-D steady-state temperature field described by the scalar function $T(x, y, z) = x^2y + z^3$.

(a) Calculate the temperature gradient, ∇T.

(b) Calculate the Laplacian of the temperature field, $\nabla^2 T$.

Problem 2: The Convective Operator

In the Navier-Stokes and thermal energy equations, the term $\mathbf{u} \cdot \nabla$ appears frequently. Suppose the fluid velocity is $\mathbf{u} = (2y, x, 0)$ and the concentration of a chemical species is $C(x, y, z) = 3x + 4y$. Evaluate the convective transport term, $\mathbf{u} \cdot \nabla C$.

Problem 3: Curl and Vorticity

Consider a simple 2-D shear flow between two plates, where the velocity field is given by $\mathbf{u} = (ky, 0, 0)$ and k is a constant shear rate. Calculate the vorticity $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. What direction does the vorticity vector point?

Problem 4: Curvilinear Coordinates

The steady-state temperature profile inside a solid heat-generating cylindrical fuel rod of radius R is given by $T(r) = T_0 \left(1 - \frac{r^2}{R^2}\right)$, where T_0 is the centerline temperature. Using the cylindrical coordinate operators from the notes, calculate the Laplacian $\nabla^2 T$. *(Hint: Since the temperature only depends on r, the θ and z derivatives are zero).*

Problem 5: The Divergence Theorem

A compressible gas is expanding radially outward such that the divergence of its velocity field is constant everywhere in space: $\nabla \cdot \mathbf{u} = 5\ \text{s}^{-1}$. Consider a spherical control volume D with a total volume of $10\ \text{m}^3$. Use the Divergence Theorem to calculate the total volumetric flow rate leaving the surface of this sphere, $\int_{\partial D} \mathbf{u} \cdot \mathbf{n}\, dA$. This is the integral of the normal component of the flux over the surface area.

Solutions to Practice Problems

Solution 1: Gradient and Laplacian

(a) The gradient is a vector operator. We take the partial derivative of T with respect to each coordinate direction:

$$\nabla T = \left(\frac{\partial T}{\partial x}, \frac{\partial T}{\partial y}, \frac{\partial T}{\partial z}\right) = \left(2xy, \quad x^2, \quad 3z^2\right)$$

(b) The Laplacian is the divergence of the gradient ($\nabla \cdot \nabla T$). We take the dot product of the ∇ operator with our vector from part (a):

$$\nabla^2 T = \frac{\partial}{\partial x}(2xy) + \frac{\partial}{\partial y}(x^2) + \frac{\partial}{\partial z}(3z^2) = 2y + 0 + 6z = 2y + 6z$$

Solution 2: The Convective Operator

First, we construct the scalar operator $\mathbf{u} \cdot \nabla$:

$$\mathbf{u} \cdot \nabla = u_x \frac{\partial}{\partial x} + u_y \frac{\partial}{\partial y} + u_z \frac{\partial}{\partial z} = 2y \frac{\partial}{\partial x} + x \frac{\partial}{\partial y} + 0 \frac{\partial}{\partial z}$$

Now, we apply this operator to the scalar concentration field C:

$$(\mathbf{u} \cdot \nabla)C = 2y\left(\frac{\partial C}{\partial x}\right) + x\left(\frac{\partial C}{\partial y}\right) = 2y(3) + x(4) = 6y + 4x$$

Solution 3: Curl and Vorticity

We evaluate the curl using the determinant formulation. We are given $u_x = ky$, $u_y = 0$, and $u_z = 0$.

$$\begin{aligned}
\nabla \times \mathbf{u} &= \begin{vmatrix} \hat{i} & \hat{j} & \hat{k} \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ ky & 0 & 0 \end{vmatrix} \\
&= \hat{i}\left(\frac{\partial(0)}{\partial y} - \frac{\partial(0)}{\partial z}\right) - \hat{j}\left(\frac{\partial(0)}{\partial x} - \frac{\partial(ky)}{\partial z}\right) + \hat{k}\left(\frac{\partial(0)}{\partial x} - \frac{\partial(ky)}{\partial y}\right) \\
&= \hat{i}(0) - \hat{j}(0) + \hat{k}(0 - k) = (0, 0, -k)
\end{aligned}$$

The vorticity vector points entirely in the $-z$ **direction** (into the page). This makes physical sense: a fluid moving rightward with speed increasing as you go "up" the y-axis induces a clockwise (negative) rotation!

Solution 4: Curvilinear Coordinates

The radial portion of the Laplacian in cylindrical coordinates is:

$$\nabla^2 T = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right)$$

Step 1: Find the first derivative $\frac{\partial T}{\partial r}$:

$$\frac{\partial T}{\partial r} = \frac{\partial}{\partial r}\left(T_0 - T_0\frac{r^2}{R^2}\right) = -2T_0\frac{r}{R^2}$$

Step 2: Multiply by r (the physical weight function):

$$r\frac{\partial T}{\partial r} = -2T_0\frac{r^2}{R^2}$$

Step 3: Take the derivative of that product with respect to r:

$$\frac{\partial}{\partial r}\left(-2T_0\frac{r^2}{R^2}\right) = -4T_0\frac{r}{R^2}$$

Step 4: Finally, divide by the r on the outside:

$$\nabla^2 T = \frac{1}{r}\left(-4T_0\frac{r}{R^2}\right) = -\frac{4T_0}{R^2}$$

Notice how if you had incorrectly used the Cartesian 1-D Laplacian, d^2T/dr^2, you would have gotten $-2T_0/R^2$, missing a factor of 2. Geometry matters!

Solution 5: The Divergence Theorem

The Divergence Theorem relates the surface flux to the volume integral of the divergence:

$$\int_{\partial D} \mathbf{u}\cdot\mathbf{n}\,dA = \int_D \nabla\cdot\mathbf{u}\,dV$$

We are given that $\nabla\cdot\mathbf{u} = 5$ everywhere inside the domain D. Therefore:

$$\int_{\partial D} \mathbf{u}\cdot\mathbf{n}\,dA = \int_D (5)\,dV$$

Because 5 is a constant, we can pull it out of the integral:

$$= 5\int_D dV$$

The term $\int_D dV$ is simply the total volume of the spherical domain, which is given as 10 m^3.

$$\int_{\partial D} \mathbf{u}\cdot\mathbf{n}\,dA = 5\times 10 = 50\ \mathrm{m}^3/\mathrm{s}$$

13 Index Notation

What is Index Notation?

Index notation is simply a compact and convenient way of representing scalars, vectors, and tensors. It is particularly useful for fluid mechanics, especially (as we shall see) at low Reynolds numbers. There is no new physics associated with index notation; however, it can reveal non-obvious symmetries and relations which were already there! It is not particularly difficult to use, but it is quite different in approach from the standard notation you are more familiar with. The tremendous simplification of equations resulting from its use, however, make the effort to understand it worthwhile!

The Starting Point

For any tensor, the **order** of the tensor is given by the **number of unrepeated indices**!

- a: no indices, scalar (0th order) (example: mass)
- x_i, u_j: one index, both are vectors (1st order) (example: velocity)
- A_{ij}: two indices, 2nd order tensor (matrix) (example: stress tensor)
- ϵ_{ijk}: three indices, 3rd order tensor (example: Levi-Civita symbol used in cross products)

The letters used as subscripts **don't matter**; e.g., x_i, x_j, x_p, etc., are perfectly equivalent.

Key Exception: In an equation, each term must have the same unrepeated indices. For example:

- $x_i = y_i$ is valid (it is the exact same as writing $\mathbf{x} = \mathbf{y}$).
- $x_i = y_j$ is an **error**!

The Dot Product and Summation

The most basic operation using index notation is the dot product. **Repeated indices (in a product) imply summation!** Thus:

$$x_i y_i \equiv \mathbf{x} \cdot \mathbf{y} = \sum_i x_i y_i$$

(e.g., $x_i y_i = x_1 y_1 + x_2 y_2 + x_3 y_3$). Just think of how you would code it up on a computer using loops!

This applies to a matrix too! For example, A_{ii} has **no unrepeated indices**, and thus is just the scalar trace of A_{ij}.

The vector composition or outer product is also simple. $\mathbf{A} = \mathbf{x}\mathbf{y}^T$ is given by $A_{ij} = x_i y_j$. Since there are two unrepeated indices, $x_i y_j$ yields a 2nd order tensor!

Important Rule: You **cannot** repeat an index in any single product term more than once.

- $x_i y_j z_i \equiv y_j(\mathbf{x} \cdot \mathbf{z})$ (OK)
- $x_i y_i z_i \equiv$ **ERROR!**

The order of multiplication (dot product) is preserved by the names/order of the indices. Remember the linear algebra equation $\mathbf{A}\mathbf{x} = \mathbf{b}$? In index notation:

$$A_{ij} x_j = b_i$$

To take the transpose, just reverse the order of the indices: $(\mathbf{A})^T = A_{ji}$.

Any repeated index may be renamed freely:

$$A_{ij} x_j = A_{ik} x_k = A_{ip} x_p$$

Free indices may never be renamed independently.

Example: The Normal Equations

In linear regression you are trying to fit a data vector of observations $\mathbf{b}$ with modeling functions evaluated at the various times (or positions) of the measurements $\mathbf{A}$ weighted by a vector of unknown parameters $\mathbf{x}$. Minimizing the 2-norm of the deviation between the model and the data (the residual) results in the Normal Equations for linear regression $\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$.

In index notation we would write this as:

$$A_{ki} A_{kj} x_j = A_{ki} b_k$$

We could also look at the residual from linear regression: $r_i = A_{ij} x_j - b_i$. The sum of the squared residuals is $\mathbf{r}^T\mathbf{r}$:

$$\mathbf{r}^T\mathbf{r} \equiv (A_{ij} x_j - b_i)(A_{ik} x_k - b_i)$$

Note that there are no unrepeated indices in the product, so the final result is a scalar. Also note that we switched a pair of 'j's to 'k's in the second bracket to avoid repeating 'j' too many times! The index 'j' was repeated already, so we can't use it again in a product. The repeated letters are used up - they can be any letter you like, but you can't multiply by a term with the same letter as an index again, as that would cause the ambiguity which the rules are designed to avoid!

Derivatives and the Kronecker Delta

We define a couple of things:

1. **Gradient Operator:** $\nabla = \dfrac{\partial}{\partial x_i}$ (or $\dfrac{\partial}{\partial x_j}$, $\dfrac{\partial}{\partial x_k}$, etc.)

2. **Identity Matrix (Kronecker Delta): I** $= \delta_{ij}$

$$\delta_{ij} = \begin{cases} 0 & i \neq j \\ 1 & i = j \end{cases}$$

Note that taking the derivative of a spatial coordinate with respect to another yields the identity matrix: $\frac{\partial x_i}{\partial x_j} = \delta_{ij}$. Also note that $\frac{\partial x_i}{\partial x_i} = \delta_{ii} = 1 + 1 + 1 = 3$.

Taking the dot product of a matrix (or vector) with the identity matrix leaves it unchanged. In index notation, this acts as an index replacement tool:

$$A_{ij}\delta_{jk} = A_{ik}$$

(You just replace the 'j' with a 'k').

Solving the Normal Equations via Index Notation

Let's use this to solve for the Normal Equations! Recall we want to find the minimum of the residual squared, so we set $\nabla(\mathbf{r}^T\mathbf{r}) = 0$.

In index notation:

$$\frac{\partial}{\partial x_i}\left\{(A_{jk}x_k - b_j)(A_{jl}x_l - b_j)\right\} = 0$$

Expanding this out:

$$= \frac{\partial}{\partial x_i}\left\{A_{jk}x_kA_{jl}x_l - A_{jk}x_kb_j - b_jA_{jl}x_l + b_jb_j\right\}$$

Since we only have to preserve the order of the indices, and noting that $A_{jk}x_k \equiv A_{jl}x_l$, we can combine the two middle terms:

$$= \frac{\partial}{\partial x_i}\left\{A_{jk}A_{jl}x_kx_l - 2A_{jk}b_jx_k + b_jb_j\right\}$$

(The use of k or l in the middle term was indeterminate because they were repeated. Only the unrepeated index must be the same on both sides!)

Now we take the derivatives. Note that A_{jk} and b_j are constants (not dependent on x_i), so they pop out of the derivative!

$$\nabla(\mathbf{r}^T\mathbf{r}) \equiv A_{jk}A_{jl}\frac{\partial}{\partial x_i}(x_kx_l) - 2A_{jk}b_j\frac{\partial x_k}{\partial x_i} + \frac{\partial(b_jb_j)}{\partial x_i}$$

The last term is zero because b_j is constant. The derivative in the second term is simply $\frac{\partial x_k}{\partial x_i} = \delta_{ki}$.

Now we compute the first term using the product rule:

$$\frac{\partial}{\partial x_i}(x_kx_l) = x_k\frac{\partial x_l}{\partial x_i} + x_l\frac{\partial x_k}{\partial x_i} = x_k\delta_{il} + x_l\delta_{ik}$$

Putting it all together:

$$\nabla(\mathbf{r}^T\mathbf{r}) \equiv A_{jk}A_{jl}(x_k\delta_{il} + x_l\delta_{ik}) - 2A_{jk}b_j\delta_{ik}$$

Applying the Kronecker delta (which replaces the indices):

$$\nabla(\mathbf{r}^T\mathbf{r}) \equiv A_{jk}A_{ji}x_k + A_{ji}A_{jl}x_l - 2A_{ji}b_j$$

Now, the first two terms are identical since in both cases 'j' and 'k' (or 'l') are repeated indices and thus indeterminate. Setting $\nabla(\mathbf{r}^T\mathbf{r}) = 0$ becomes:

$$\begin{aligned} 2A_{ji}A_{jk}x_k - 2A_{ji}b_j &= 0 \\ A_{ji}A_{jk}x_k &= A_{ji}b_j \end{aligned}$$

Which is exactly the same as:

$$\mathbf{A}^T\mathbf{A}\mathbf{x} = \mathbf{A}^T\mathbf{b}$$

The Alternating Tensor and Cross Products

In addition to the Kronecker delta δ_{ij}, there is another special quantity we will use: the **Alternating Tensor** ϵ_{ijk}. This is also known as the Levi-Civita symbol or permutation symbol. It is defined as:

$$\epsilon_{ijk} = \begin{cases} 1 & \text{if } i,j,k \text{ are cyclic (123, 231, 312)} \\ -1 & \text{if } i,j,k \text{ are counter-cyclic (321, 213, 132)} \\ 0 & \text{if any two indices are the same (e.g., } i = j) \end{cases}$$

We use this tensor to compute the **cross product**. The cross product $\mathbf{C} = \mathbf{A} \times \mathbf{B}$ is written in index notation as:

$$C_i = \epsilon_{ijk}A_jB_k$$

Note that in a cross product between two vectors the index of the first vector (in this case A_j) corresponds to the second index of ϵ_{ijk}, the index of the second vector (in this case B_k) corresponds to the last index of ϵ_{ijk}, and the free index result (e.g., C_i) corresponds to the first index of ϵ_{ijk}. Drawing arrows when converting an expression involving cross products into index notation is often helpful until you get good at it (and sometimes even then for complex expressions).

Properties of the Alternating Tensor and Symmetric Matrices

Note that just as $\mathbf{A} \times \mathbf{B} = -\mathbf{B} \times \mathbf{A}$, in index notation we have:

$$\epsilon_{ijk} = -\epsilon_{jik}$$

Switching the order of any pair of indices throws in a negative sign! If ϵ_{ijk} is cyclic, ϵ_{jik} must be counter-cyclic and vice versa.

Technically, any matrix for which $A_{ij} = A_{ji}$ is termed **symmetric**. A matrix for which $B_{ij} = -B_{ji}$ is **anti-symmetric**.

Note: The double dot product (e.g., $A_{ij}B_{ij}$ – no unrepeated indices) of a symmetric and an anti-symmetric matrix is **zero**!

$$\begin{aligned} A_{ij}B_{ij} &= A_{ji}B_{ij} \quad (\text{if } A^T = A) \\ &= -A_{ji}B_{ji} \quad (\text{if } B^T = -B) \\ &\equiv -A_{ij}B_{ij} \quad (\text{relabeling repeated indices}) \end{aligned}$$

Thus, since $A_{ij}B_{ij} = -A_{ij}B_{ij}$, both are zero!

We can use this to prove that $\nabla \times (\nabla\phi) = 0$:

$$\epsilon_{ijk}\frac{\partial}{\partial x_j}\left(\frac{\partial \phi}{\partial x_k}\right) = \underbrace{\epsilon_{ijk}}_{\text{anti-symmetric}} \underbrace{\frac{\partial^2}{\partial x_j \partial x_k}(\phi)}_{\text{symmetric}} = 0$$

This is actually a very important relationship in fluid mechanics: it says that the velocity field in potential flow (where the velocity is proportional to the gradient of a scalar potential field) is irrotational (the vorticity, or curl of the velocity is zero). The flip side of this is that if the vorticity is non-zero, you can't have potential flow.

Isotropy and General Tensors

Another useful concept is **isotropy**. Mathematically, a tensor is isotropic if it is invariant under rotation of the coordinate system. Physically, it is isotropic if it looks the same from all directions!

- A sphere is isotropic, a football isn't!
- All scalars are isotropic.
- No vectors are isotropic (except a vector of length zero)!
- The most general 2nd order isotropic tensor is $\lambda\delta_{ij}$ (where λ is a constant scalar).
- The most general 3rd order isotropic tensor is $\lambda\epsilon_{ijk}$.
- The most general 4th order isotropic tensor is:

$$A_{ijkl} = \lambda_1\delta_{ij}\delta_{kl} + \lambda_2\delta_{ik}\delta_{jl} + \lambda_3\delta_{il}\delta_{jk}$$

Where $\lambda_1, \lambda_2, \lambda_3$ are scalars.

Proving Vector Calculus Identities

We can use this to prove vector calculus identities. From texts, we have:

$$\nabla \times (\nabla \times \mathbf{u}) = \nabla(\nabla \cdot \mathbf{u}) - \nabla^2\mathbf{u}$$

Let's prove this!

$$\nabla \times (\nabla \times \mathbf{u}) \equiv \epsilon_{ijk}\frac{\partial}{\partial x_j}\left(\epsilon_{klm}\frac{\partial u_m}{\partial x_l}\right)$$

Note the order of the indices. This is important when working with ϵ_{ijk}. Try drawing some arrows with this one!

Thus:

$$\nabla \times (\nabla \times \mathbf{u}) \equiv \epsilon_{ijk}\epsilon_{klm}\frac{\partial^2 u_m}{\partial x_j \partial x_l}$$

What's $\epsilon_{ijk}\epsilon_{klm}$? The index k is repeated (meaning it is summed over), but there are 4 unrepeated free indices (i, j, l, m), so the result is a 4th-order tensor. Because ϵ_{ijk} is isotropic, the product is also isotropic. In fact, the product of *any* isotropic tensors is also isotropic, which is *very* useful! Hence:

$$\epsilon_{ijk}\epsilon_{klm} = \lambda_1\delta_{ij}\delta_{lm} + \lambda_2\delta_{il}\delta_{jm} + \lambda_3\delta_{im}\delta_{jl}$$

where $\lambda_1, \lambda_2, \lambda_3$ are to be determined. We can calculate these by multiplying both sides by each of the three terms on the right hand side (one at a time!) which then yields three equations for the three λ's.

1. Multiply by $\delta_{ij}\delta_{lm}$:

$$\text{LHS: } \epsilon_{ijk}\epsilon_{klm}\delta_{ij}\delta_{lm} \equiv \epsilon_{iik}\epsilon_{kll} = 0$$

This is zero because ϵ_{iik} is zero for all i, k. Also, $\epsilon_{ijk}\delta_{ij} = 0$ because ϵ_{ijk} is anti-symmetric and δ_{ij} is symmetric, so the double-dot product of the two is likewise zero!

Now for the RHS:

$$\begin{aligned}(\lambda_1\delta_{ij}\delta_{lm} + \lambda_2\delta_{il}\delta_{jm} + \lambda_3\delta_{im}\delta_{jl}) \times (\delta_{ij}\delta_{lm}) &= \lambda_1\delta_{ii}\delta_{ll} + \lambda_2\delta_{im}\delta_{im} + \lambda_3\delta_{im}\delta_{im} \\ &= \lambda_1(3)(3) + \lambda_2(3) + \lambda_3(3)\end{aligned}$$

since $\delta_{ii} = 1 + 1 + 1 = 3$! We thus get the first equation:

$$0 = 9\lambda_1 + 3\lambda_2 + 3\lambda_3$$

2. Multiply by $\delta_{il}\delta_{jm}$:

$$\text{RHS: } \epsilon_{ijk}\epsilon_{klm}\delta_{il}\delta_{jm} = 3\lambda_1 + 9\lambda_2 + 3\lambda_3$$

Where the RHS was calculated the same way as before. The LHS is:

$$\epsilon_{ijk}\epsilon_{kij}$$

Now if ϵ_{ijk} is cyclic, so is ϵ_{kij}. Likewise, if ϵ_{ijk} is counter-cyclic, so is ϵ_{kij}. Thus, the product is just $(1)(1) = 1$ or $(-1)(-1) = 1$ for all six non-zero elements! This yields:

$$6 = 3\lambda_1 + 9\lambda_2 + 3\lambda_3$$

3. Multiply by $\delta_{im}\delta_{jl}$: Likewise, the multiplication by the last term yields:

$$\text{RHS: } 3\lambda_1 + 3\lambda_2 + 9\lambda_3$$

$$\text{LHS: } \epsilon_{ijk}\epsilon_{klm}\delta_{im}\delta_{jl} = \epsilon_{ijk}\epsilon_{kji} = -6$$

These three equations have the exact solution: $\lambda_1 = 0$, $\lambda_2 = 1$, $\lambda_3 = -1$. Thus, we arrive at the famous epsilon-delta identity:

$$\epsilon_{ijk}\epsilon_{klm} = \delta_{il}\delta_{jm} - \delta_{im}\delta_{jl}$$

And hence returning to our vector identity:

$$\begin{aligned}
\nabla \times (\nabla \times \mathbf{u}) &\equiv \epsilon_{ijk}\epsilon_{klm}\frac{\partial^2 u_m}{\partial x_j \partial x_l} \\
&= (\delta_{il}\delta_{jm} - \delta_{im}\delta_{jl})\frac{\partial^2 u_m}{\partial x_j \partial x_l} \\
&= \delta_{il}\delta_{jm}\frac{\partial^2 u_m}{\partial x_j \partial x_l} - \delta_{im}\delta_{jl}\frac{\partial^2 u_m}{\partial x_j \partial x_l} \\
&= \frac{\partial}{\partial x_i}\left(\frac{\partial u_j}{\partial x_j}\right) - \frac{\partial^2 u_i}{\partial x_j^2} \\
&\equiv \nabla(\nabla \cdot \mathbf{u}) - \nabla^2 \mathbf{u}
\end{aligned}$$

which completes the identity!

Pseudo-Tensors vs. Physical Tensors

The last concept we wish to explore is the difference between pseudo-tensors and physical tensors. This distinction arises from the choice of right-handed or left-handed coordinate systems. A pseudo-tensor is one whose sign depends on this choice; a physical tensor is one which doesn't! The reason why we make this distinction is that a physical tensor and a pseudotensor can never be equal!

Let's look at some examples:

- **Physical:** velocity, force, stress, δ_{ij}
- **Pseudo:** angular velocity, torque, vorticity, ϵ_{ijk}

We go from one to the other via the cross-product!

For instance, vorticity is defined as:

$$\omega_i \equiv \epsilon_{ijk}\frac{\partial u_k}{\partial x_j} \quad \text{(e.g., } \boldsymbol{\omega} = \nabla \times \mathbf{u}\text{)}$$

Here, ω_i is a pseudovector, u_k is a physical vector, and likewise ϵ_{ijk} operates as a pseudotensor connecting them.

Practice Problems

Problem 1: Index Replacement and Contraction

Simplify the following expressions in 3D space using the properties of the Kronecker delta (δ_{ij}):

(a) $\delta_{ij}\delta_{jk}\delta_{ki}$

(b) $\frac{\partial}{\partial x_j}(x_i\delta_{ij})$

Problem 2: Symmetries and Vector Calculus

Use index notation to prove the identity $\nabla \cdot (\nabla \times \mathbf{u}) = 0$ for any smooth vector field $\mathbf{u}$. Explicitly identify the symmetric and anti-symmetric tensors involved in the proof.

Problem 3: The Product Rule and Cross Products

Prove the vector identity $\nabla \cdot (\mathbf{u} \times \mathbf{v}) = \mathbf{v} \cdot (\nabla \times \mathbf{u}) - \mathbf{u} \cdot (\nabla \times \mathbf{v})$ using index notation.

Problem 4: Derivatives of Quadratic Forms

Let $\mathbf{A}$ be a constant, *non-symmetric* second-order tensor, and let $\mathbf{x}$ be the position vector. Find the index notation expression for the gradient of the scalar field $\phi(\mathbf{x}) = x_i A_{ij} x_j$, denoted as $\frac{\partial \phi}{\partial x_k}$. Show how your final expression simplifies if $\mathbf{A}$ is symmetric.

Solutions to Practice Problems

Solution 1: Index Replacement and Contraction

(a) We apply the index-replacement property of the Kronecker delta sequentially from left to right:

$$(\delta_{ij}\delta_{jk})\delta_{ki} = \delta_{ik}\delta_{ki}$$

Applying it a second time replaces the index k with i:

$$\delta_{ik}\delta_{ki} = \delta_{ii}$$

As shown in the notes, the repeated index implies summation over all three spatial dimensions:

$$\delta_{ii} = \delta_{11} + \delta_{22} + \delta_{33} = 1 + 1 + 1 = 3$$

(b) First, evaluate the expression inside the parenthesis. The Kronecker delta δ_{ij} collapses the index i into a j:

$$x_i\delta_{ij} = x_j$$

Now, substitute this back into the derivative:

$$\frac{\partial}{\partial x_j}(x_j) = \delta_{jj} = 3$$

Solution 2: Symmetries and Vector Calculus

We begin by expressing the curl operator in index notation:

$$(\nabla \times \mathbf{u})_i = \epsilon_{ijk}\frac{\partial u_k}{\partial x_j}$$

Taking the divergence of this vector field means applying the gradient operator $\frac{\partial}{\partial x_i}$ with a matching repeated index:

$$\nabla \cdot (\nabla \times \mathbf{u}) \equiv \frac{\partial}{\partial x_i}\left(\epsilon_{ijk}\frac{\partial u_k}{\partial x_j}\right)$$

Since the alternating tensor ϵ_{ijk} consists of constant scalars, it can be factored out of the spatial derivative:

$$= \underbrace{\epsilon_{ijk}}_{\text{anti-symmetric}} \underbrace{\frac{\partial^2 u_k}{\partial x_i \partial x_j}}_{\text{symmetric}}$$

Here, we analyze the two constituent tensors with respect to the indices i and j:

- ϵ_{ijk} is completely **anti-symmetric** under the permutation of i and j ($\epsilon_{ijk} = -\epsilon_{jik}$).

- By Clairaut's theorem for smooth fields, the order of spatial differentiation is interchangeable. Thus, the second derivative operator is **symmetric** under the permutation of i and j ($\frac{\partial^2}{\partial x_i \partial x_j} = \frac{\partial^2}{\partial x_j \partial x_i}$).

As derived in the notes, the double-dot product of any symmetric and anti-symmetric tensor over the same pair of repeated indices is identically zero. Therefore:

$$\nabla \cdot (\nabla \times \mathbf{u}) = 0$$

Solution 3: The Product Rule and Cross Products

First, write the cross product in index notation: $(\mathbf{u} \times \mathbf{v})_i = \epsilon_{ijk} u_j v_k$. The divergence is:

$$\nabla \cdot (\mathbf{u} \times \mathbf{v}) \equiv \frac{\partial}{\partial x_i} \left(\epsilon_{ijk} u_j v_k \right)$$

Pulling out the constant alternating tensor and applying the product rule yields:

$$= \epsilon_{ijk} \left(\frac{\partial u_j}{\partial x_i} v_k + u_j \frac{\partial v_k}{\partial x_i} \right) = \epsilon_{ijk} \frac{\partial u_j}{\partial x_i} v_k + \epsilon_{ijk} u_j \frac{\partial v_k}{\partial x_i}$$

Let's analyze and rearrange each of these two terms independently to recover standard vector operations:

For the **first term**, rearrange the scalars to isolate the derivative: $\left(\epsilon_{ijk} \frac{\partial u_j}{\partial x_i} \right) v_k$. Since cyclic permutations of indices do not change the sign of the alternating tensor ($\epsilon_{ijk} = \epsilon_{kij}$), we can rewrite this as:

$$\left(\epsilon_{kij} \frac{\partial u_j}{\partial x_i} \right) v_k = (\nabla \times \mathbf{u})_k v_k = \mathbf{v} \cdot (\nabla \times \mathbf{u})$$

For the **second term**, shift the vector component u_j to the front: $u_j \left(\epsilon_{ijk} \frac{\partial v_k}{\partial x_i} \right)$. Swapping the first two indices of the alternating tensor flips its sign ($\epsilon_{ijk} = -\epsilon_{jik}$):

$$= -u_j \left(\epsilon_{jik} \frac{\partial v_k}{\partial x_i} \right) = -u_j (\nabla \times \mathbf{v})_j = -\mathbf{u} \cdot (\nabla \times \mathbf{v})$$

Combining both evaluated terms yields the final proof:

$$\nabla \cdot (\mathbf{u} \times \mathbf{v}) = \mathbf{v} \cdot (\nabla \times \mathbf{u}) - \mathbf{u} \cdot (\nabla \times \mathbf{v})$$

Solution 4: Derivatives of Quadratic Forms

We apply the gradient operator with respect to index k:

$$\frac{\partial \phi}{\partial x_k} = \frac{\partial}{\partial x_k} \left(x_i A_{ij} x_j \right)$$

Because $\mathbf{A}$ is constant, A_{ij} moves outside of the derivative:

$$= A_{ij} \frac{\partial}{\partial x_k} (x_i x_j)$$

Applying the product rule to the position vectors:

$$= A_{ij} \left(\frac{\partial x_i}{\partial x_k} x_j + x_i \frac{\partial x_j}{\partial x_k} \right) = A_{ij} \left(\delta_{ik} x_j + x_i \delta_{jk} \right)$$

Distribute A_{ij} across the parenthetical terms:

$$= A_{ij} \delta_{ik} x_j + A_{ij} x_i \delta_{jk}$$

Now use the Kronecker delta to substitute indices:

- In the first term, δ_{ik} replaces index i with k, yielding $A_{kj} x_j$.
- In the second term, δ_{jk} replaces index j with k, yielding $A_{ik} x_i$.

Thus, the most general derivative is:

$$\frac{\partial \phi}{\partial x_k} = A_{kj} x_j + A_{ik} x_i$$

Symmetric Case: If $\mathbf{A}$ is symmetric, then $A_{ik} = A_{ki}$. We can rewrite the second term by renaming its dummy/repeated variable from i to j ($A_{ik} x_i \equiv A_{jk} x_j$). Applying the symmetry condition $A_{jk} = A_{kj}$, we get:

$$\frac{\partial \phi}{\partial x_k} = A_{kj} x_j + A_{kj} x_j = 2 A_{kj} x_j$$

Which is equivalent to the vector form $\nabla \phi = 2\mathbf{A}\mathbf{x}$.

14 Advanced Applications of Index Notation

In the previous chapter, we developed the machinery of index notation, including the Kronecker delta, the alternating tensor, and the rules of symmetry. Here, we will demonstrate the profound power of these tools when applied to physical problems—specifically, the kinematics of fluid motion and the behavior of particles in Stokes flow.

Symmetric and Antisymmetric Decomposition

A fundamental property in tensor mathematics is that *any* second-rank tensor T_{ij} can be uniquely decomposed into the sum of a symmetric tensor S_{ij} and an antisymmetric tensor A_{ij}.

We can write this simply as:

$$T_{ij} = \frac{1}{2}(T_{ij} + T_{ji}) + \frac{1}{2}(T_{ij} - T_{ji}) \tag{14.1}$$

Let us define the first term as $S_{ij} = \frac{1}{2}(T_{ij} + T_{ji})$ and the second term as $A_{ij} = \frac{1}{2}(T_{ij} - T_{ji})$.

By inspection, if we swap the indices i and j, we see that $S_{ji} = S_{ij}$ (it is symmetric) and $A_{ji} = -A_{ij}$ (it is antisymmetric). As we will see, in physics, these two mathematical halves often represent entirely different physical phenomena.

Kinematics: Rate of Strain and Vorticity

Consider the velocity gradient tensor in a fluid, $\dfrac{\partial u_i}{\partial x_j}$. This tensor describes how the velocity changes as we move through space. Using our decomposition rule, we can split this into symmetric and antisymmetric parts:

$$\frac{\partial u_i}{\partial x_j} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{14.2}$$

We define the symmetric part as the **rate of strain tensor**, E_{ij}:

$$E_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{14.3}$$

This symmetric tensor physically represents the pure stretching and shearing deformation of a fluid element, without any rotation.

The antisymmetric part is the **spin tensor**, W_{ij}:

$$W_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right) \tag{14.4}$$

This tensor represents pure rotation without any deformation.

Because W_{ij} is an antisymmetric 3×3 tensor, it only has 3 independent components (the diagonal elements are all zero, and the off diagonal elements are mirror images with a minus sign thrown in). As a consequence, it must be relatable to a pseudovector, which also has three independent elements. Let's relate it to the classical vorticity vector, $\omega_k = \epsilon_{kpq}\dfrac{\partial u_q}{\partial x_p}$ (which is simply $\nabla \times \mathbf{u}$).

By multiplying the vorticity vector by the alternating tensor, it is straightforward to show the relationship between the vector and the tensor representations:

$$W_{ij} = -\frac{1}{2}\epsilon_{ijk}\omega_k$$

Solid Body Rotation

To understand the vorticity vector ω_i, let's apply index notation to a fluid undergoing pure solid body rotation with an angular velocity Ω_i. The velocity field for solid body rotation is given by the cross product $\mathbf{u} = \mathbf{\Omega} \times \mathbf{x}$, which in index notation is:

$$u_i = \epsilon_{ijk}\Omega_j x_k \tag{14.5}$$

What is the vorticity ω_p of this flow? We simply take the curl:

$$\omega_p = \epsilon_{pqi}\frac{\partial u_i}{\partial x_q} = \epsilon_{pqi}\frac{\partial}{\partial x_q}(\epsilon_{ijk}\Omega_j x_k)$$

Since Ω_j is a constant for solid body rotation, we pull it out. The derivative of the position vector x_k with respect to x_q is simply the Kronecker delta δ_{kq}:

$$\omega_p = \epsilon_{pqi}\epsilon_{ijk}\Omega_j\delta_{kq}$$

Applying the delta to change k to q, we get:

$$\omega_p = \epsilon_{pqi}\epsilon_{ijq}\Omega_j$$

The quantity $\epsilon_{pqi}\epsilon_{ijq}$ is the product of two isotropic tensors and has two free indices. Thus we know that it must be a member of the class of isotropic second order tensors: $\epsilon_{pqi}\epsilon_{ijq} = \lambda\delta_{pj}$ where λ is a constant. This is easily determined by multiplying both sides of the above expression by δ_{pj}. The right hand side yields $\lambda\delta_{jj} = 3\lambda$. The left hand side yields $\epsilon_{jqi}\epsilon_{ijq} = 3(1)^2 + 3(-1)^2 = 6$. Thus $\lambda = 2$ and $\epsilon_{pqi}\epsilon_{ijq} = 2\delta_{pj}$. Putting this together with the expression for ω_p we have:

$$\omega_p = 2\delta_{pj}\Omega_j = 2\Omega_p$$

Thus, we mathematically prove that the vorticity ω_i is exactly **twice** the angular velocity Ω_i of the fluid. Let's apply this!

Rotation of a Sphere in a Linear Flow Field

We examine a fundamental problem in suspension mechanics at low Reynolds numbers (Stokes flow): a sphere placed in a general linear flow field $u_i = (E_{ij} + W_{ij})x_j$. Does the sphere rotate?

Because of the linearity of the governing Stokes flow equations, the hydrodynamic torque L_i (a pseudovector) exerted by the fluid on the sphere must be a linear function of the flow field components: the rate of strain E_{jk} and the fluid vorticity ω_j.

$$L_i = C_{ijk}E_{jk} + D_{ij}\omega_j \tag{14.6}$$

Because the sphere is perfectly geometrically isotropic, the tensors C_{ijk} and D_{ij} must be constructed strictly from isotropic tensors (δ_{ij} and ϵ_{ijk}).

For C_{ijk} (which maps a symmetric tensor E_{jk} to a pseudovector L_i), it must be an isotropic pseudotensor of rank 3. The only one that exists is the alternating tensor, so $C_{ijk} = \lambda\epsilon_{ijk}$.

Thus, the torque contribution from the straining flow is $\lambda\epsilon_{ijk}E_{jk}$. However, we know that the inner product of an antisymmetric tensor (ϵ_{ijk}) and a symmetric tensor (E_{jk}) is identically **zero**! Therefore, a pure straining flow exerts absolutely no torque on a sphere.

For D_{ij} (which maps a pseudovector ω_j to a pseudovector L_i), it must be an isotropic physical tensor of rank 2, meaning $D_{ij} = \lambda\delta_{ij}$. Thus, the torque is simply proportional to the fluid vorticity. To experience zero net torque, the sphere must rotate with the local surrounding fluid. Since vorticity is twice the angular velocity of the fluid, the sphere's angular velocity is exactly half the local vorticity:

$$\Omega_i^{sphere} = \frac{1}{2}\omega_i \tag{14.7}$$

We achieved this result entirely through symmetry and index notation, without integrating a single stress profile over the surface of the sphere! Note that this is specific to **Stokes flow** only, because it relies on linearity. If inertia is important the non-linear terms in the Navier-Stokes equations yield a different result. It also applies only to spheres - if you have a rod or any body of revolution the additional director leads to a non-zero contribution to the rotation from E_{ij}. This causes non-spherical particles to continuously tumble in a shear flow, a complex periodic motion known as **Jeffery's Orbits**.

Application: Settling of a Body of Revolution at Zero Re

This powerful principle—that the *linearity* of Stokes flow dictates that "effects" (like translational or angular velocities) must be directly proportional to "causes" (like applied forces or shear)—extends far beyond simple spheres. We can use this exact same tensor logic to analyze the settling of non-spherical particles, yielding some of the most elegant simplifications in suspension mechanics.

Consider the following problem. Suppose we have a body of revolution whose orientation is specified by the unit vector p_i, that is settling under gravity with a net force F_i (physical vector) in a fluid otherwise at rest. At very low Reynolds number (Re), how does its angular velocity Ω_i (pseudovector) depend on p_i?

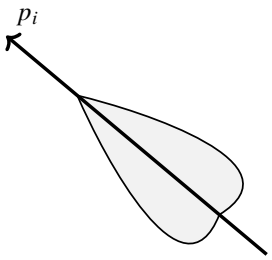


Figure 14.1: A schematic representation of an asymmetric body of revolution with its orientation director vector p_i.

At low Re, we can show that the angular velocity is proportional to the force F_i. Thus:

$$\Omega_i = A_{ik} F_k$$

where A_{ik} must be a **pseudotensor** which depends only on p_i and the object's shape!

There is only one way to do this!!

$$A_{ik} = \lambda \epsilon_{ijk} p_j$$

where λ is some scalar!

Thus, the angular velocity is:

$$\Omega_i = \lambda \epsilon_{ijk} p_j F_k$$

and a single experiment can determine λ, which is constant for all orientations!

Likewise, if the object has fore-and-aft symmetry (e.g., an oblate spheroid such as an M&M candy, which looks the same for $\mathbf{p}$ and $-\mathbf{p}$ orientations), we have that A_{ik} must be an *even* function of $\mathbf{p}$. Since the only possible non-zero pseudotensor form of A_{ik} is odd in p_i, λ must be zero for such objects!

Thus, for example, rods and disks (fore-aft symmetric cylinders) don't rotate when settling in an unbounded fluid at low Re, regardless of orientation!

Settling Velocity

We can also look at the settling velocity U_i (physical vector) for some applied force F_j:

$$U_i = B_{ij} F_j$$

Here B_{ij} is a **physical tensor** which depends on p_i. The most general physical tensor form is:

$$B_{ij} = \lambda_1 \delta_{ij} + \lambda_2 p_i p_j$$

Thus, the settling velocity is:

$$U_i = (\lambda_1 \delta_{ij} + \lambda_2 p_i p_j) F_j$$

where λ_1 and λ_2 must be determined from experiment or calculation.

Actually, by measuring the settling velocity of a rod broadside-on and end-on, you can get λ_1 and λ_2, allowing you to calculate U_i for *all* orientations—including the lateral velocity for inclined rods! This makes a nice classroom demonstration of the linearity of Stokes flow as you can easily predict and observe the sideways drift of a settling rod or disk from two simple time measurements.

Isotropic Settling of a Regular Polyhedron

Finally, let us extend this symmetry logic to a remarkable conclusion regarding regular polyhedra (like a cube). If a cube settles under gravity at zero Reynolds number, does its drift velocity depend on the orientation? Will it settle faster if it is point on or flat side on?

The settling velocity is again proportional to the force:

$$U_i = M_{ij} F_j$$

Because the governing Stokes equations are linear, M_{ij} is a physical tensor that must share the geometric symmetries of the cube itself.

If we align our coordinate axes with the faces of the cube, the cube looks identical under a 90° rotation about any axis. A 90° rotation about the x_3-axis transforms $x_1 \to x_2$ and $x_2 \to -x_1$. For the tensor M_{ij} to remain invariant under this rotation, the diagonal components must be equal ($M_{11} = M_{22}$). Applying this logic to all three axes dictates that $M_{11} = M_{22} = M_{33}$.

Furthermore, a 180° rotation about the x_1-axis ($x_2 \to -x_2$, $x_3 \to -x_3$) forces the off-diagonal components (like M_{12}) to equal their own negative ($-M_{12}$), meaning all off-diagonal components must be strictly zero. An alternative way of seeing this is that if the force is in the same direction as one of the faces, there will not be any sideways drift (the velocity and force must be in the same direction due to symmetry).

Therefore, the mobility matrix for a cube reduces to:

$$M_{ij} = \begin{bmatrix} M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & M \end{bmatrix} = M\delta_{ij}$$

Substituting this back into our velocity equation yields:

$$U_i = M\delta_{ij} F_j = M F_i$$

The mobility tensor is thus **isotropic**. A cube falls with the same velocity regardless of orientation, just like a sphere, even though it is not itself isotropic. The same would apply to any regular polyhedra such as a tetrahedron, which is a rather surprising result if you examine the shape. It is strictly a consequence of linearity, though, and would **not** be true at finite Re where fluid inertia makes the problem non-linear.

Practice Problems

Problem 1: Rotation in Simple Shear Flow

A classic problem in fluid mechanics is simple shear flow, such as the flow between two parallel plates where one plate is moving. The velocity field is given by:

$$u_i = \dot{\gamma}\delta_{i1}x_2$$

where $\dot{\gamma}$ is the constant shear rate. Determine the vorticity vector ω_k of this flow field, and use the results from this chapter to find the angular velocity Ω_i of a freely suspended, force-free sphere placed in this flow.

Problem 2: Trajectory and Drift of a Settling Rod

In a classroom demonstration of Stokes flow, a slender rod is dropped into a tall beaker of viscous fluid. When oriented horizontally (broadside, $p_i = \delta_{i1}$), it takes a time $t_\perp$ to fall a height H. When oriented vertically (end-on, $p_i = \delta_{i3}$), it takes a time $t_\parallel$ to fall the same height H.

The settling velocity is given by the mobility tensor equation $U_i = (\lambda_1\delta_{ij} + \lambda_2 p_i p_j)F_j$, where gravity acts downward such that $F_j = -F\delta_{j3}$.

(a) Find $\lambda_1 F$ and $\lambda_2 F$ in terms of H, $t_\perp$, and $t_\parallel$.

(b) Suppose the rod is released at an angle θ to the vertical in the $x_1 - x_3$ plane ($p_1 = \sin\theta$, $p_3 = \cos\theta$). Derive an expression for its trajectory, specifically the ratio of its lateral drift velocity to its downward velocity (U_1/U_3). Show that this trajectory depends only on the shape of the rod and is completely independent of the force F (its mass).

(c) Predict the horizontal distance Δx the rod will drift by the time it reaches the bottom of the beaker (falling a vertical distance H), purely in terms of H, θ, $t_\perp$, and $t_\parallel$.

Problem 3: Incompressibility and the Tensor Trace

The continuity equation for an incompressible fluid states that the divergence of the velocity field must be zero: $\nabla \cdot \mathbf{u} = 0$. In index notation, this is written as $\dfrac{\partial u_i}{\partial x_i} = 0$.

Using the definitions of the rate of strain tensor E_{ij} and the spin tensor W_{ij}, prove that the trace of the rate of strain tensor (E_{ii}) is equal to the divergence of the velocity field. Furthermore, prove that the trace of the spin tensor (W_{ii}) is identically zero for *any* flow field, compressible or not.

Solutions to Practice Problems

Solution 1: Rotation in Simple Shear Flow

First, we calculate the vorticity vector using the alternating tensor:

$$\omega_k = \epsilon_{kpq} \frac{\partial u_q}{\partial x_p}$$

Substitute the given velocity field:

$$\omega_k = \epsilon_{kpq} \frac{\partial}{\partial x_p} (\dot{\gamma} \delta_{q1} x_2)$$

Since $\dot{\gamma}$ is constant, we pull it out. The derivative $\dfrac{\partial x_2}{\partial x_p}$ is simply the Kronecker delta δ_{p2}.

$$\omega_k = \dot{\gamma} \epsilon_{kpq} \delta_{q1} \delta_{p2}$$

Applying the deltas to change the indices of the alternating tensor ($q \to 1$ and $p \to 2$):

$$\omega_k = \dot{\gamma} \epsilon_{k21}$$

Since ϵ_{k21} is only non-zero when the indices are all different (meaning k must be 3), it behaves exactly like the Kronecker delta δ_{k3}. Thus, we have (swapping two indices of the alternating tensor flips the sign) $\epsilon_{k21} = -\epsilon_{12k} = -\delta_{k3}$, and:

$$\omega_k = -\dot{\gamma} \delta_{k3}$$

The fluid vorticity is entirely in the negative x_3-direction (into the page). Because a pure straining flow exerts no torque, the sphere rotates with exactly half the fluid vorticity:

$$\Omega_i^{sphere} = -\frac{\dot{\gamma}}{2} \delta_{i3}$$

Solution 2: Trajectory and Drift of a Settling Rod

(a): For the broadside-on case ($p_i = \delta_{i1}$), the downward velocity is $-H/t_\perp$:

$$-\frac{H}{t_\perp} = U_3 = (\lambda_1 \delta_{33} + \lambda_2 p_3 p_3)(-F)$$

Since $p_3 = 0$ for a horizontal rod, this yields $\lambda_1 F = \frac{H}{t_\perp}$.

For the end-on case ($p_i = \delta_{i3}$), the downward velocity is $-H/t_\parallel$:

$$-\frac{H}{t_\parallel} = U_3 = (\lambda_1 \delta_{33} + \lambda_2 p_3 p_3)(-F)$$

Since $p_3 = 1$ for a vertical rod, we get $(\lambda_1 + \lambda_2)F = \frac{H}{t_\parallel}$. Substituting $\lambda_1 F$, we find $\lambda_2 F = H\left(\frac{1}{t_\parallel} - \frac{1}{t_\perp}\right)$.

(b): For the inclined rod, the lateral velocity U_1 and downward velocity U_3 are:

$$U_1 = (\lambda_1 \delta_{13} + \lambda_2 p_1 p_3)(-F) = -\lambda_2 F \sin\theta\cos\theta$$
$$U_3 = (\lambda_1 \delta_{33} + \lambda_2 p_3 p_3)(-F) = -F(\lambda_1 + \lambda_2 \cos^2\theta)$$

The trajectory is the ratio of these velocities:

$$\frac{U_1}{U_3} = \frac{-\lambda_2 F \sin\theta\cos\theta}{-F(\lambda_1 + \lambda_2\cos^2\theta)} = \frac{\lambda_2 \sin\theta\cos\theta}{\lambda_1 + \lambda_2\cos^2\theta}$$

Notice that the force F perfectly cancels out! The trajectory is determined entirely by the geometry of the rod (λ_1 and λ_2) and its orientation (θ), independent of its weight.

(c): We multiply the numerator and denominator of our trajectory ratio by F and substitute the experimental values from Part 1:

$$\frac{U_1}{U_3} = \frac{H\left(\frac{1}{t_\parallel} - \frac{1}{t_\perp}\right)\sin\theta\cos\theta}{\frac{H}{t_\perp} + H\left(\frac{1}{t_\parallel} - \frac{1}{t_\perp}\right)\cos^2\theta}$$

The height H also cancels out. Multiplying top and bottom by $t_\perp t_\parallel$ yields a very clean expression:

$$\frac{U_1}{U_3} = \frac{(t_\perp - t_\parallel)\sin\theta\cos\theta}{t_\parallel + (t_\perp - t_\parallel)\cos^2\theta}$$

Since both velocities are constant in Stokes flow, the ratio of the velocities is exactly equal to the ratio of the distances traveled ($\Delta x/H$). Thus, the total horizontal drift Δx is:

$$\Delta x = H\left[\frac{(t_\perp - t_\parallel)\sin\theta\cos\theta}{t_\parallel + (t_\perp - t_\parallel)\cos^2\theta}\right]$$

This allows you to predict the exact landing spot of the rod using only a stopwatch and a protractor!

Solution 3: Incompressibility and the Tensor Trace

To find the trace of a tensor, we set the two indices to be equal ($i = j$) and invoke the Einstein summation convention.

For the rate of strain tensor:

$$E_{ii} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_i} + \frac{\partial u_i}{\partial x_i}\right) = \frac{1}{2}\left(2\frac{\partial u_i}{\partial x_i}\right) = \frac{\partial u_i}{\partial x_i} = \nabla \cdot \mathbf{u}$$

Thus, for an incompressible fluid, $E_{ii} = 0$.

For the spin tensor:

$$W_{ii} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_i} - \frac{\partial u_i}{\partial x_i}\right) = 0$$

This relies purely on the fact that any term subtracted from itself is zero. It reflects the geometric reality that the diagonal elements of any antisymmetric tensor are strictly zero, meaning pure rotation involves absolutely no volumetric expansion or contraction.

15 Index Notation and Laplace's Equation

In transport phenomena, we frequently encounter Laplace's equation, $\nabla^2\phi = 0$. It governs steady-state heat conduction, electrostatics, mass diffusion, and irrotational fluid flow. Solutions to Laplace's equation are known as **harmonic functions**.

For unbounded 3D problems where the influence of an object must decay to zero at infinity, the fundamental solutions are known as *decaying solid spherical harmonics*. Traditionally, these are taught using separation of variables in spherical coordinates, leading to a rather cumbersome series of Legendre polynomials. However, index notation yields a surprisingly simple, elegant, and purely Cartesian way to generate and use these harmonics.

Generating Spherical Harmonics

Since the Laplacian operator $\nabla^2 = \dfrac{\partial^2}{\partial x_k \partial x_k}$ commutes with spatial derivatives, any spatial derivative of a harmonic function is identically another harmonic function.

Our fundamental (zeroth-order) decaying harmonic is simply the inverse of the radial distance:

$$H^{(0)} = \frac{1}{r}$$

Since $\nabla^2(1/r) = 0$ everywhere except at the origin, we can generate a family of higher-order tensor harmonics by taking successive derivatives of this base function:

$$\frac{\partial}{\partial x_i}\left(\frac{1}{r}\right), \quad \frac{\partial^2}{\partial x_i \partial x_j}\left(\frac{1}{r}\right), \quad \cdots$$

Let us carefully evaluate the first derivative, as the chain rule can be slightly tricky at first glance. First, note that the radial distance squared is simply the inner product of the position vector with itself:

$$r^2 = x_k x_k$$

To find how r changes with respect to a spatial coordinate x_i, we differentiate both sides:

$$\frac{\partial r^2}{\partial x_i} = \frac{\partial}{\partial x_i}(x_k x_k)$$

Applying the chain rule and product rule gives:

$$2r\frac{\partial r}{\partial x_i} = \frac{\partial x_k}{\partial x_i}x_k + x_k\frac{\partial x_k}{\partial x_i}$$

Since x_k and x_i are orthogonal Cartesian coordinates, their derivative is simply the Kronecker delta δ_{ki}:

$$2r\frac{\partial r}{\partial x_i} = \delta_{ki}x_k + x_k\delta_{ki} = 2x_i$$

Dividing by $2r$ yields a tremendously useful identity:

$$\frac{\partial r}{\partial x_i} = \frac{x_i}{r}$$

Now, we apply the chain rule to our base harmonic $1/r$ (which is r^{-1}):

$$\frac{\partial}{\partial x_i}\left(r^{-1}\right) = -r^{-2}\frac{\partial r}{\partial x_i} = -r^{-2}\left(\frac{x_i}{r}\right) = -\frac{x_i}{r^3}$$

Because any constant multiple of a harmonic function is still a harmonic function, we can absorb the negative sign to define our first-order (vector) harmonic. Continuing this process of differentiation yields the standard family of decaying harmonics:

$$H^{(0)} = \frac{1}{r} \tag{15.1}$$

$$H_i^{(1)} = \frac{x_i}{r^3} \tag{15.2}$$

$$H_{ij}^{(2)} = \frac{x_i x_j}{r^5} - \frac{\delta_{ij}}{3r^3} \tag{15.3}$$

Note: We have absorbed scalar constants into the definition of the harmonics for convenience, as this may be done without loss of generality.

Example: Steady-State Conduction from a Sphere

Consider a sphere of radius a held at a constant temperature T_0 in an unbounded domain that decays to $T = 0$ at infinity. The governing equation is steady heat conduction: $\nabla^2 T = 0$.

Because temperature is a scalar, and the boundary condition T_0 is a scalar, our solution must be built strictly from a scalar harmonic. The only purely isotropic scalar harmonic we have is $H^{(0)}$. Thus, by inspection:

$$T = C\frac{1}{r}$$

Applying the boundary condition at the surface of the sphere ($r = a$):

$$T(a) = \frac{C}{a} = T_0 \implies C = aT_0$$

The exact temperature field is therefore $T = T_0\frac{a}{r}$.

The Uniqueness Theorem: Why "Guessing" Works

You might be wondering: we just deduced the form of the solution using scalar symmetry, found a constant that satisfied the boundary conditions, and abruptly declared the problem solved. Is it really that simple? What if there is another, more complicated temperature field that also works?

This is where the **Uniqueness Theorem** for linear partial differential equations comes to our rescue. The theorem states that for linear equations like Laplace's equation (and, as we will see, the creeping flow equations), if you can find *any* solution that satisfies both the governing differential equation and all the boundary conditions, it is the *only* mathematically possible solution.

Because of uniqueness, we do not have to grind through the formal derivation of solutions using tedious infinite series or separation of variables. If we can creatively assemble a combination of harmonic tensors that matches the physics and the geometry at the boundaries, we can stop right there. The uniqueness theorem gives us the mathematical license to "guess" solutions by inspection using our index notation building blocks, guaranteeing that we have the one and only correct answer.

All this applies only to linear systems: if a problem is non-linear in the dependent variables then there is often more than one possible solution. For example, in Stokes flow (zero Reynolds number) the governing equations are linear, and there is only a unique solution. At finite Re, however, the more general Navier-Stokes equations apply with their non-linear convective terms and uniqueness is no longer guaranteed. That is why flow through a tube can yield both laminar flow (e.g., Poiseuille flow) and turbulent flow in the exact same geometry!

Example: Dipole Surface Temperature

Now consider a more interesting case. Suppose the sphere is subjected to a surface temperature distribution that varies linearly across it (a dipole distribution), such as $T(r = a) = G_i x_i$, where G_i is a constant applied temperature gradient vector.

Our temperature field T must still be a scalar, but it must now be linearly proportional to the vector G_i. To create a scalar from a vector, we must take the inner product of G_i with a vector harmonic. Therefore, we use the first-order harmonic $H_i^{(1)}$:

$$T = \lambda G_i H_i^{(1)} = \lambda G_i \frac{x_i}{r^3}$$

Applying our boundary condition at the surface ($r = a$):

$$T(a) = \lambda G_i \frac{x_i}{a^3} = G_i x_i$$

By matching coefficients, we see that $\lambda/a^3 = 1$, or $\lambda = a^3$. The resulting temperature field in the fluid is simply:

$$T = a^3 G_i \frac{x_i}{r^3}$$

Extensions to Complex Boundaries

Because Laplace's equation is linear, any arbitrary surface temperature distribution on a sphere can be expressed as a linear superposition (a sum) of these different harmonic combinations (monopole, dipole, quadrupole, etc.). Each geometric component of the boundary condition simply activates the corresponding tensor harmonic.

Furthermore, if the heated object were *not* a sphere (for instance, an ellipsoid or a settling rod), the geometry itself would introduce a new structural unit vector (e.g., p_i pointing along the primary axis). This would give us more vector building blocks, allowing for even richer (but, alas, more complicated) combinations of harmonics to satisfy the boundary conditions.

Application to Stokes Flow

Let's apply this machinery to the creeping flow (Stokes flow) equations:

$$\frac{1}{\mu}\frac{\partial p}{\partial x_i} = \frac{\partial^2 u_i}{\partial x_j^2}$$

$$\frac{\partial u_j}{\partial x_j} = 0$$

If we take the divergence of the creeping flow momentum equation, the right side becomes the Laplacian of the divergence of velocity. By continuity, this is zero!

$$\frac{1}{\mu}\frac{\partial^2 p}{\partial x_i^2} = \frac{\partial^2}{\partial x_j^2}\left(\frac{\partial u_i}{\partial x_i}\right) = 0$$

Thus, $\nabla^2 p = 0$. **The pressure field in Stokes flow is harmonic.**

Because $\nabla^2 u_i = \dfrac{1}{\mu}\dfrac{\partial p}{\partial x_i}$, we can break the velocity vector u_i into a particular solution $u_i^{(p)}$ and a homogeneous solution $u_i^{(h)}$:

$$u_i = u_i^{(p)} + u_i^{(h)}$$

If we know the harmonic pressure field p, it is easily verified by substitution that the particular solution takes the form:

$$u_i^{(p)} = \frac{x_i}{2\mu}p$$

Note that $u_i^{(p)}$ is not itself a harmonic function and, although it satisfies the Stokes flow equations, it would not in general satisfy any boundary conditions or the continuity equation. We fix this by adding a harmonic homogeneous solution $u_i^{(h)}$ that satisfies $\nabla^2 u_i^{(h)} = 0$, meaning it may be constructed directly from our family of spherical harmonics.

The Rotating Sphere

Let's solve for a sphere of radius a rotating in a fluid otherwise at rest. We want to determine the *disturbance velocity field* produced by the sphere subject to the surface boundary condition at $r = a$:

$$u_i|_{r=a} = \epsilon_{ijk}\Omega_j x_k$$

First, the pressure p must be a physical scalar, linear in the angular velocity pseudovector Ω_k, and constructed from a harmonic. The only way to form such a scalar from a pseudovector is via the alternating tensor. The most general form using the second harmonic would be:

$$p = \lambda\epsilon_{ijk}\Omega_k\left(\frac{x_i x_j}{r^5} - \frac{\delta_{ij}}{3r^3}\right)$$

However, the harmonic tensor bracket is symmetric in i and j, whereas ϵ_{ijk} is antisymmetric in i and j. The inner product of a symmetric and antisymmetric tensor is identically zero. Therefore, $p = 0$.

Because the pressure is zero, the particular solution vanishes ($u_i^{(p)} = 0$). The entire velocity field is given by the homogeneous solution, which must be a physical vector, linear in the pseudovector Ω_j. Because of the symmetry of higher order harmonics, the only decaying vector harmonic that can contribute is:

$$u_i = \lambda\epsilon_{ijk}\Omega_j\frac{x_k}{r^3}$$

Applying the boundary condition at $r = a$:

$$u_i|_{r=a} = \lambda\epsilon_{ijk}\Omega_j\frac{x_k}{a^3} = \epsilon_{ijk}\Omega_j x_k$$

By matching coefficients, $\lambda = a^3$. The flow field around the rotating sphere is simply:

$$u_i = a^3\epsilon_{ijk}\Omega_j\frac{x_k}{r^3}$$

This is known as a *couplet* or *rotlet* and is one of the fundamental singularities in Stokes flow: the disturbance velocity distribution produced by a point torque.

Rigid Sphere in a Translating Flow

Now let's analyze the classic problem of a uniform flow U_i moving past a stationary sphere. The far-field boundary condition is $u_i \to U_i$ as $r \to \infty$, and the no-slip condition is $u_i = 0$ at $r = a$. Note that this problem is the same as a sphere translating with a velocity U_i in a fluid at rest.

By inspection, the scalar pressure p must be linear in U_i and constructed from a decaying solid harmonic. The only option is the physical scalar formed from the vector harmonic:

$$p = \lambda U_j\frac{x_j}{r^3}$$

Using this pressure, our particular velocity solution becomes:

$$u_i^{(p)} = \frac{x_i}{2\mu} p = \frac{\lambda}{2\mu} U_j \frac{x_i x_j}{r^3}$$

We pair this with a homogeneous solution $u_i^{(h)}$ that is linear in U_i and by necessity built from the harmonics one higher and lower in the series than the harmonic used to obtain the scalar pressure. To satisfy the far-field boundary condition, we also add the constant background flow U_i:

$$u_i = U_i + \left[\frac{\lambda}{2\mu} \frac{x_i x_j}{r^3} + \lambda_1 \frac{\delta_{ij}}{r} + \lambda_2 \left(\frac{x_i x_j}{r^5} - \frac{\delta_{ij}}{3r^3} \right) \right] U_j$$

We find the three constants $(\lambda, \lambda_1, \lambda_2)$ using physical constraints. First, enforcing the incompressibility continuity equation ($\frac{\partial u_i}{\partial x_i} = 0$) and simplifying the index contractions yields:

$$\lambda_1 = \frac{\lambda}{2\mu}$$

Note that the second harmonic (multiplied by λ_2) vanishes from the divergence automatically, which is very convenient.

Substituting this back in, we apply the no-slip boundary condition at the sphere's surface ($u_i = 0$ at $r = a$):

$$0 = U_i + \left[\frac{\lambda}{2\mu} \left(\frac{\delta_{ij}}{a} + \frac{x_i x_j}{a^3} \right) + \lambda_2 \left(\frac{x_i x_j}{a^5} - \frac{\delta_{ij}}{3a^3} \right) \right] U_j$$

By grouping the isotropic δ_{ij} terms and the directional $x_i x_j$ terms independently, we force both groups to vanish identically. This algebraic exercise yields:

$$\lambda_2 = -\frac{a^2 \lambda}{2\mu} \quad \text{and} \quad \lambda = -\frac{3}{2}\mu a$$

Substituting these resolved constants back into our equations yields the exact, explicit solutions for the fluid pressure and velocity fields around a sphere in creeping flow:

$$p = -\frac{3a\mu}{2} U_j \frac{x_j}{r^3}$$

$$u_i = U_i - \frac{3}{4} a U_j \left\{ \frac{x_i x_j}{r^3} + \frac{\delta_{ij}}{r} \right\} + \frac{1}{4} a^3 U_j \left\{ \frac{3x_i x_j}{r^5} - \frac{\delta_{ij}}{r^3} \right\}$$

Once these pressure and velocity distributions are known, we can define the total hydrodynamic stress tensor $\sigma_{ij} = -p\delta_{ij} + \mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right)$. By integrating the traction vector ($\sigma_{ij} n_j$) over the total surface area of the sphere, we can determine the net hydrodynamic drag force F_i. This integration yields the famous result for Stokes' Law: $F_i = 6\pi\mu a U_i$.

Note that all these examples were done for spheres, which are isotropic shapes. For more general shapes such as bodies of revolution the additional vector that the problem depends on intrinsically introduces additional harmonics to the pressure and velocity, greatly complicating the calculations (and in general preventing analytic solutions from being found). The index notation approach is still of use, however, because in many problems you can represent a surface as a distribution of point sources - which are provided by the far-field limit of the solution for a translating sphere given above. Such calculations form the basis of much of the literature in the subject of microhydrodynamics, but are beyond what is considered here.

Practice Problems

Problem 1: Proof of the Particular Solution

In the derivation of the velocity field for Stokes flow, we decomposed the velocity into a particular and homogeneous solution, $u_i = u_i^{(p)} + u_i^{(h)}$, where the pressure field is known to be harmonic ($\nabla^2 p = 0$).

Prove that the proposed particular solution:

$$u_i^{(p)} = \frac{x_i}{2\mu} p$$

satisfies the momentum equation $\nabla^2 u_i^{(p)} = \frac{1}{\mu} \frac{\partial p}{\partial x_i}$.

Problem 2: Verifying a Harmonic Function

In the text, the first-order decaying vector harmonic is defined as $H_i^{(1)} = \frac{x_i}{r^3}$. Since it was generated by taking the spatial derivative of a harmonic function, it must itself be a harmonic function.

Prove this directly by showing that the Laplacian of $H_i^{(1)}$ is zero. That is, evaluate:

$$\frac{\partial^2}{\partial x_j \partial x_j} \left(\frac{x_i}{r^3} \right) = 0$$

Hint: You will need to use the product rule, the identity $\dfrac{\partial r}{\partial x_j} = \dfrac{x_j}{r}$*, and remember that* $\delta_{jj} = 3$.

Problem 3: Quadrupole Surface Temperature

Consider a sphere of radius a located at the origin in an unbounded domain. The surface of the sphere is subjected to a steady, complex temperature distribution given by:

$$T(r = a) = C_{ij} x_i x_j$$

where C_{ij} is a constant, symmetric, traceless tensor (meaning its trace $C_{ii} = 0$).

Using index notation and the appropriate solid spherical harmonic, find the steady-state temperature field $T(x_i)$ everywhere in the fluid ($r \geq a$). Explain what happens to the isotropic term in the harmonic tensor when contracted with C_{ij}.

Solutions to Practice Problems

Solution 1: Proof of the Particular Solution

We must apply the Laplacian operator $\nabla^2 = \dfrac{\partial^2}{\partial x_j \partial x_j}$ to our proposed particular solution. Since $\dfrac{1}{2\mu}$ is a constant, we can pull it out front:

$$\nabla^2 u_i^{(p)} = \frac{1}{2\mu} \frac{\partial^2}{\partial x_j \partial x_j}(x_i p)$$

First, we apply the inner derivative using the product rule:

$$\frac{\partial}{\partial x_j}(x_i p) = \frac{\partial x_i}{\partial x_j} p + x_i \frac{\partial p}{\partial x_j} = \delta_{ij} p + x_i \frac{\partial p}{\partial x_j}$$

Now we take the second derivative (the outer derivative) of this result:

$$\frac{\partial}{\partial x_j}\left(\delta_{ij} p + x_i \frac{\partial p}{\partial x_j}\right) = \delta_{ij} \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j}\left(x_i \frac{\partial p}{\partial x_j}\right)$$

Applying the product rule again to the second term:

$$= \frac{\partial p}{\partial x_i} + \left(\frac{\partial x_i}{\partial x_j}\frac{\partial p}{\partial x_j} + x_i \frac{\partial^2 p}{\partial x_j \partial x_j}\right)$$

$$= \frac{\partial p}{\partial x_i} + \delta_{ij} \frac{\partial p}{\partial x_j} + x_i(\nabla^2 p)$$

Applying the Kronecker delta in the middle term changes the index j to i. Furthermore, we are given that the pressure field is harmonic, so the last term vanishes ($\nabla^2 p = 0$).

$$= \frac{\partial p}{\partial x_i} + \frac{\partial p}{\partial x_i} + 0 = 2\frac{\partial p}{\partial x_i}$$

Multiplying back the $\dfrac{1}{2\mu}$ constant we pulled out at the beginning yields:

$$\nabla^2 u_i^{(p)} = \frac{1}{2\mu}\left(2\frac{\partial p}{\partial x_i}\right) = \frac{1}{\mu}\frac{\partial p}{\partial x_i}$$

This perfectly matches the required momentum equation, proving our particular solution is valid.

Solution 2: Verifying a Harmonic Function

We evaluate the Laplacian of $x_i r^{-3}$ by taking two successive spatial derivatives. First derivative:

$$\frac{\partial}{\partial x_j}\left(x_i r^{-3}\right) = \frac{\partial x_i}{\partial x_j} r^{-3} + x_i(-3r^{-4})\frac{\partial r}{\partial x_j}$$

Using $\frac{\partial x_i}{\partial x_j} = \delta_{ij}$ and $\frac{\partial r}{\partial x_j} = \frac{x_j}{r}$:

$$\frac{\partial}{\partial x_j}\left(x_i r^{-3}\right) = \delta_{ij} r^{-3} - 3x_i x_j r^{-5}$$

Now we take the derivative of this result with respect to x_j:

$$\frac{\partial}{\partial x_j}\left(\delta_{ij} r^{-3} - 3x_i x_j r^{-5}\right) = \delta_{ij}(-3r^{-4})\frac{x_j}{r} - 3\frac{\partial}{\partial x_j}(x_i x_j r^{-5})$$

The first term simplifies to: $-3\delta_{ij} x_j r^{-5} = -3x_i r^{-5}$.

Now we carefully apply the product rule to the second term (treating x_i, x_j, and r^{-5} as a triple product):

$$-3\left[\frac{\partial x_i}{\partial x_j} x_j r^{-5} + x_i \frac{\partial x_j}{\partial x_j} r^{-5} + x_i x_j (-5r^{-6})\frac{\partial r}{\partial x_j}\right]$$

Using $\frac{\partial x_i}{\partial x_j} = \delta_{ij}$, $\frac{\partial x_j}{\partial x_j} = \delta_{jj} = 3$, and $\frac{\partial r}{\partial x_j} = \frac{x_j}{r}$:

$$-3\left[\delta_{ij} x_j r^{-5} + x_i(3) r^{-5} - 5x_i x_j r^{-6}\left(\frac{x_j}{r}\right)\right]$$

Simplify by applying the delta to the first term, and recognizing that $x_j x_j = r^2$ in the final term:

$$-3\left[x_i r^{-5} + 3x_i r^{-5} - 5x_i r^2 r^{-7}\right] = -3\left[4x_i r^{-5} - 5x_i r^{-5}\right] = -3[-x_i r^{-5}] = +3x_i r^{-5}$$

Adding the first term and the second term together yields:

$$-3x_i r^{-5} + 3x_i r^{-5} = 0$$

The Laplacian is exactly zero, verifying the function is harmonic.

Solution 3: Quadrupole Surface Temperature

The temperature field T must be a scalar, it must be linearly proportional to the constant tensor C_{ij}, and it must decay at infinity. To create a scalar from a second-order tensor, we must take the inner product of C_{ij} with our second-order solid harmonic $H^{(2)}_{ij}$.

$$T = \lambda C_{ij} H^{(2)}_{ij} = \lambda C_{ij}\left(\frac{x_i x_j}{r^5} - \frac{\delta_{ij}}{3r^3}\right)$$

Let us distribute the contraction.

$$T = \lambda\left(C_{ij}\frac{x_i x_j}{r^5} - C_{ij}\frac{\delta_{ij}}{3r^3}\right)$$

When C_{ij} is contracted with the Kronecker delta, it forces $i = j$, yielding the trace of the tensor C_{ii}. Because we are given that C_{ij} is traceless ($C_{ii} = 0$), the entire isotropic second term vanishes identically!

$$T = \lambda C_{ij} \frac{x_i x_j}{r^5}$$

Now we apply the boundary condition at the surface of the sphere ($r = a$):

$$T(a) = \lambda C_{ij} \frac{x_i x_j}{a^5} = C_{ij} x_i x_j$$

By matching coefficients, we see that $\frac{\lambda}{a^5} = 1$, which means $\lambda = a^5$. The final steady-state temperature field in the fluid is:

$$T = a^5 C_{ij} \frac{x_i x_j}{r^5}$$

It is interesting to note what happens if the tensor C_{ij} is *not* traceless. In this case you would get an additional contribution to the solution involving the zero harmonic such that:

$$T = \lambda C_{ij} \left(\frac{x_i x_j}{r^5} - \frac{\delta_{ij}}{3r^3} \right) + \lambda_2 C_{ij} \frac{\delta_{ij}}{r}$$

The extra term (also determined from satisfying the boundary condition at $r = a$) corresponds to some non-zero average temperature at the surface. Because of linearity the two effects can simply be added together!

Part IV

Analytical Solutions to PDEs

16 Formulating Boundary Conditions

Students often have difficulty formulating the boundary conditions which are necessary to solve both Ordinary Differential Equations (ODEs) and Partial Differential Equations (PDEs). In this chapter, we shall take a "step back" and examine the types of conditions usually found in transport problems before we dive into analytical solutions.

Boundary Conditions in ODEs

Because ODEs are in one dimension, boundary conditions or initial conditions are located at discrete points along the range of the independent variable.

The number of required boundary conditions is determined by the order of the differential equation: a first-order equation requires one (usually the initial condition), a second-order requires two, and so forth. These conditions may be specified entirely at the beginning of the domain (e.g., at $x = 0$ if you are solving over the domain $[0, 1]$), or they can be distributed at both ends.

Note that the order of the differential equation determines the number of conditions required, but not necessarily the number of conditions applied at physical boundaries. Integral constraints, boundedness requirements, and interface conditions can also supply the needed mathematical information.

Specified Value (Dirichlet Condition)

The simplest boundary condition is simply specifying the value of the function itself. For example, you could specify the temperature T at both $x = 0$ and $x = 1$ for a second-order heat conduction equation. In fluid mechanics, the "no-slip" condition at a stationary solid wall ($U = 0$) is a specified value condition.

Specified Derivative (Neumann Condition)

Another common boundary condition is the derivative condition. This arises because, in transport, the flux of mass, momentum, or energy via diffusion is proportional to the derivative. A constant heat flux would specify $-k\dfrac{dT}{dx} = q_0$ at a particular location. Insulated (or no-flux) boundary conditions simply dictate that the derivative is equal to zero at the boundary.

A zero-derivative condition also frequently arises from symmetry. For plane-Poiseuille flow through a slit, for example, the velocity is an even function of y (if $y = 0$ is the centerline). We can take advantage of this by only solving over half the domain and setting the derivative to be zero at the plane of symmetry ($\frac{du}{dy} = 0$).

Mixed (Robin) Boundary Conditions

Often you will have a boundary condition which specifies a relationship between both the function and its derivative. This is called a canonical Robin boundary condition. This occurs, for example, if there is a convective heat transfer coefficient h relating the conductive heat flux at a surface to a temperature differential between the surface and the bulk fluid (T_∞):

$$-k \left.\frac{dT}{dx}\right|_{surface} = h(T_{surface} - T_\infty)$$

Interface Conditions

A somewhat more complicated problem is when you have an interface between two discrete layers with different properties (in heat transfer, it could be two materials with different thermal conductivities). In this case, you have a pair of interface conditions.

First, in the absence of an interfacial resistance (such as a thermal contact resistance), the function (e.g., temperature) must be continuous across the boundary. Second, in general, the flux must be continuous across the boundary as well. If the conductivities are different, this means that the derivative itself is discontinuous:

$$-k_1 \left.\frac{dT_1}{dx}\right|_{interface} = -k_2 \left.\frac{dT_2}{dx}\right|_{interface}$$

There are exceptions to this flux continuity condition, notably the case of a phase change (such as a melting boundary), where the discontinuity in the heat flux is chewed up by the latent heat of melting.

Surface Sources and Sinks (Flux-Jump Conditions)

While flux is generally continuous across an interface, this continuity is broken—resulting in a "jump" in the flux—if mass or energy is being generated or consumed exactly at the surface. A classic chemical engineering example is heterogeneous catalysis. If a species diffuses to a solid wall and undergoes a first-order surface reaction, the diffusive flux to the wall is exactly balanced by the rate of consumption at the wall:

$$-D \left.\frac{dc}{dx}\right|_{surface} = k_s c_{surface}$$

where D is the diffusion coefficient and k_s is the surface reaction rate constant. Mathematically, this takes the form of a Robin condition, but its physical origin is a reactive source or sink at the boundary rather than convective transport to a bulk fluid.

Boundedness (or Singularity) Conditions

In many coordinate systems, particularly cylindrical and spherical coordinates, the mathematical form of the differential equation introduces a singularity at the origin ($r = 0$). For example, solving the steady-state heat equation in a solid cylinder without heat generation yields a general solution of the form:

$$T(r) = C_1 \ln(r) + C_2$$

Because the physical temperature at the center of the solid cylinder cannot be infinite, the mathematics must reflect this physical reality. We apply a *boundedness condition*, requiring that the solution remains finite at $r = 0$. This immediately forces the constant multiplying the singular term to be zero ($C_1 = 0$). Note that if the point $r = 0$ is outside the domain (e.g., an annulus rather than a solid cylinder) the logarithmic term would survive - and be determined from the now two boundary locations.

Asymptotic Conditions (Semi-Infinite Domains)

In some transport problems, the physical domain is so large compared to the region where transport is actively occurring that we can treat it mathematically as extending to infinity. This is common in boundary layer flows or short-time heat conduction into a thick slab.

Instead of specifying a condition at a discrete physical boundary like $y = L$, we specify an asymptotic condition as the coordinate approaches infinity. For example, in flow over a flat plate, the fluid velocity must smoothly approach the free-stream velocity (U_∞) far from the wall:

$$u_x \to U_\infty \quad \text{as} \quad y \to \infty$$

Periodic Boundary Conditions

In some problems, the geometry or the physics repeats itself over a characteristic length scale L. Examples include flow through a packed bed, heat transfer in a repeating array of cooling fins, or flow through a periodic porous medium. Instead of specifying a fixed value or flux, we enforce that the state of the system at the beginning of the unit cell is identical to the state at the end:

$$T(0) = T(L) \quad \text{and} \quad \left.\frac{dT}{dx}\right|_{x=0} = \left.\frac{dT}{dx}\right|_{x=L}$$

Periodic conditions are particularly important in computational fluid dynamics and advanced analytical methods like Taylor-Aris dispersion analyses.

Integral Conditions

Somewhat less frequently, you encounter an integral condition. This occurs, for example, if in flow through a slit you do not know the pressure gradient (thus there is an unknown parameter in the differential equation), but you do know the total flow rate, which is the integral of the velocity.

Such problems are usually solved by calculating the velocity profile in terms of the unknown pressure gradient (applying the known spatial boundary conditions) and then integrating that profile. This

result is set equal to the prescribed total flow rate, thereby determining the necessary pressure gradient.

Unknown Boundary Locations

Still another case is where the location of a boundary is unknown, and must be determined in the course of the solution. As an example, suppose you pour fluid down an incline at some prescribed flow rate per unit width Q/W. The unknown here would be the thickness of the resulting fluid layer. Again, you would solve for the velocity profile in terms of the unknown thickness, and then integrate to determine the required thickness to satisfy the integral flow condition.

Summary of Boundary Condition Types

Table 16.1 provides a quick reference for the most common types of boundary conditions encountered in transport phenomena.

Table 16.1: Common boundary conditions and their physical origins.

Type	Mathematical Form	Physical Origin
Dirichlet	$T = T_0$	Specified value at a boundary
Neumann	$-k\dfrac{\partial T}{\partial x} = q_0$	Specified flux (or insulated if $q_0 = 0$)
Robin	$aT + b\dfrac{\partial T}{\partial x} = c$	Coupled transport (e.g., convective cooling)
Interface	$-k_1 \left.\dfrac{\partial T_1}{\partial x}\right\|_i = -k_2 \left.\dfrac{\partial T_2}{\partial x}\right\|_i$	Conservation of flux across a boundary
Interface	$T_1\|_i = T_2\|_i$	Continuity across a boundary
Boundedness	$\|T\| < \infty$ at $r = 0$	Physical realizability at a singular point
Integral	$\int u\, dA = Q$	Global conservation or prescribed total rate
Asymptotic	$T \to T_\infty$ as $x \to \infty$	Semi-infinite domains (boundary layers)
Periodic	$T(0) = T(L)$	Repeating geometry or unit cells

Examples of Complex Conditions

Example 1: Couette-Poiseuille Flow (Integral Condition)

Let us apply an integral condition. Consider plane-Couette flow, where we impose a velocity U at the upper surface ($y = H$) and the lower surface is stationary ($y = 0$). However, we also impose an unknown pressure gradient $\dfrac{dp}{dx}$ along the flow direction. We want to find the specific pressure gradient necessary for the net volumetric flow to be zero. This is a 1D approximation of what happens in a large aspect ratio lid-driven cavity.

The governing ODE is:

$$\mu\frac{d^2u}{dy^2} = \frac{dp}{dx}$$

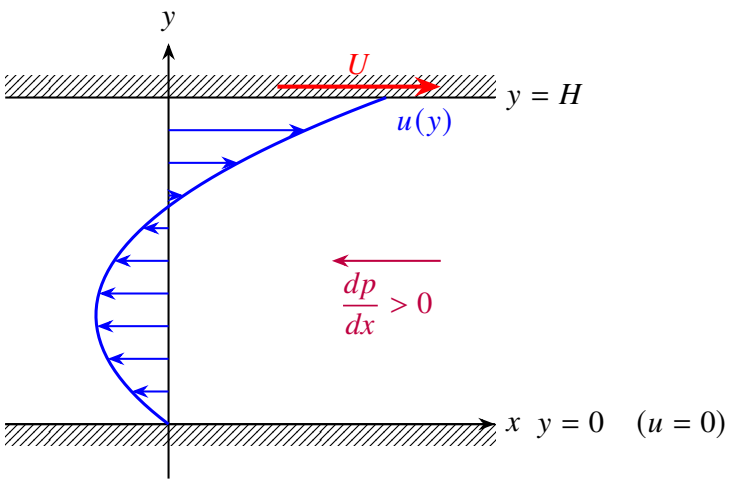


Figure 16.1: Couette-Poiseuille flow with zero net volumetric flow. The positive (adverse) pressure gradient exactly balances the forward pumping action of the upper plate.

Integrating twice yields the general solution:

$$u(y) = \frac{1}{2\mu}\frac{dp}{dx}y^2 + C_1 y + C_2$$

Applying our specified value boundary conditions, $u(0) = 0 \implies C_2 = 0$, and $u(H) = U$ allows us to solve for C_1:

$$C_1 = \frac{U}{H} - \frac{H}{2\mu}\frac{dp}{dx}$$

Substituting this back, we get the velocity profile in terms of the unknown pressure gradient:

$$u(y) = \frac{U}{H}y + \frac{1}{2\mu}\frac{dp}{dx}(y^2 - Hy)$$

Now we apply our integral condition. The net flow must be zero:

$$\int_0^H u(y)dy = \int_0^H \left[\frac{U}{H}y + \frac{1}{2\mu}\frac{dp}{dx}(y^2 - Hy)\right] dy = 0$$

Evaluating the integral gives:

$$\frac{UH}{2} - \frac{H^3}{12\mu}\frac{dp}{dx} = 0$$

Solving for the pressure gradient yields:

$$\frac{dp}{dx} = \frac{6\mu U}{H^2}$$

Note: The positive sign indicates that the pressure must increase in the +x direction, creating an adverse pressure gradient that exactly opposes the Couette pumping action of the upper plate.

Example 2: Falling Film (Unknown Boundary Location)

Now let us look at an unknown boundary location. Consider a fluid falling down a wall inclined at an angle β to the horizontal. We know the flow rate per unit width Q/W, but the film thickness h is unknown.

The governing ODE is:

$$\mu \frac{d^2u}{dy^2} + \rho g \sin\beta = 0$$

Integrating twice gives:

$$u(y) = -\frac{\rho g \sin\beta}{2\mu} y^2 + C_1 y + C_2$$

The boundary conditions are no-slip at the wall ($u(0) = 0 \implies C_2 = 0$) and a zero-shear (specified derivative) condition at the unknown surface ($y = h$):

$$\left.\frac{du}{dy}\right|_{y=h} = 0 \implies -\frac{\rho g \sin\beta}{\mu} h + C_1 = 0 \implies C_1 = \frac{\rho g \sin\beta}{\mu} h$$

This gives a velocity profile of:

$$u(y) = \frac{\rho g \sin\beta}{\mu} \left(hy - \frac{1}{2} y^2 \right)$$

To find the unknown boundary h, we integrate to match the prescribed flow rate:

$$\frac{Q}{W} = \int_0^h u(y) dy = \frac{\rho g \sin\beta}{\mu} \left(\frac{h^3}{2} - \frac{h^3}{6} \right) = \frac{\rho g \sin\beta}{3\mu} h^3$$

Solving for h yields the location of our unknown boundary:

$$h = \left(\frac{3\mu Q}{\rho g W \sin\beta} \right)^{1/3}$$

Example 3: Composite Nuclear Fuel Plate (Symmetry, Interface, and Robin Conditions)

Often, a single problem will require you to chain multiple types of boundary conditions together. Consider a nuclear fuel cell plate of thickness $2b$ with a uniform volumetric heat generation rate S. The fuel meat has a thermal conductivity k_f. It is clad on both sides by a Zircaloy layer of thickness δ and conductivity k_c. The plate is cooled by pressurized water at a bulk temperature T_a with a convective heat transfer coefficient h. We want to find the maximum temperature at the center of the fuel cell.

Because the problem is symmetric, we only need to solve over half the domain, setting our origin $x = 0$ at the centerline of the fuel plate. This gives us two regions: the fuel ($0 \le x \le b$) and the cladding ($b \le x \le b + \delta$).

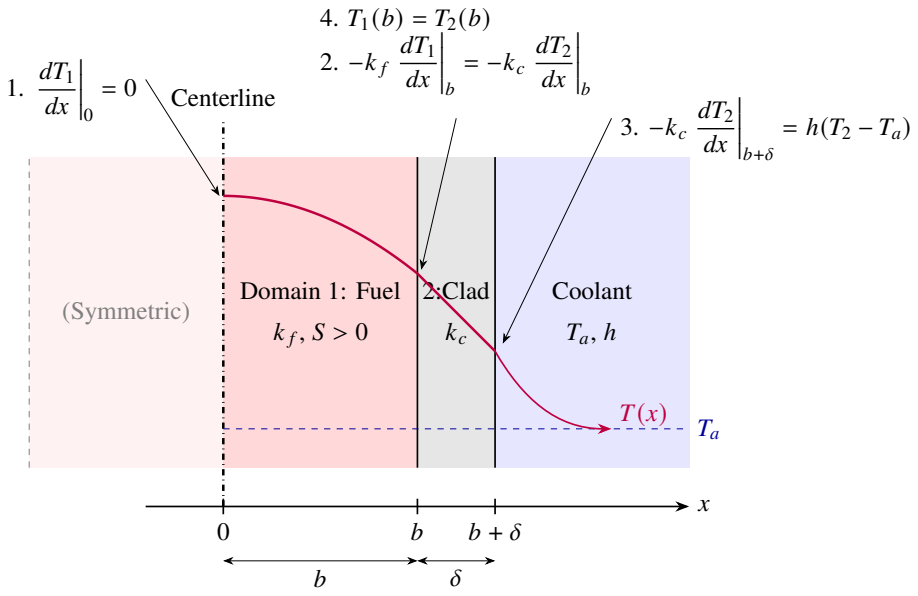


Figure 16.2: Cross-section of a composite nuclear fuel plate. The parabolic temperature profile $T_1(x)$ in the heat-generating fuel must smoothly transition into a linear profile $T_2(x)$ in the cladding, balancing conductive flux with convective cooling at the outer surface.

The governing ODEs for the temperature in the fuel (T_1) and the cladding (T_2) are:

$$k_f \frac{d^2T_1}{dx^2} + S = 0 \quad \text{and} \quad k_c \frac{d^2T_2}{dx^2} = 0$$

Integrating these twice gives our general solutions:

$$T_1(x) = -\frac{S}{2k_f}x^2 + C_1x + C_2$$
$$T_2(x) = C_3x + C_4$$

We have four unknown constants (C_1, C_2, C_3, C_4), so we need four boundary conditions.

1. **Symmetry Condition:** The temperature profile must be symmetric about the centerline ($x = 0$), meaning the derivative is zero.

$$\left.\frac{dT_1}{dx}\right|_{x=0} = 0 \implies C_1 = 0$$

2. **Interface Flux Continuity:** The heat flux leaving the fuel must equal the heat flux entering the cladding at $x = b$.

$$-k_f \left.\frac{dT_1}{dx}\right|_{x=b} = -k_c \left.\frac{dT_2}{dx}\right|_{x=b} \implies Sb = -k_cC_3 \implies C_3 = -\frac{Sb}{k_c}$$

Note that Sb is simply the total heat generated in the half-plate, which all must pass through the cladding.

3. **Robin (Convective) Condition:** At the outer edge of the cladding ($x = b + \delta$), the conductive flux must match the convective cooling to the fluid.

$$-k_c \left.\frac{dT_2}{dx}\right|_{x=b+\delta} = h(T_2(b+\delta) - T_a)$$

Since we already know the flux through the cladding is constant ($-k_c C_3 = Sb$), we can write:

$$Sb = h\left(-\frac{Sb}{k_c}(b+\delta) + C_4 - T_a\right)$$

Solving for C_4 yields:

$$C_4 = T_a + \frac{Sb}{h} + \frac{Sb(b+\delta)}{k_c}$$

4. **Interface Temperature Continuity:** The temperature must be continuous at the fuel-cladding interface ($x = b$).

$$T_1(b) = T_2(b) \implies -\frac{Sb^2}{2k_f} + C_2 = -\frac{Sb}{k_c}(b) + C_4$$

Substituting our known C_4 and solving for C_2 (which is our centerline temperature $T_1(0)$):

$$C_2 = -\frac{Sb^2}{k_c} + \left(T_a + \frac{Sb}{h} + \frac{Sb^2}{k_c} + \frac{Sb\delta}{k_c}\right) + \frac{Sb^2}{2k_f}$$

Simplifying this reveals a beautiful physical result. The centerline temperature is the ambient temperature plus the sum of the temperature drops across each thermal resistance (the fluid film, the cladding, and the fuel meat):

$$T_{center} = C_2 = T_a + \frac{Sb}{h} + \frac{Sb\delta}{k_c} + \frac{Sb^2}{2k_f}$$

*Note: If this fuel element had been a cylindrical rod rather than a flat plate, the symmetry condition at the center ($\frac{dT_1}{dr} = 0$) would be accompanied or replaced by a **boundedness condition**, requiring the temperature to remain finite at $r = 0$, which mathematically eliminates the singular $\ln(r)$ term that arises in the general solution for cylindrical coordinates.*

Boundary Conditions in PDEs: From Points to Surfaces

As we move from one-dimensional ODEs to multi-dimensional PDEs, the fundamental physical arguments that generate boundary conditions—specified values, fluxes, and interfaces—remain exactly the same. What changes is the geometry of the domain and the mathematical notation required to describe it.

In a one-dimensional ODE, the domain is a simple line segment (for example, $0 \leq x \leq L$), and the boundaries are discrete points ($x = 0$ and $x = L$).

In a PDE, our domain has two or more independent variables. If we are solving a 2D spatial problem over x and y, the domain is an area, and the boundaries are *curves* or *edges*. If we are solving a 3D problem, the domain is a volume, and the boundaries are *surfaces*. Therefore, a boundary condition in a PDE does not just apply to a single point; it must be satisfied everywhere along that bounding curve or surface. Consequently, a PDE boundary condition is often itself a function, not merely a number.

The Normal Derivative and Flux

In 1D, the diffusive flux of heat is simply $-k\frac{dT}{dx}$. If we want to set a flux boundary condition, we evaluate that single derivative.

In multiple dimensions, flux is a vector quantity. Fourier's Law becomes $\mathbf{q} = -k\nabla T$. When we specify a flux across a boundary surface, we only care about the transport strictly perpendicular to that surface. We define an outward-pointing unit normal vector, $\mathbf{n}$, at the surface. The flux across the boundary is the dot product of the flux vector and this normal vector:

$$q_{surface} = \mathbf{q} \cdot \mathbf{n} = -k\nabla T \cdot \mathbf{n}$$

It is important to look at the units here: the gradient ∇T carries the spatial units (e.g., K/m), while the unit normal $\mathbf{n}$ is a dimensionless directional pointer.

If your boundaries align with your coordinate system, this dot product simplifies back into a standard partial derivative. For example, on a surface located at $x = L$, the outward normal vector points in the positive x-direction ($\mathbf{n} = \mathbf{i}$). The outward flux is simply $-k\frac{\partial T}{\partial x}$.

However, if the outward normal points in the *negative* direction (such as the left boundary at $x = 0$, where $\mathbf{n} = -\mathbf{i}$), the flux leaving the domain is $+k\frac{\partial T}{\partial x}$. You must always pay strict attention to your coordinate directions when formulating PDE flux conditions!

Initial Conditions in PDEs

In transient PDEs, time (t) acts as an additional independent variable. Just as we have spatial boundaries that enclose our physical domain, time has a "boundary" at $t = 0$.

The initial condition is simply the specified value of the function at this temporal boundary. However, unlike an ODE where the initial condition is just the value at a point, a PDE initial condition defines the state of the system over the *entire spatial domain* at $t = 0$. For a 3D transient heat problem, you must specify $T(x, y, z, 0) = f(x, y, z)$.

Moving Boundary Problems

In transient PDE problems, we sometimes encounter situations where the spatial domain itself evolves over time. A classic fluid mechanics example is the deformation and breakup of a drop in a shear flow; in heat and mass transfer, it might be the advancing phase-change interface of a melting solid (a Stefan problem).

These are known as moving boundary (or free boundary) problems. They are significantly more difficult to formulate and solve because the location of the boundary is an unknown that must be determined as part of the solution. To accomplish this, you need additional boundary conditions that describe the physics of the moving surface itself. This typically involves a kinematic condition (dictating that the interface moves with the local fluid velocity) and a dynamic condition (describing the balance of forces, such as surface tension, pressure, and viscous stress, across the interface).

While the fundamental principle remains the same—translating physical balances into mathematical equations at a boundary—the analytical techniques required to solve moving boundary problems are generally beyond the scope of this primer.

Example 4: The 2D Rectangular Plate

To see how these concepts come together, let us set up the boundary conditions for a steady-state 2D heat conduction problem. Consider a rectangular plate of width W (in the x-direction) and height H (in the y-direction).

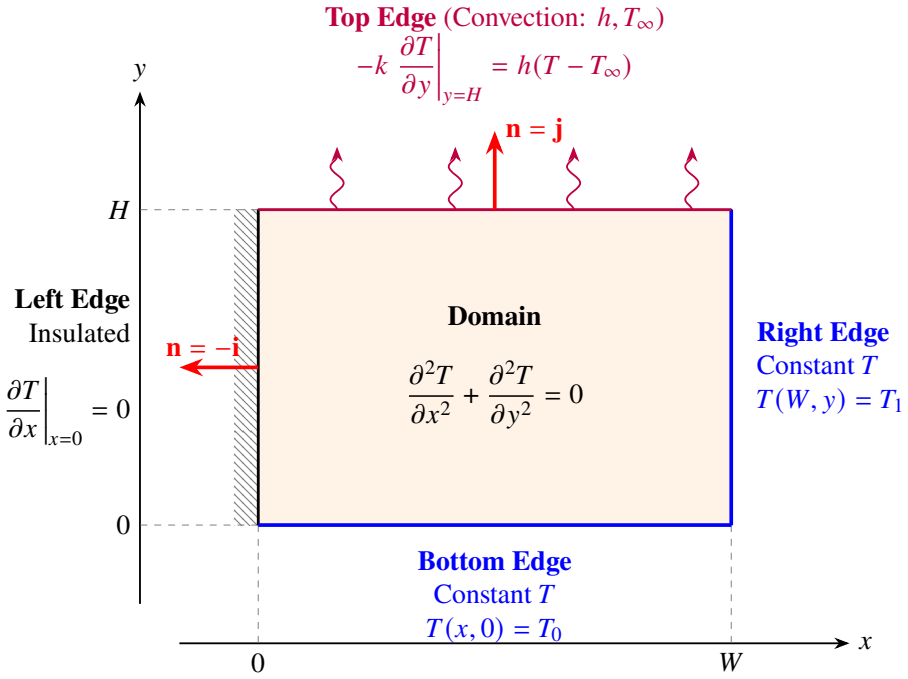


Figure 16.3: Boundary conditions for a 2D rectangular plate. Note how the outward unit normal vector **n** dictates the sign of the flux gradient on the left and top boundaries.

The governing PDE is Laplace's equation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

Because we have second derivatives in both x and y, we need two boundary conditions for the x-direction (at $x = 0$ and $x = W$) and two for the y-direction (at $y = 0$ and $y = H$). Let us apply a different physical constraint to each of the four edges:

1. **Left Edge ($x = 0$):** The edge is perfectly insulated (Neumann condition). The outward normal points in the negative x direction ($\mathbf{n} = -\mathbf{i}$), so the flux leaving the plate is $+k\dfrac{\partial T}{\partial x}$. Setting this to zero gives:

$$\left.\frac{\partial T}{\partial x}\right|_{x=0} = 0 \quad \text{for all } 0 \leq y \leq H$$

2. **Right Edge ($x = W$):** The edge is held at a constant temperature T_1 (Dirichlet condition).

$$T(W, y) = T_1 \quad \text{for all } 0 \leq y \leq H$$

3. **Bottom Edge ($y = 0$):** The edge is held at a constant temperature T_0 (Dirichlet condition).

$$T(x, 0) = T_0 \quad \text{for all } 0 \leq x \leq W$$

4. **Top Edge ($y = H$):** The edge is exposed to a cooling fluid at temperature T_∞ with a heat transfer coefficient h (Robin condition). The outward normal is in the positive y-direction ($\mathbf{n} = \mathbf{j}$), so the conductive flux reaching the surface is $-k\dfrac{\partial T}{\partial y}$. Equating this to the convective flux leaving the surface yields:

$$-k \left.\frac{\partial T}{\partial y}\right|_{y=H} = h\left(T(x, H) - T_\infty\right) \quad \text{for all } 0 \leq x \leq W$$

Notice how every single boundary condition is specified for an entire range of the orthogonal variable. *(Note: Because this problem features non-homogeneous conditions on three different boundaries, it does not have a simple, single-step analytical solution. As we will see in later chapters, solving this specific problem requires a technique called superposition, where we break it down into multiple simpler problems!)*

The true mathematical challenge in PDEs, which we will begin to tackle using Separation of Variables, is finding cohesive functions that satisfy the differential equation inside the domain while simultaneously satisfying these distinct boundary curves.

Practice Problems

Test your skills on these problems. Remember to always think physically about what the math is doing, and pay close attention to your coordinate system directions!

Problem 1: The Boundedness Trap (with Heat Generation)

Consider a solid spherical catalyst pellet of radius R and thermal conductivity k. An exothermic reaction occurs uniformly throughout the pellet, resulting in a constant volumetric heat generation rate S. The pellet is suspended in a fluid at ambient temperature T_∞ with a convective heat transfer coefficient h.

The governing ODE for the temperature $T(r)$ in spherical coordinates is:

$$\frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dT}{dr}\right) + \frac{S}{k} = 0$$

Integrating this twice yields the general solution:

$$T(r) = -\frac{S}{6k}r^2 - \frac{C_1}{r} + C_2$$

Apply the appropriate boundary conditions at the center of the pellet ($r = 0$) and at the surface ($r = R$) to find the specific constants C_1 and C_2.

Problem 2: Spot the Mistake (The Left Wall)

A student is analyzing a 1D heat conduction problem in a slab of thickness L (from $x = 0$ to $x = L$). The left wall ($x = 0$) is cooled by a fluid at ambient temperature T_a with a heat transfer coefficient h. The student writes the Robin boundary condition at $x = 0$ as:

$$-k\left.\frac{dT}{dx}\right|_{x=0} = h(T(0) - T_a)$$

This equation contains a classic, fatal sign error. Identify the error physically and write the correct boundary condition. *Hint: Think about the direction of the outward normal vector at $x = 0$, or just imagine the case where $T(0) > T_a$ and ask which way the heat must flow!*

Problem 3: The Catalytic Surface

A reactant diffuses through a stagnant liquid film of thickness δ (spanning from $y = 0$ to $y = \delta$). At the right boundary ($y = \delta$), the reactant hits a solid catalytic wall where it is instantly consumed by a first-order surface reaction: $Rate = k_s c_{surface}$. Write the boundary condition at $y = \delta$. (Assume the diffusion coefficient is D).

Problem 4: Integral Condition in a Pipe

You are given the general velocity profile for pressure-driven (Poiseuille) flow in a cylindrical pipe of radius R:

$$u_z(r) = u_{max}\left(1 - \frac{r^2}{R^2}\right)$$

However, you don't know u_{max}. You only know the total volumetric flow rate Q passing through the pipe. Set up and evaluate the integral condition to find u_{max} in terms of Q and R. *Hint: The differential area element for a circular cross-section is $dA = 2\pi r\, dr$.*

Problem 5: Periodic Temperature

Fluid flows through a long, corrugated channel where the geometry repeats exactly every L centimeters. Because the heating elements are also periodic, the temperature profile $T(x)$ reaches a "fully developed" periodic state. Write the two boundary conditions relating the temperature and the heat flux at the start of a repeating unit ($x = 0$) to the end of that unit ($x = L$).

Solutions to Practice Problems

Solution 1: The Boundedness Trap (with Heat Generation)

We have two boundaries: the center of the sphere ($r = 0$) and the surface ($r = R$).

Step 1: The Boundedness Condition Because the pellet is a solid sphere, the domain includes $r = 0$. If C_1 is anything other than zero, the $-C_1/r$ term will cause the temperature to approach infinity at the center. To maintain a physically finite temperature, we must apply a boundedness condition:

$$C_1 = 0$$

This reduces our temperature profile to $T(r) = -\frac{S}{6k}r^2 + C_2$.

Step 2: The Robin Condition At the surface ($r = R$), the conductive flux reaching the edge must equal the convective flux entering the fluid. The outward normal points in the $+r$ direction:

$$-k \left.\frac{dT}{dr}\right|_{r=R} = h(T(R) - T_\infty)$$

First, calculate the derivative of our reduced temperature profile:

$$\frac{dT}{dr} = -\frac{S}{3k}r \quad \Longrightarrow \quad \left.\frac{dT}{dr}\right|_{r=R} = -\frac{SR}{3k}$$

(Note: Multiplying this by $-k$ gives $+SR/3$, which is exactly the total heat generated divided by the surface area. The physics works!)

Now plug the derivative and $T(R)$ into the Robin condition:

$$-k\left(-\frac{SR}{3k}\right) = h\left(-\frac{SR^2}{6k} + C_2 - T_\infty\right)$$

Divide by h and isolate C_2:

$$\frac{SR}{3h} = -\frac{SR^2}{6k} + C_2 - T_\infty \quad \Longrightarrow \quad C_2 = T_\infty + \frac{SR}{3h} + \frac{SR^2}{6k}$$

(Optional: If you plug C_1 and C_2 back into the general solution, you get the classic parabolic profile: $T(r) = T_\infty + \frac{SR}{3h} + \frac{S}{6k}(R^2 - r^2)$.)

Solution 2: Spot the Mistake (The Left Wall)

The student blindly used the standard 1D flux formula ($-k\frac{dT}{dx}$) without accounting for the geometry.

Let's look at the physics. If the wall is hotter than the fluid ($T(0) > T_a$), heat must leave the slab and go into the fluid. That means heat is flowing to the *left* (in the negative x-direction). For heat to flow

to the left, the temperature must *increase* as you move to the right, meaning the slope $\frac{dT}{dx}$ must be positive.

If we look at the student's equation when $T(0) > T_a$, the right side is positive, which would force $\frac{dT}{dx}$ to be negative. This contradicts our physical reality!

Mathematically, the flux leaving a boundary is $-k\nabla T \cdot \mathbf{n}$. At the left boundary ($x = 0$), the outward normal points left ($\mathbf{n} = -\mathbf{i}$).

$$q_{out} = -k\left(\frac{dT}{dx}\mathbf{i}\right) \cdot (-\mathbf{i}) = +k\frac{dT}{dx}$$

Equating the outward conductive flux to the convective flux gives the correct boundary condition:

$$+k \left.\frac{dT}{dx}\right|_{x=0} = h(T(0) - T_a)$$

Solution 3: The Catalytic Surface

This is a Surface Sink (or Jump) condition. This condition arises from a surface sink and has the mathematical form of a Robin boundary condition. We must balance the mass diffusing *to* the wall with the mass being consumed *by* the wall. The diffusive flux reaching the wall at $y = \delta$ (where the normal vector points in the $+y$ direction) is given by Fick's Law:

$$Flux_{in} = -D \left.\frac{dc}{dy}\right|_{y=\delta}$$

This flux is entirely consumed by the first-order reaction:

$$-D \left.\frac{dc}{dy}\right|_{y=\delta} = k_s c(\delta)$$

Solution 4: Integral Condition in a Pipe

The total volumetric flow rate Q is the integral of the velocity profile over the entire cross-sectional area of the pipe.

$$Q = \int_{Area} u_z \, dA = \int_0^R u_{max}\left(1 - \frac{r^2}{R^2}\right)(2\pi r)\, dr$$

Pull out the constants and distribute the r:

$$Q = 2\pi u_{max} \int_0^R \left(r - \frac{r^3}{R^2}\right) dr$$

Integrate term by term:

$$Q = 2\pi u_{max}\left[\frac{r^2}{2} - \frac{r^4}{4R^2}\right]_0^R$$

Evaluate at the limits (R and 0):

$$Q = 2\pi u_{max}\left(\frac{R^2}{2} - \frac{R^4}{4R^2}\right) = 2\pi u_{max}\left(\frac{R^2}{4}\right) = u_{max}\frac{\pi R^2}{2}$$

Finally, solve for the unknown parameter u_{max}:

$$u_{max} = \frac{2Q}{\pi R^2}$$

Note: Since the average velocity is $u_{avg} = Q/(\pi R^2)$, this proves the classic fluid mechanics result that the maximum velocity in a laminar pipe flow is exactly twice the average velocity!

Solution 5: Periodic Temperature

For a periodic boundary condition, the state of the system at the end of the repeating unit ($x = L$) must be identical to the state at the beginning of the unit ($x = 0$). First, the temperatures must match (Dirichlet-style equivalence):

$$T(0) = T(L)$$

Second, the temperature gradients (and therefore the heat fluxes) must match (Neumann-style equivalence):

$$\left.\frac{dT}{dx}\right|_{x=0} = \left.\frac{dT}{dx}\right|_{x=L}$$

Note: These two conditions provide the two pieces of information needed to solve a second-order ODE over the domain $0 \leq x \leq L$, without needing to specify a fixed numerical value at either boundary.

17 PDE Simplification & Estimation via Scaling

A very useful technique for analyzing PDEs (and ODEs) is scaling analysis. This forms the basis of scale-up used in many engineering problems.

The core idea is that variables (such as position, velocity, time, etc.) all have units, but any physical relation should be able to be rendered **dimensionless**.

We scale all dependent (e.g., velocity, temperature, etc.) and independent (e.g., position, time) variables with parameters so that they are of order $O(1)$ in the region of interest.

To see how this works, look at a classic problem from heat transfer:

Example: Convective Heat Transfer from a Plate

Consider convective heat transfer from a plate at a constant temperature T_0. The fluid flows over it with a simple shear velocity profile: $u_x = \dot{\gamma} y$ and $u_y = 0$.

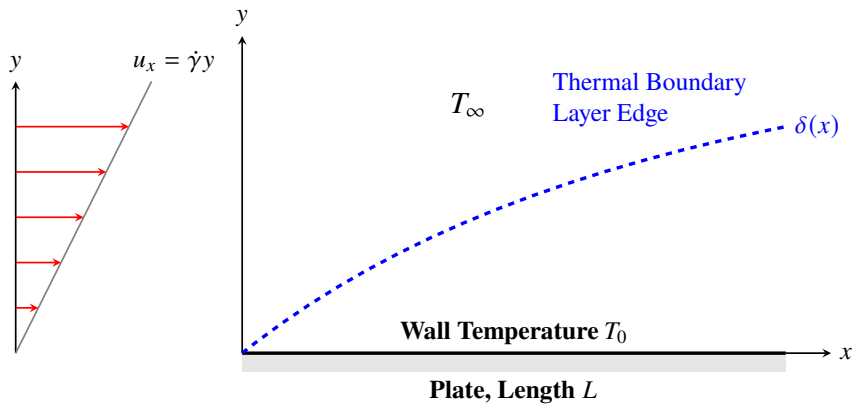


Figure 17.1: Convective heat transfer over a flat plate. A fluid with a simple shear velocity profile $u_x = \dot{\gamma} y$ and freestream temperature T_∞ flows over a plate of length L held at a constant temperature T_0. The thermal boundary layer $\delta(x)$ bounds the region where the temperature transitions from T_0 to T_∞.

We want to find the wall heat flux at $y = 0$:

$$q|_{y=0} = -k \left.\frac{\partial T}{\partial y}\right|_{y=0}$$

The total heat loss from a plate of length L and width W is:

$$\frac{Q}{W} = \int_0^L q|_{y=0}\, dx = \int_0^L -k \left.\frac{\partial T}{\partial y}\right|_{y=0} dx$$

Question: What is the rate of heat loss from the plate and how does it depend on parameters such as the shear rate $\dot{\gamma}$ and the material properties?

To solve this, we will scale the energy equation:

$$\rho \hat{C}_p \mathbf{u} \cdot \nabla T = k \nabla^2 T$$

In 2-D, this becomes:

$$\rho \hat{C}_p \left(u_x \frac{\partial T}{\partial x} + u_y \frac{\partial T}{\partial y} \right) = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

Substitute our simple shear velocity profile ($u_x = \dot{\gamma} y$, $u_y = 0$):

$$\rho \hat{C}_p \dot{\gamma} y \frac{\partial T}{\partial x} = k \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right)$$

Let's Scale!

From the geometry, we know x goes from 0 to L. So we scale x by L:

$$x^* = \frac{x}{L}$$

Our problem is linear in T, so we can subtract off a reference temperature T_∞ (the ambient temperature far away) and scale by an unknown characteristic temperature difference ΔT_c:

$$T^* = \frac{T - T_\infty}{\Delta T_c}$$

From our boundary condition at the wall ($y = 0$), we know $T|_{y=0} = T_0$.

$$T^*|_{y^*=0} = \frac{T_0 - T_\infty}{\Delta T_c}$$

For T^* to be order $O(1)$, we should set this equal to 1, which gives us our temperature scale:

$$\Delta T_c = T_0 - T_\infty \qquad \Longrightarrow \qquad T^*|_{y^*=0} = 1$$

Now for y. The boundary condition far away is $T|_{y\to\infty} = T_\infty$, so $T^*|_{y^*\to\infty} = 0$. We could use $y^* = \frac{y}{L}$, but that is not the best choice as that is not the length scale over which y is varying. Remember we want to scale y so that y^* is of $O(1)$ and the temperature changes over a much thinner region near the wall. Instead, we scale with an **unknown parameter** δ (the thermal boundary layer thickness):

$$y^* = \frac{y}{\delta}$$

The purpose of scaling is often to discover unknown length scales. If we already knew the thermal boundary layer thickness δ, then we could have used it in the scaling directly - and then perhaps determined the magnitude of fluid velocity necessary to achieve it. Generally, there is something that you don't know about the problem that you must determine via scaling.

Now we plug our dimensionless variables ($x = x^*L$, $y = y^*\delta$, $T = T^*\Delta T_c + T_\infty$) into the PDE:

$$\rho\hat{C}_p\dot{\gamma}(\delta y^*)\frac{\Delta T_c}{L}\frac{\partial T^*}{\partial x^*} = k\left(\frac{\Delta T_c}{L^2}\frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\Delta T_c}{\delta^2}\frac{\partial^2 T^*}{\partial y^{*2}}\right)$$

To finish, you must divide through by one of the groups of parameters. The key is to choose the group multiplying a term (a physical mechanism) you think is important so that it is scaled to be $O(1)$.

For this problem heat can only enter the fluid via conduction away from the plate in the y-direction so the diffusion in this direction *must* matter. We thus divide the entire equation by its scaling group, $\frac{k\Delta T_c}{\delta^2}$:

$$\left[\frac{\rho\hat{C}_p\dot{\gamma}\delta^3}{kL}\right] y^*\frac{\partial T^*}{\partial x^*} = \left[\frac{\delta^2}{L^2}\right]\frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}}$$

We determine δ by balancing y-conduction with another physical mechanism: x-direction convection. Thus, we set the coefficient on the left side equal to 1:

$$\frac{\rho\hat{C}_p\dot{\gamma}\delta^3}{kL} = 1 \qquad \Longrightarrow \qquad \delta = \left[\frac{kL}{\rho\hat{C}_p\dot{\gamma}}\right]^{1/3}$$

This tells us that the Boundary Layer Thickness (where T^* is going from 1 to 0) is of order $O(\delta)$, and it shows exactly how the thickness varies with our parameters!

What about the heat loss?

We had:

$$\frac{Q}{W} = \int_0^L -k\left.\frac{\partial T}{\partial y}\right|_{y=0} dx$$

Substituting our dimensionless variables:

$$\frac{Q}{W} = -\frac{k\Delta T_c L}{\delta}\int_0^1 \left.\frac{\partial T^*}{\partial y^*}\right|_{y^*=0} dx^*$$

If it is scaled right, the remaining integral is just a number of order $O(1)$! So just from scaling, we learn how Q/W depends on our parameters and its approximate magnitude:

$$\frac{Q}{W} \sim \frac{k\Delta T_c L}{\delta}$$

Simplifying the PDE

We need to plug our newly found δ back into the scaled PDE:

$$y^* \frac{\partial T^*}{\partial x^*} = \frac{\partial^2 T^*}{\partial y^{*2}} + \left[\frac{\delta^2}{L^2}\right] \frac{\partial^2 T^*}{\partial x^{*2}}$$

Let's look at that leftover coefficient on the x-diffusion term:

$$\frac{\delta^2}{L^2} = \frac{\left[\frac{kL}{\rho \hat{C}_p \dot{\gamma}}\right]^{2/3}}{L^2} = \left[\frac{k}{\rho \hat{C}_p \dot{\gamma} L^2}\right]^{2/3}$$

Provided that $\frac{\delta^2}{L^2} \ll 1$, which is equivalent to saying $L \gg \left(\frac{k}{\rho \hat{C}_p \dot{\gamma}}\right)^{1/2}$, diffusion in the x-direction is negligible. Thus, we get the simplified dimensionless equation:

$$y^* \frac{\partial T^*}{\partial x^*} = \frac{\partial^2 T^*}{\partial y^{*2}}$$

with boundary conditions $T^*|_{y^*=0} = 1$ and $T^*|_{y^* \to \infty} = 0$.

So, just from scaling we learn:

1. How Q/W depends on our parameters (and its approximate magnitude).
2. How the boundary layer thickness δ behaves.
3. How long the plate L has to be for our boundary layer approximation (neglecting x-diffusion) to be valid.
4. How to simplify our PDE!

This is most of what we want to know without even formally solving the equation! Scaling analysis does not merely simplify equations. It often discovers the characteristic length, time, velocity, or temperature scales of the problem.

Summary of Scaling Analysis Steps

1. Render all dependent and independent variables dimensionless with respect to parameters.
2. Often, scaling parameters are determined directly from the geometry or the boundary conditions.

3. If scalings aren't known (or obvious), use an "unknown characteristic value" (like δ).
4. Divide the entire equation by the group of parameters multiplying a term (a physical mechanism) that you are pretty sure is important.
5. Determine unknown scaling parameters by setting other important groups equal to 1.
6. Check to make sure the "leftovers" (the remaining dimensionless groups) really are small—otherwise, rescale!

Remember that setting a dimensionless coefficient in an equation equal to 1 is equivalent to requiring that the two physical mechanisms making up that ratio contribute at the same order of magnitude. If the leftover groups are $>> O(1)$ then you probably need to rescale (e.g., you have picked the wrong mechanism as being physically dominant). If they are of $O(1)$ then a full solution, sometimes numerical, is required as all mechanisms are of similar importance (e.g., you can't throw any of them out).

Exact solutions of PDEs are rare. Scaling analysis is often the first calculation performed by engineers because it identifies the dominant physics, estimates the answer, and frequently simplifies the governing equations before any detailed solution is attempted.

Practice Problems

Test your skills on these problems. Remember the core objective of scaling: we want to render the terms in our equations dimensionless such that the remaining derivatives are of order $O(1)$. This allows the coefficients to tell us the relative size of each physical mechanism without actually solving the PDE!

Problem 1: Thermal Protection for Spacecraft Reentry

Both the legacy Space Shuttle and the modern Starship rely on silica ceramic tiles to protect the vessel from the extreme heat of atmospheric reentry. Reentry is a transient process that lasts approximately 20 minutes ($t_c \approx 1200$ s). The tiles must be thick enough that the heat pulse cannot penetrate to the more delicate hull during this time. The 1-D transient energy equation is:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial y^2}$$

where α is the thermal diffusivity of the silica tiles (for shuttle tiles, $\alpha \approx 10^{-6}$ m^2/s).
Scale time using the reentry time ($t^* = t/t_c$) and position using an unknown tile thickness δ ($y^* = y/\delta$). Balance the equation to find a formula for the required tile thickness δ. Plug in the numbers to estimate how thick the tiles on the Space Shuttle needed to be!

Problem 2: Finding an Unknown Length Scale (A Falling Film)

Consider a layer of paint flowing steadily down a vertical wall due to gravity. Instead of knowing the film thickness h, suppose you know the volumetric flow rate of paint per unit width of the wall,

$\frac{Q}{W}$. The downward velocity profile $u_y(x)$ is governed by the balance of viscosity and gravity:

$$\mu \frac{d^2 u_y}{dx^2} + \rho g = 0$$

Scale the x-coordinate by the unknown film thickness h, and the velocity by an unknown characteristic velocity U.

(a) Balance the ODE to find how the velocity scale U depends on h.

(b) We know the flow rate per unit width scales as $\frac{Q}{W} \sim Uh$. Substitute your result from (a) into this expression to find how the paint film thickness h scales with the flow rate $\frac{Q}{W}$ and the fluid properties.

Problem 3: Radiator Fin in Space

On a spacecraft, heat cannot be removed by convection; it must be radiated into the vacuum of space. To maximize surface area, spacecraft use thin radiator fins. Consider a 2-D fin of thickness $2h$ and length L. The base of the fin is at a hot temperature T_b. The 2-D steady heat equation inside the fin is:

$$k \frac{\partial^2 T}{\partial x^2} + k \frac{\partial^2 T}{\partial y^2} = 0$$

The fin radiates from its top surface ($y = h$) to deep space (0^oK). According to the Stefan-Boltzmann law, the boundary condition at the surface is $-k \frac{\partial T}{\partial y} = \epsilon \sigma T^4$.

Assume the centerline ($y = 0$) is a line of symmetry ($\frac{\partial T}{\partial y} = 0$).

Integrate the 2-D PDE from $y = 0$ to $y = h$ to turn it into a 1-D ODE in the x-direction (appropriate if $h << L$). Then, scale the temperature by T_b and the x-coordinate by the unknown fin length L. Balance the resulting equation to estimate how long the fin should be!

Solutions to Practice Problems

Solution 1: Thermal Protection for Spacecraft Reentry

We define our dimensionless variables as:

$$T^* = \frac{T - T_{hull}}{T_{surface} - T_{hull}} \implies dT = \Delta T_c \, dT^*, \quad t = t_c t^*, \quad y = \delta y^*$$

Substitute these into the transient heat equation:

$$\frac{\Delta T_c}{t_c}\frac{\partial T^*}{\partial t^*} = \alpha \frac{\Delta T_c}{\delta^2}\frac{\partial^2 T^*}{\partial y^{*2}}$$

Cancel the ΔT_c and divide the entire equation by the conduction coefficient, $\left[\frac{\alpha}{\delta^2}\right]$:

$$\left[\frac{\delta^2}{\alpha t_c}\right]\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial y^{*2}}$$

To balance the transient accumulation of heat with the conduction of heat, we set the coefficient on the left equal to 1:

$$\frac{\delta^2}{\alpha t_c} = 1 \qquad \implies \qquad \delta = \sqrt{\alpha t_c}$$

Now we plug in the real values for the Space Shuttle:

$$\delta = \sqrt{(10^{-6}\ \text{m}^2/\text{s})(1200\ \text{s})} = \sqrt{0.0012\ \text{m}^2} \approx 0.034 \text{ meters}$$

The characteristic penetration depth is 3.4 cm. Actual Space Shuttle thermal tiles ranged from 2.5 cm to 12.5 cm (1 to 5 inches) thick depending on their location on the orbiter. Our simple scaling perfectly predicted the engineering reality!

Solution 2: Finding an Unknown Length Scale (A Falling Film)

(a) Define our dimensionless variables: $x = hx^*$, and $u_y = Uu^*$. Substitute into the ODE:

$$\mu\frac{d^2(Uu^*)}{d(hx^*)^2} + \rho g = 0 \qquad \implies \qquad \left[\frac{\mu U}{h^2}\right]\frac{d^2u^*}{dx^{*2}} + \rho g = 0$$

Divide by the viscous group to make the derivative $O(1)$:

$$\frac{d^2u^*}{dx^{*2}} + \left[\frac{\rho g h^2}{\mu U}\right] = 0$$

Set the dimensionless group to 1 to balance gravity and viscosity:

$$\frac{\rho g h^2}{\mu U} = 1 \qquad \implies \qquad U \sim \frac{\rho g h^2}{\mu}$$

(b) We are given that $\frac{Q}{W} \sim Uh$. Substitute our velocity scale U into this relation:

$$\frac{Q}{W} \sim \left(\frac{\rho g h^2}{\mu}\right) h = \frac{\rho g h^3}{\mu}$$

Now, solve this relation for the unknown film thickness h:

$$h \sim \left(\frac{\mu}{\rho g}\frac{Q}{W}\right)^{1/3}$$

Solution 3: Radiator Fin in Space

First, we simplify the 2-D PDE into a 1-D ODE by integrating across the fin thickness (from $y = 0$ to $y = h$):

$$\int_0^h k\frac{\partial^2 T}{\partial x^2}dy + \int_0^h k\frac{\partial^2 T}{\partial y^2}dy = 0$$

Assuming the x-derivative doesn't vary strongly across the very thin fin (appropriate if $h << L$), the first term becomes $h \cdot k\frac{d^2T}{dx^2}$. The second term evaluates exactly:

$$kh\frac{d^2T}{dx^2} + \left[k\frac{\partial T}{\partial y}\right]_{y=h} - \left[k\frac{\partial T}{\partial y}\right]_{y=0} = 0$$

Substitute the surface boundary conditions. The centerline ($y = 0$) is a line of symmetry, so the flux is zero. The top ($y = h$) radiates to space, so $k\frac{\partial T}{\partial y} = -\epsilon\sigma T^4$.

$$kh\frac{d^2T}{dx^2} - \epsilon\sigma T^4 = 0$$

Now we scale this 1-D equation! Let $x = Lx^*$ and $T = T_b T^*$:

$$\left[\frac{khT_b}{L^2}\right]\frac{d^2T^*}{dx^{*2}} - \left[\epsilon\sigma T_b^4\right](T^*)^4 = 0$$

Divide the equation by the conduction scaling group to make the derivative $O(1)$:

$$\frac{d^2T^*}{dx^{*2}} - \left[\frac{\epsilon\sigma T_b^3 L^2}{kh}\right](T^*)^4 = 0$$

To balance the heat conducted down the fin with the heat radiated away into space, set the remaining dimensionless group equal to 1:

$$\frac{\epsilon\sigma T_b^3 L^2}{kh} = 1 \quad \Longrightarrow \quad L = \sqrt{\frac{kh}{\epsilon\sigma T_b^3}}$$

18 Solving PDEs by Simplification

PDEs are far more difficult to solve than ODEs. Thus, the most common method is to turn them into ODEs!

Simplification by Change of Variables

One method is via simplification or change of variables. Suppose a long vertical tube of radius a is filled with a viscous liquid. Under steady conditions, gravity drives a fully developed downward flow. The velocity is governed by the PDE:

$$\frac{\partial^2 u_z}{\partial x^2} + \frac{\partial^2 u_z}{\partial y^2} = -\frac{\rho g}{\mu}$$

with the boundary condition that $u_z = 0$ on the tube walls, which are at $x^2 + y^2 = a^2$.

We can simplify this via a change of variables:

$$r = (x^2 + y^2)^{\frac{1}{2}}, \quad x = r\cos\theta, \quad y = r\sin\theta$$

The conversion from Cartesian coordinates to cylindrical coordinates yields:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u_z}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 u_z}{\partial \theta^2} = -\frac{\rho g}{\mu}$$

with the boundary conditions that u_z is finite at $r = 0$, and $u_z|_{r=a} = 0$.

Because the geometry, forcing, and boundary conditions are axisymmetric, the solution cannot depend on θ. Therefore, $\partial u_z/\partial\theta = 0$. Thus, we have the ODE:

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{du_z}{dr}\right) = -\frac{\rho g}{\mu}$$

In general, you don't want to break up the operator. Here we just multiply by r and integrate:

$$r\frac{du_z}{dr} = -\frac{1}{2}r^2\frac{\rho g}{\mu} + C_1$$

Divide by r:

$$\frac{du_z}{dr} = -\frac{1}{2}r\frac{\rho g}{\mu} + \frac{C_1}{r}$$

Because the velocity (and its derivative) must be finite at the centerline $r = 0$, the $\ln(r)$ term that would result from integrating $1/r$ must vanish. Therefore, $C_1 = 0$. Note that for an annular geometry (e.g., an artery with a catheter in the center) this term is preserved due to the different boundary conditions.

We integrate again:

$$u_z = -\frac{1}{4}r^2\frac{\rho g}{\mu} + C_2$$

Applying the boundary condition at $r = a$ ($u_z = 0$):

$$0 = -\frac{1}{4}a^2\frac{\rho g}{\mu} + C_2 \quad \Longrightarrow \quad C_2 = \frac{1}{4}a^2\frac{\rho g}{\mu}$$

This gives our final velocity profile:

$$u_z = \frac{\rho g a^2}{4\mu}\left(1 - \frac{r^2}{a^2}\right)$$

which is much easier than solving it in Cartesian coordinates!

Rule 1: In general, always pick a coordinate system in which the boundary has a convenient representation!

Allowing Boundary Conditions to Suggest the Form of the Solution

In addition to choosing the best coordinate system, you often can simplify a problem by looking at the boundary conditions and allowing them to suggest the form of the solution.

Example: 2-D Heat Transfer from a Circular Dipole

Consider a 2-D domain with a circular inclusion of radius a. The boundary condition far away is $T|_{r\to\infty} = 0$. On the boundary of the inclusion, $T|_{r=a} = \lambda x$ (hot on one side, cold on the other).

The governing equation is Laplace's Equation:

$$\nabla^2 T = 0 \quad \Longrightarrow \quad \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

First, since the boundary is a circle (cylinder), use cylindrical coordinates!

$$x = r\cos\theta$$
$$y = r\sin\theta$$

In 2-D, the Laplacian becomes:

$$\nabla^2 T = \frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right) + \frac{1}{r^2}\frac{\partial^2 T}{\partial \theta^2} = 0$$

And the boundary conditions become:

$$T|_{r\to\infty} = 0, \quad T|_{r=a} = \lambda a \cos\theta$$

since $\lambda x = \lambda r cos(\theta)$ is evaluated at $r = a$.

This time the problem is a function of both r and θ. But the ∇^2 operator only has independent derivatives with respect to r or θ. Thus, we allow the boundary condition to suggest the form of the solution! Let:

$$T(r,\theta) = \lambda \cos\theta \cdot f(r)$$

Our boundary conditions for $f(r)$ are now:

$$f|_{r=a} = a \quad \text{and} \quad f|_{r\to\infty} = 0$$

Substituting our guessed form into the PDE:

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{df}{dr}\right)\cos\theta - \frac{f}{r^2}\cos\theta = 0$$

Factoring out the $\cos\theta$, we get:

$$\frac{1}{r}\frac{d}{dr}\left(r\frac{df}{dr}\right) - \frac{f}{r^2} = 0$$

Which is not only an ODE, it's an Euler equation. Expanding the derivative:

$$r^2 f'' + r f' - f = 0$$

Let $f(r) = r^n$:

$$n(n-1)r^n + nr^n - r^n = 0 \quad \Longrightarrow \quad n^2 - 1 = 0 \quad \Longrightarrow \quad n = \pm 1$$

So the general solution is $f(r) = C_1 r + \dfrac{C_2}{r}$.

Applying the boundary condition at infinity, $f(\infty) = 0$, dictates that $C_1 = 0$.

Applying the boundary condition at the inclusion, $f(a) = a$:

$$\frac{C_2}{a} = a \quad \Longrightarrow \quad C_2 = a^2$$

So $f(r) = \dfrac{a^2}{r}$, and our final temperature profile is:

$$T(r,\theta) = \frac{\lambda a^2}{r}\cos\theta$$

Rule 2: Allow the boundary conditions to suggest the form of the solution!

Simplification via Analytic Continuation

Another important way to simplify PDEs arises for linear problems which are driven by periodic forcing. Examples range from the pulsatile flow of your heart to the periodic heating of the earth over the days and seasons!

For a linear problem, we can solve these via analytic continuation into the complex plane. The key idea is Euler's formula:

$$e^{i\omega t} = \cos(\omega t) + i\sin(\omega t)$$

So if $e^{i\omega t}$ is your driving force, then the real part of the solution is that driven by $\cos(\omega t)$ and the imaginary part is that driven by $\sin(\omega t)$!

Why does this help? Two ways:

1. For a linear problem, the solution will also be periodic.
2. Usually, you have first derivatives with respect to t, so we have $\frac{\partial}{\partial t}(e^{i\omega t}) = i\omega e^{i\omega t}$, and you get $e^{i\omega t}$ back again!

An Example: Periodic Heating

We have the 1-D unsteady heat equation:

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial y^{*2}}$$

with boundary conditions $T^*|_{y^*=0} = \cos(t^*)$ and $T^*|_{y^*\to\infty} = 0$.

Let's solve via analytic continuation. Let $T^* = \mathrm{Re}\{\hat{T}\}$. We expect $\hat{T}$ to be periodic:

$$\hat{T} = e^{it^*} f(y^*)$$

The superscript * denotes dimensionless variables, while the hat denotes the complex-valued auxiliary solution.

Substitute this into the PDE:

$$\frac{\partial \hat{T}}{\partial t^*} = ie^{it^*} f(y^*) = \frac{\partial^2 \hat{T}}{\partial y^{*2}} = e^{it^*} f''(y^*)$$

Dividing out the exponential gives:

$$f'' = if$$

subject to $f(0) = 1$ and $f(\infty) = 0$.

This is a constant coefficient ODE. The characteristic roots are $\pm\sqrt{i}$. We want the decaying part (due to the boundary condition at infinity), so we take the negative root:

$$f(y^*) = Ae^{-\sqrt{i}y^*}$$

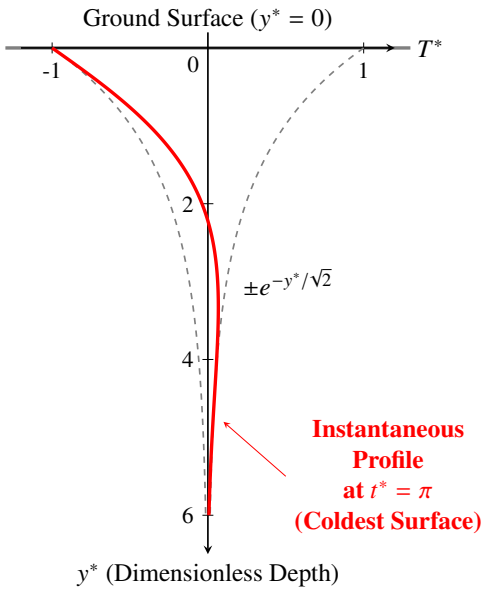


Figure 18.1: The dimensionless temperature distribution T^* as a function of depth y^* for the periodic heating problem. The solid red line represents a snapshot in time at $t^* = \pi$, corresponding to the exact moment of the coldest surface temperature ($T^*|_{y^*=0} = -1$). The dashed gray lines indicate the exponential decay envelope. Notice how the temperature oscillations are damped quickly with depth and exhibit a significant phase shift relative to the surface forcing.

With $f(0) = 1$, we find $A = 1$. Thus:

$$\hat{T} = e^{it^*} e^{-\sqrt{i}y^*} = e^{it^* - \sqrt{i}y^*}$$

Now, note that $\sqrt{i} = \frac{1+i}{\sqrt{2}}$. We choose the branch with positive real part so that the exponential decays as $y^* \to \infty$. Thus:

$$\hat{T} = e^{it^* - \left(\frac{1+i}{\sqrt{2}}\right)y^*} = e^{-\frac{y^*}{\sqrt{2}}} e^{i\left(t^* - \frac{y^*}{\sqrt{2}}\right)}$$

Since $T^* = \mathrm{Re}\{\hat{T}\}$, we take the real part of this expression (using Euler's formula on the complex exponential):

$$T^* = e^{-\frac{y^*}{\sqrt{2}}} \cos\left(t^* - \frac{y^*}{\sqrt{2}}\right)$$

So our temperature profile is just an exponentially decaying wave with a phase shift. This expression actually finds use in determining how deeply you have to bury pipes to avoid freezing in the winter - and even when the pipes are most likely to freeze if not buried deeply enough. Because of the phase shift, it is actually not on the coldest day of the year!

Note that if you had a bounded domain (instead of $y \to \infty$) you would use hyperbolics rather than exponentials in y. This yields a result containing functions of complex arguments, but MATLAB will plot up the real or imaginary part of a complex solution just fine!

Practice Problems

Test your skills on these problems. The goal here is not to do pages of brute-force math, but to use clever simplifications to turn scary PDEs into manageable ODEs!

Problem 1: Simplification by Change of Variables

Consider the steady-state diffusion of a drug from a spherical pill of radius R into an infinite fluid medium. The concentration C far away is 0, and at the pill surface is C_0. The Cartesian PDE for 3-D steady diffusion is:

$$\frac{\partial^2 C}{\partial x^2} + \frac{\partial^2 C}{\partial y^2} + \frac{\partial^2 C}{\partial z^2} = 0$$

Assume the concentration field is spherically symmetric (i.e., C only depends on the radial distance r). Convert this PDE into spherical coordinates using the Laplacian operators from the previous chapter, and solve for the concentration profile $C(r)$.

Problem 2: Boundary Conditions Suggesting the Form

Consider steady 2-D heat conduction in a semi-infinite rectangular plate located at $0 \le x \le L$ and $0 \le y < \infty$. The side boundaries at $x = 0$ and $x = L$, as well as the top boundary at $y \to \infty$, are kept at 0°C. The bottom edge at $y = 0$ is heated with a sinusoidal profile:

$$T(x, 0) = T_0 \sin\left(\frac{\pi x}{L}\right)$$

The governing PDE is Laplace's equation: $\dfrac{\partial^2 T}{\partial x^2} + \dfrac{\partial^2 T}{\partial y^2} = 0$.

Use the boundary condition at $y = 0$ to suggest a form for the solution: $T(x, y) = \sin\left(\frac{\pi x}{L}\right) f(y)$. Find the ODE for $f(y)$ and solve for the final temperature profile.

Problem 3: Analytic Continuation (Stokes' Second Problem)

An infinite flat plate is located at $y = 0$, with a semi-infinite body of viscous fluid above it ($y > 0$). The plate oscillates back and forth in the x-direction with velocity $u(0, t) = U_0 \sin(\omega t)$. The fluid velocity $u(y, t)$ is governed by:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

where ν is the kinematic viscosity. The fluid is stationary far away: $u(\infty, t) = 0$.

Use analytic continuation to find the steady-periodic velocity profile $u(y, t)$.

(Hint: Since the driving force is a sine wave, you will want the imaginary part of your complex solution! Let $\hat{u}(y, t) = U_0 e^{i\omega t} f(y)$.)

Solutions to Practice Problems

Solution 1: Simplification by Change of Variables

Because the problem is spherically symmetric, the derivatives with respect to θ and ϕ are zero. Using the spherical Laplacian from Section 12, the PDE becomes:

$$\nabla^2 C = \frac{1}{r^2}\frac{d}{dr}\left(r^2\frac{dC}{dr}\right) = 0$$

Multiply both sides by r^2 and integrate once:

$$r^2\frac{dC}{dr} = C_1 \qquad \Longrightarrow \qquad \frac{dC}{dr} = \frac{C_1}{r^2}$$

Integrate a second time:

$$C(r) = -\frac{C_1}{r} + C_2$$

Now we apply the boundary conditions. Far away ($r \to \infty$), the concentration is zero:

$$C(\infty) = -\frac{C_1}{\infty} + C_2 = 0 \qquad \Longrightarrow \qquad C_2 = 0$$

At the pill surface ($r = R$), the concentration is C_0:

$$C(R) = -\frac{C_1}{R} = C_0 \qquad \Longrightarrow \qquad C_1 = -C_0 R$$

Substitute C_1 and C_2 back into the general solution to get the final profile:

$$C(r) = C_0\frac{R}{r}$$

Solution 2: Boundary Conditions Suggesting the Form

We guess the solution takes the form $T(x, y) = \sin\left(\frac{\pi x}{L}\right) f(y)$. We substitute this into the PDE:

$$\frac{\partial^2}{\partial x^2}\left[\sin\left(\frac{\pi x}{L}\right) f(y)\right] + \frac{\partial^2}{\partial y^2}\left[\sin\left(\frac{\pi x}{L}\right) f(y)\right] = 0$$

Evaluating the derivatives:

$$-\left(\frac{\pi}{L}\right)^2 \sin\left(\frac{\pi x}{L}\right) f(y) + \sin\left(\frac{\pi x}{L}\right) f''(y) = 0$$

Divide out the common sine term to yield a simple, constant-coefficient ODE for $f(y)$:

$$f''(y) - \left(\frac{\pi}{L}\right)^2 f(y) = 0$$

The roots of the characteristic equation are $\pm\pi/L$, so the general solution is:

$$f(y) = C_1 e^{\frac{\pi}{L}y} + C_2 e^{-\frac{\pi}{L}y}$$

Since the temperature must go to zero as $y \to \infty$, the positive exponential term must vanish, so $C_1 = 0$.

To find C_2, we use the boundary condition at $y = 0$:

$$T(x,0) = \sin\left(\frac{\pi x}{L}\right) f(0) = T_0 \sin\left(\frac{\pi x}{L}\right) \quad \Longrightarrow \quad f(0) = T_0$$

Since $f(0) = C_2 e^0 = C_2$, we find $C_2 = T_0$. Thus, $f(y) = T_0 e^{-\frac{\pi}{L}y}$.
Our final temperature profile is:

$$T(x,y) = T_0 e^{-\frac{\pi y}{L}} \sin\left(\frac{\pi x}{L}\right)$$

Solution 3: Analytic Continuation (Stokes' Second Problem)

We solve the complex problem $\dfrac{\partial \hat{u}}{\partial t} = \nu \dfrac{\partial^2 \hat{u}}{\partial y^2}$ and guess the form $\hat{u}(y,t) = U_0 e^{i\omega t} f(y)$. Substituting this into the PDE:

$$i\omega U_0 e^{i\omega t} f(y) = \nu U_0 e^{i\omega t} f''(y)$$

Dividing out the $U_0 e^{i\omega t}$ terms gives:

$$i\omega f = \nu f'' \quad \Longrightarrow \quad f'' - \frac{i\omega}{\nu} f = 0$$

The roots of this ODE are $r = \pm\sqrt{\frac{i\omega}{\nu}}$. Using the identity $\sqrt{i} = \frac{1+i}{\sqrt{2}}$, we get:

$$r = \pm\sqrt{\frac{\omega}{2\nu}}(1+i)$$

Let's define a constant $\beta = \sqrt{\frac{\omega}{2\nu}}$ to keep the math clean. Our general solution is:

$$f(y) = Ae^{-\beta(1+i)y} + Be^{\beta(1+i)y}$$

The fluid is stationary at $y \to \infty$, meaning we must reject the growing exponential, so $B = 0$.
At the wall ($y = 0$), the complex velocity is $\hat{u}(0,t) = U_0 e^{i\omega t}$. This requires $f(0) = 1$, which gives $A = 1$.

Putting it all together:

$$\hat{u}(y,t) = U_0 e^{i\omega t} e^{-\beta(1+i)y}$$
$$\hat{u}(y,t) = U_0 e^{-\beta y} e^{i(\omega t - \beta y)}$$

Using Euler's formula to expand the complex exponential:

$$\hat{u}(y,t) = U_0 e^{-\beta y}\left[\cos(\omega t - \beta y) + i\sin(\omega t - \beta y)\right]$$

Because our original physical driving force was a sine wave ($U_0 \sin(\omega t)$), our actual physical solution is the **imaginary part** of $\hat{u}$:

$$u(y,t) = U_0 e^{-y\sqrt{\frac{\omega}{2\nu}}} \sin\left(\omega t - y\sqrt{\frac{\omega}{2\nu}}\right)$$

(Notice the similarity to the periodic heating problem: a decaying amplitude with a phase shift!)

19 Solving PDEs via Self-Similarity

A very important (if special) class of problems are "length or time-scale deficient"—this includes problems such as many Boundary Layer (BL) problems, unsteady diffusion from a point source, etc.

Such problems often admit **self-similar solutions** where a transformed dependent variable is a function of a reduced set of independent variables. For example, you can turn a 2-D PDE into an ODE—a huge simplification!

This is codified by **Morgan's Theorem**:

1. If a well-posed problem is invariant to a one-parameter group of continuous transformations, the number of independent variables may be reduced by one.
2. The reduction is accomplished by choosing as new independent and dependent variables combinations that are invariant under the transformation.

Ok, what does all this mean?

- **Well-posed problem:** Consists of the PDE, independent and dependent variables, boundary conditions (BCs), location of BCs, etc.
- **Invariant:** Does not change (identical).
- **One parameter:** At least one parameter of the group of transformations is undetermined, but the problem is still invariant.
- **Continuous transformation:** Can include shifting or scaling, but doesn't include non-continuous transformations (e.g., rotation through a fixed angle).
- **Reduction:** New dependent and independent variables are combinations of the old ones that are invariant under the transformation.

Affine Stretching

There are many ways to apply this, but by far the simplest is **Affine Stretching**: Just stretch all dependent and independent variables by an arbitrary factor. If some combination of these parameters exists that leaves the problem invariant but not all the parameters are forced to be 1 (e.g., "one free wish" is left over), you win! Mathematically, this corresponds to the number of stretching parameters exceeding the number of independent restrictions by one. We will refer to each unrestricted parameter as a "free wish" because it can be chosen arbitrarily without changing

the problem. Note that if you have no free wishes left over, you lose - no self-similarity is revealed by the stretching. If you have *two* free wishes (which sometimes happens) you *may* get a double reduction (a really big win!) but you have to do it in two steps: eliminate one independent variable, stretch the transformed problem, and then see if you really do get a second "free wish" left over.

Let's use the heated plate example again, but with a constant heat flux boundary condition:

$$\left.\frac{\partial T^*}{\partial y^*}\right|_{y^*=0} = -1 \quad \text{(constant heat flux)}$$

So our governing equation and boundary conditions are:

$$y^* \frac{\partial T^*}{\partial x^*} = \frac{\partial^2 T^*}{\partial y^{*2}}$$

with $T^*|_{x^*=0} = 0$, $\quad T^*|_{y^*\to\infty} = 0$, $\quad$ and $\quad \left.\dfrac{\partial T^*}{\partial y^*}\right|_{y^*=0} = -1.$

Let's stretch the dependent and independent variables:

$$T^* = A\overline{T}, \quad x^* = B\overline{x}, \quad y^* = C\overline{y}$$

Substituting this into the differential equation yields:

$$\frac{CA}{B}\overline{y}\frac{\partial \overline{T}}{\partial \overline{x}} = \frac{A}{C^2}\frac{\partial^2 \overline{T}}{\partial \overline{y}^2}$$

You want the minimum number of restrictions, so you divide out by A/C^2:

$$\frac{C^3}{B}\overline{y}\frac{\partial \overline{T}}{\partial \overline{x}} = \frac{\partial^2 \overline{T}}{\partial \overline{y}^2}$$

So the DE is invariant if:

$$\frac{C^3}{B} = 1$$

We must do the boundary conditions too! In general, homogeneous (e.g., = 0) BCs don't add restrictions as the stretching parameters just divide out!

$$T^*|_{y^*\to\infty} = 0 \quad \Longrightarrow \quad A\overline{T}|_{C\overline{y}\to\infty} = 0$$

or $\overline{T}|_{\overline{y}\to\infty} = 0$ (no restrictions).

Note that a BC at $y^* = L$ would yield a restriction $C = 1$ and prevent self-similarity—but ∞ behaves the same way as a homogeneous boundary condition location since it is invariant to division by any parameter.

And we have the flux boundary condition at the wall:

$$\left.\frac{\partial T^*}{\partial y^*}\right|_{y^*=0} = -1 \quad \Longrightarrow \quad \left.\frac{A}{C}\frac{\partial \overline{T}}{\partial \overline{y}}\right|_{\overline{y}=0} = -1$$

So this is invariant if:

$$\frac{A}{C} = 1$$

But we're done! We have 3 parameters (A, B, C) and 2 restrictions, leaving 1 free wish! Any combination of T^*, x^*, y^* which are invariant under these transformations will work.

Finding the Similarity Variable

We have:

$$\frac{A}{C} = 1 \quad \text{and} \quad \frac{C^3}{B} = 1$$

You could try setting $\dfrac{T^*}{y^*} = f\left(\dfrac{y^{*3}}{x^*}\right)$ to yield an ODE, but this is a **terrible** choice!

Instead, use the **Canonical Form**: Put all the complexity in the "time or time-like" variable. Alternatively, put complexity in the variable whose highest derivative in the DE is the lowest (this yields the same result). Note that while by Morgan's Theorem any combination that satisfies invariance (such as the bad choice above) will work, ones done via canonical form nearly always work much better!

For this problem, we have a first derivative with respect to x^* and a second derivative with respect to y^*. Thus, put the complexity in x^*!

Since $x^* = B\overline{x}$, put the complexity in B.

We had: $\dfrac{A}{C} = 1$ and $\dfrac{C^3}{B} = 1$. Solving for A and C in terms of B:

$$C = B^{1/3} \quad \text{and} \quad A = B^{1/3}$$

So the invariant combinations are:

$$\frac{C}{B^{1/3}} = 1 \quad \text{and} \quad \frac{A}{B^{1/3}} = 1$$

This dictates our transformation. We define a similarity variable η and a similarity rule $f(\eta)$:

$$\eta = \frac{y^*}{x^{*1/3}} \quad \text{(similarity variable)}$$

$$\frac{T^*}{x^{*1/3}} = f(\eta) \quad \Longrightarrow \quad T^* = x^{*1/3} f(\eta) \quad \text{(similarity rule)}$$

Notice that the similarity variable contains the same $x^{1/3}$ dependence that emerged from the scaling analysis in the previous chapter. This is not a coincidence. Scaling often provides strong clues about the form of a similarity transformation, and in fact the mechanical processes of scaling and affine stretching are identical!

Proving it Works (Converting the PDE to an ODE)

Let's substitute this back into our original PDE to prove it works. First, take the y^*-derivatives:

$$\frac{\partial T^*}{\partial y^*} = \frac{\partial}{\partial y^*}\left(x^{*1/3} f(\eta)\right) = x^{*1/3}\frac{\partial f}{\partial y^*}$$

By the chain rule, $\frac{\partial f}{\partial y^*} = f'(\eta)\frac{\partial \eta}{\partial y^*}$. We know:

$$\frac{\partial \eta}{\partial y^*} = \frac{\partial}{\partial y^*}\left(\frac{y^*}{x^{*1/3}}\right) = \frac{1}{x^{*1/3}}$$

So:

$$\frac{\partial T^*}{\partial y^*} = x^{*1/3} f'(\eta)\left(\frac{1}{x^{*1/3}}\right) = f'(\eta)$$

This makes evaluating the BC at $y^* = 0$ very easy:

$$\left.\frac{\partial T^*}{\partial y^*}\right|_{y^*=0} = f'(0) = -1 \qquad \Longrightarrow \qquad f'(0) = -1$$

We also need the second derivative with respect to y^*:

$$\frac{\partial^2 T^*}{\partial y^{*2}} = \frac{\partial}{\partial y^*}\left(\frac{\partial T^*}{\partial y^*}\right) = \frac{\partial f'}{\partial y^*} = f''(\eta)\frac{\partial \eta}{\partial y^*} = x^{*-1/3} f''(\eta)$$

And finally, the x^*-derivative:

$$\frac{\partial T^*}{\partial x^*} = \frac{\partial}{\partial x^*}\left(x^{*1/3} f(\eta)\right) = \frac{1}{3}x^{*-2/3} f(\eta) + x^{*1/3} f'(\eta)\frac{\partial \eta}{\partial x^*}$$

Now, evaluate $\frac{\partial \eta}{\partial x^*}$:

$$\frac{\partial \eta}{\partial x^*} = \frac{\partial}{\partial x^*}\left(y^* x^{*-1/3}\right) = -\frac{1}{3} y^* x^{*-4/3} = -\frac{1}{3}\frac{\eta}{x^*}$$

In general, if $\eta = \frac{y^*}{x^{*n}}$*, then* $\frac{\partial \eta}{\partial x^*} = -n\frac{\eta}{x^*}$*. This is a very handy relationship to remember!*

So the x^*-derivative becomes:

$$\frac{\partial T^*}{\partial x^*} = \frac{1}{3}x^{*-2/3} f(\eta) - \frac{1}{3}x^{*-2/3}\eta f'(\eta) = \frac{1}{3}x^{*-2/3}(f - \eta f')$$

Now substitute everything into the original DE ($y^*\frac{\partial T^*}{\partial x^*} = \frac{\partial^2 T^*}{\partial y^{*2}}$):

$$y^*\left[\frac{1}{3}x^{*-2/3}(f - \eta f')\right] = x^{*-1/3} f''(\eta)$$

Since $y^* = \eta x^{*1/3}$, we can substitute this in on the left side:

$$(\eta x^{*1/3}) \left[\frac{1}{3} x^{*-2/3} (f - \eta f') \right] = x^{*-1/3} f''(\eta)$$

$$\frac{1}{3} \eta x^{*-1/3} (f - \eta f') = x^{*-1/3} f''(\eta)$$

Multiply both sides by $x^{*1/3}$ to eliminate the x^* terms completely:

$$f'' = \frac{1}{3} \eta (f - \eta f')$$

Subject to the boundary conditions: $f(\infty) = 0$ and $f'(0) = -1$.

This ODE is very easy to solve numerically (and even has an analytic solution!).

Note that we can now answer the core engineering question "**What is the temperature of the plate itself?**" without actually solving the problem:

$$T^*|_{y^*=0} = x^{*1/3} f(0)$$

Where $f(0)$ is simply an $O(1)$ constant that you have to solve the ODE to obtain. Actually, we find $f(0) = 1.5363$ using just a few lines of code!

Summary of Affine Stretching Procedure

1. **Define Stretched Variables:** Introduce arbitrary scaling parameters for all dependent and independent variables (e.g., let $T^* = A\overline{T}$, $x^* = B\overline{x}$, and $y^* = C\overline{y}$).
2. **Determine PDE Restrictions:** Substitute the stretched variables into the governing Partial Differential Equation (PDE). Factor out the leading parameter coefficient and set the remaining parameter groups equal to 1 to force the equation to be invariant.
3. **Determine Boundary Condition Restrictions:** Substitute the stretched variables into the boundary conditions (BCs). Homogeneous BCs (e.g., = 0) and BCs at 0 or ∞ typically add no restrictions. Non-homogeneous BCs will generate additional parameter restrictions (e.g., $A/C = 1$).
4. **Check for the "Free Wish":** Count the total number of arbitrary parameters and the total number of restrictions. If there is at least one more parameter than there are restrictions, the problem is invariant to a one-parameter group of continuous transformations and admits a self-similar solution.
5. **Apply the Canonical Form:** Choose one parameter to retain, typically the one associated with the "time-like" variable or the variable whose highest derivative in the PDE is the lowest. Solve for all other parameters in terms of this chosen parameter.
6. **Define Similarity Variables:** Group the variables to form invariant combinations. Use these combinations to define the independent similarity variable (e.g., $\eta = y^*/x^{*1/3}$) and the dependent similarity rule (e.g., $T^* = x^{*1/3} f(\eta)$).

7. **Transform the PDE to an ODE:** Use the chain rule to express the original partial derivatives in terms of the new similarity variable η and the function $f(\eta)$. Substitute these into the original PDE to reduce it to an Ordinary Differential Equation (ODE).
8. **Transform the Boundary Conditions:** Convert the original boundary conditions into boundary conditions for the new ODE in terms of f and η (e.g., evaluate at $\eta = 0$ and $\eta \to \infty$).

Similarity solutions do not solve every PDE. However, when they exist they often reveal the dominant physics more clearly than a full numerical solution because they expose the natural scaling structure of the problem. The affine stretching method described here captures a subset of possible self-similar solutions, however more sophisticated techniques such as trial functions can reveal additional cases. Historically, novel self-similar solutions to important transport problems have often become highly influential.

Practice Problems

Test your skills on these problems by using the Affine Stretching method to convert these Partial Differential Equations into Ordinary Differential Equations.

Problem 1: Stokes' First Problem (Suddenly Accelerated Plate)

Consider a semi-infinite body of viscous fluid ($y > 0$) that is initially at rest. At time $t = 0$, the solid boundary plate at $y = 0$ suddenly begins moving at a constant velocity U_0. The momentum diffuses into the fluid according to the 1-D unsteady Navier-Stokes equation:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

with boundary conditions $u(0, t) = U_0$ and $u(\infty, t) = 0$.

(a) First, render the problem dimensionless. Introduce an arbitrary characteristic time t_c, and use scaling to find the derived characteristic length L. Let $u^* = u/U_0$, $y^* = y/L$, and $t^* = t/t_c$.

(b) Stretch the dimensionless variables ($u^* = A\bar{u}$, $y^* = B\bar{y}$, $t^* = C\bar{t}$) to find the dimensionless similarity variable η and similarity rule $f(\eta)$.

(c) Transform the PDE into an ODE in terms of f and η.

Problem 2: Self-Similarity in a Non-Linear PDE

Affine stretching is particularly powerful because it works on non-linear PDEs where other methods fail. Consider the spreading of a viscous fluid through a porous medium. The fluid height $h(x, t)$ is governed by the non-linear Boussinesq equation:

$$\frac{\partial h}{\partial t} = \alpha \frac{\partial}{\partial x}\left(h \frac{\partial h}{\partial x}\right)$$

where α is a constant. The boundary conditions are $h(0,t) = H_0$ and $h(\infty,t) = 0$.

First, render the entire problem dimensionless using $h^* = h/H_0$, $x^* = x/x_c$, and $t^* = t/t_c$. Note how the lack of an independent length or time scale couples x_c and t_c. Then, use affine stretching on the dimensionless variables ($h^* = A\bar{h}^*$, $x^* = B\bar{x}^*$, $t^* = C\bar{t}^*$) to find the similarity variable η, the similarity rule $h(x,t) = H_0 f(\eta)$, and the resulting parameter-free non-linear ODE.

Problem 3: Similarity with an Integral Constraint (Instantaneous Source)

Suppose a syringe rapidly injects a total mass of dye M_0 into a narrow tube filled with water at $x = 0$ and $t = 0$. The dye diffuses outward according to:

$$\frac{\partial c}{\partial t} = D\frac{\partial^2 c}{\partial x^2}$$

We have boundary conditions $c(\pm\infty, t) = 0$. However, we don't have a constant concentration at the boundary to anchor our stretching! Instead, we use the fact that total mass is conserved:

$$\int_{-\infty}^{\infty} c(x,t)\,dx = M_0$$

(a) Render the problem dimensionless. Introduce an arbitrary characteristic time t_c. Use scaling on the PDE to find the length scale L, and use the integral constraint to find the concentration scale C_0.

(b) Stretch the dimensionless variables ($c^* = A\bar{c}$, $x^* = B\bar{x}$, $t^* = E\bar{t}$) to find your two restrictions. *(Note: We use E for the time parameter to avoid confusion with c).*

(c) Find the similarity variable η, show that the rule takes the form $c^* = (t^*)^{-1/2} f(\eta)$, and convert the PDE into an ODE.

Solutions to Practice Problems

Solution 1: Stokes' First Problem (Suddenly Accelerated Plate)

(a) Define dimensionless variables $u^* = u/U_0$, $t^* = t/t_c$, and $y^* = y/L$. Substitute into the PDE:

$$\left[\frac{U_0}{t_c}\right]\frac{\partial u^*}{\partial t^*} = \nu\left[\frac{U_0}{L^2}\right]\frac{\partial^2 u^*}{\partial y^{*2}} \quad \Longrightarrow \quad \frac{\partial u^*}{\partial t^*} = \left[\frac{\nu t_c}{L^2}\right]\frac{\partial^2 u^*}{\partial y^{*2}}$$

To balance the equation, set the group equal to 1, yielding $L = \sqrt{\nu t_c}$.

Our dimensionless PDE is now simply $\dfrac{\partial u^*}{\partial t^*} = \dfrac{\partial^2 u^*}{\partial y^{*2}}$ with boundary condition $u^*(0, t^*) = 1$.

(b) Stretch the variables: $u^* = A\bar{u}$, $y^* = B\bar{y}$, and $t^* = C\bar{t}$.

$$\frac{A}{C}\frac{\partial \bar{u}}{\partial \bar{t}} = \frac{A}{B^2}\frac{\partial^2 \bar{u}}{\partial \bar{y}^2} \quad \Longrightarrow \quad \frac{B^2}{C} = 1 \text{ (PDE Restriction)}$$

From the wall boundary condition, $u^*(0, t^*) = 1 \implies A\bar{u}(0, \bar{t}) = 1 \implies A = 1$.

Put the complexity in C: we get $A = 1$ and $B = C^{1/2}$.

Our invariant combinations are $u^*/1 = f(\eta)$ and $\eta = y^*/(t^*)^{1/2}$. Notice what happens when we plug our scales back into η:

$$\eta = \frac{y/L}{(t/t_c)^{1/2}} = \frac{y/\sqrt{\nu t_c}}{\sqrt{t/t_c}} = \frac{y}{\sqrt{\nu t}}$$

We recover the famous dimensionless similarity variable! Our rule is $u^* = f(\eta)$.

(c) Transform the derivatives using the chain rule on the dimensionless PDE:

$$\frac{\partial \eta}{\partial t^*} = -\frac{1}{2}\frac{\eta}{t^*} \quad \text{and} \quad \frac{\partial \eta}{\partial y^*} = \frac{1}{(t^*)^{1/2}}$$

$$\frac{\partial u^*}{\partial t^*} = f'(\eta)\left(-\frac{\eta}{2t^*}\right) \quad \text{and} \quad \frac{\partial^2 u^*}{\partial y^{*2}} = f''(\eta)\frac{1}{t^*}$$

Substitute into the PDE and multiply by t^*:

$$-\frac{1}{2}\eta f' = f'' \quad \Longrightarrow \quad f'' + \frac{1}{2}\eta f' = 0$$

with boundary conditions $f(0) = 1$ and $f(\infty) = 0$.

Notice how the ν is completely gone from the ODE thanks to our non-dimensionalization!

Solution 2: Self-Similarity in a Non-Linear PDE

We begin by defining dimensionless variables using characteristic scales for height (H_0), length (x_c), and time (t_c):

$$h^* = \frac{h}{H_0}, \quad x^* = \frac{x}{x_c}, \quad t^* = \frac{t}{t_c}$$

Substituting these into the Boussinesq equation yields:

$$\frac{H_0}{t_c}\frac{\partial h^*}{\partial t^*} = \frac{\alpha H_0^2}{x_c^2}\frac{\partial}{\partial x^*}\left(h^*\frac{\partial h^*}{\partial x^*}\right) \quad \Longrightarrow \quad \frac{\partial h^*}{\partial t^*} = \left(\frac{\alpha H_0 t_c}{x_c^2}\right)\frac{\partial}{\partial x^*}\left(h^*\frac{\partial h^*}{\partial x^*}\right)$$

To render the PDE completely parameter-free, we require the dimensionless grouping to equal unity:

$$\frac{\alpha H_0 t_c}{x_c^2} = 1 \quad \Longrightarrow \quad x_c = \sqrt{\alpha H_0 t_c}$$

Crucially, because the domain is semi-infinite ($x^* \in [0, \infty)$) and there is no fixed process time, the physical problem provides no independent value for x_c or t_c. They are inextricably linked. This collapse of independent scales is the definitive signature of self-similarity.

We can now define our dimensionless independent variables cleanly by choosing an arbitrary $t_c = t$, which forces $x_c = \sqrt{\alpha H_0 t}$. This directly yields the similarity variable η and similarity rule:

$$\eta = \frac{x^*}{\sqrt{t^*}} = \frac{x/x_c}{\sqrt{t/t_c}} = \frac{x}{\sqrt{\alpha H_0 t}} \quad \text{and} \quad h^* = f(\eta) \implies h(x,t) = H_0 f(\eta)$$

To formally complete the affine stretching exercise on our fully dimensionless system, we substitute $h^* = A\bar{h}^*$, $x^* = B\bar{x}^*$, and $t^* = C\bar{t}^*$ into our parameter-free dimensionless PDE:

$$\frac{A}{C}\frac{\partial \bar{h}^*}{\partial \bar{t}^*} = \frac{A^2}{B^2}\frac{\partial}{\partial \bar{x}^*}\left(\bar{h}^*\frac{\partial \bar{h}^*}{\partial \bar{x}^*}\right) \quad \Longrightarrow \quad \frac{B^2}{AC} = 1 \quad \text{(PDE Restriction)}$$

The wall boundary condition requires $h^*(0, t^*) = 1 \implies A\bar{h}^*(0, \bar{t}^*) = 1$, meaning we must select $A = 1$ for invariance.

Substituting $A = 1$ into our restriction gives $B^2/C = 1 \implies B = C^{1/2}$. The invariant combination of independent coordinates is:

$$\eta = \frac{x^*}{(t^*)^{1/2}} = \frac{x}{\sqrt{\alpha H_0 t}}$$

which perfectly matches our dimensional analysis prediction.

Next, we transform the PDE into an ODE by evaluating the derivatives of $h^* = f(\eta)$ via the chain rule:

$$\frac{\partial h^*}{\partial t^*} = -\frac{\eta}{2t^*}f' \quad \text{and} \quad \frac{\partial h^*}{\partial x^*} = \frac{1}{(t^*)^{1/2}}f'$$

Substituting these into the dimensionless PDE:

$$-\frac{\eta}{2t^*}f' = \frac{1}{(t^*)^{1/2}}\frac{\partial}{\partial \eta}\left(f \cdot \frac{1}{(t^*)^{1/2}}f'\right) = \frac{1}{t^*}(ff')'$$

Multiplying through by t^* eliminates the remaining time coordinate, leaving the pristine, parameter-free non-linear ODE:

$$-\frac{1}{2}\eta f' = (ff')'$$

subject to the cleanly normalized boundary conditions $f(0) = 1$ and $f(\infty) = 0$.

Solution 3: Similarity with an Integral Constraint (Instantaneous Source)

(a) Define $t^* = t/t_c$, $x^* = x/L$, and $c^* = c/C_0$. Substitute into the PDE:

$$\left[\frac{C_0}{t_c}\right]\frac{\partial c^*}{\partial t^*} = D\left[\frac{C_0}{L^2}\right]\frac{\partial^2 c^*}{\partial x^{*2}} \quad \Longrightarrow \quad \frac{\partial c^*}{\partial t^*} = \left[\frac{Dt_c}{L^2}\right]\frac{\partial^2 c^*}{\partial x^{*2}}$$

Set the group to 1 to find $L = \sqrt{Dt_c}$. The PDE becomes $\dfrac{\partial c^*}{\partial t^*} = \dfrac{\partial^2 c^*}{\partial x^{*2}}$.

Substitute into the integral constraint:

$$\int_{-\infty}^{\infty} C_0 c^* L\, dx^* = M_0 \quad \Longrightarrow \quad \left[\frac{C_0 L}{M_0}\right]\int_{-\infty}^{\infty} c^*\, dx^* = 1$$

Set the group to 1 to find $C_0 = M_0/L = M_0/\sqrt{Dt_c}$. The constraint becomes $\int c^* dx^* = 1$.

(b) Stretch the dimensionless variables: $c^* = A\bar{c}$, $x^* = B\bar{x}$, and $t^* = E\bar{t}$.

$$\frac{A}{E}\frac{\partial \bar{c}}{\partial \bar{t}} = \frac{A}{B^2}\frac{\partial^2 \bar{c}}{\partial \bar{x}^2} \quad \Longrightarrow \quad \frac{B^2}{E} = 1 \text{ (PDE Restriction)}$$

$$AB\int_{-\infty}^{\infty} \bar{c}\, d\bar{x} = 1 \quad \Longrightarrow \quad AB = 1 \text{ (Integral Restriction)}$$

(c) Put the complexity in E. From $B^2/E = 1$, we get $B = E^{1/2}$.

Substitute B into the integral restriction: $A(E^{1/2}) = 1 \implies A = E^{-1/2}$.

Our invariant combinations are $\eta = x^*/(t^*)^{1/2}$ and $f(\eta) = c^*/(t^*)^{-1/2}$. Note that η:

$$\eta = \frac{x/L}{\sqrt{t/t_c}} = \frac{x/\sqrt{Dt_c}}{\sqrt{t/t_c}} = \frac{x}{\sqrt{Dt}} \quad \text{(Dimensionless!)}$$

And our rule is $c^* = (t^*)^{-1/2} f(\eta)$. Taking the derivatives:

$$\frac{\partial c^*}{\partial t^*} = -\frac{1}{2}(t^*)^{-3/2}(f + \eta f') \quad \text{and} \quad \frac{\partial^2 c^*}{\partial x^{*2}} = (t^*)^{-3/2} f''(\eta)$$

Substitute these back into the dimensionless PDE:

$$-\frac{1}{2}(t^*)^{-3/2}(f + \eta f') = (t^*)^{-3/2} f'' \quad \Longrightarrow \quad f'' + \frac{1}{2}(f + \eta f') = 0$$

(Note: Since $f + \eta f'$ is the exact derivative of (ηf), this ODE can be directly integrated!)

20 Sturm-Liouville Theory

As we shall see in the next chapter, many partial differential equations can be solved via separation of variables. In PDEs arising from heat and mass transfer, the spatial part of such problems often result in a 2nd-order ODE called a Sturm-Liouville Boundary Value problem. The detailed theory behind the solution to such problems is described in many texts, notably in Chapter 11 of Boyce & DiPrima. That text, first published in 1965, remains the dominant text for undergraduate differential equations courses over 60 years later. Here we just state the theorem and look at a couple of examples.

A Sturm-Liouville (SL) problem has the form:

$$[p(x)y']' - q(x)y + \lambda w(x)y = 0$$

on the interval $0 < x < 1$, with homogeneous boundary conditions:

$$a_1 y(0) + a_2 y'(0) = 0$$
$$b_1 y(1) + b_2 y'(1) = 0$$

where p, p', q, and w are continuous functions of x on $[0, 1]$, and $p(x) > 0$ and $w(x) > 0$ on $[0, 1]$ (this condition is relaxed for singular SL problems, which are very similar).

For such a problem, we have the following properties:

1. All eigenvalues and corresponding eigenfunctions are real (not complex).
2. All eigenvalues are simple: there is a one-to-one correspondence with eigenfunctions, and they may be ordered by magnitude such that $\lambda_1 < \lambda_2 < \ldots$
3. **Orthogonality:** If $\phi_i(x)$ and $\phi_j(x)$ are two eigenfunctions (solutions) with corresponding eigenvalues $\lambda_i \neq \lambda_j$, then:

$$\int_0^1 \phi_i(x)\phi_j(x)w(x)\,dx = 0 \quad \text{if } i \neq j$$

   (where the weight $w(x)$ comes directly from the differential equation).
4. The set of eigenfunctions is **complete**: you can represent any function over $[0, 1]$ as a linear combination of the ϕ_i.

The Sturm–Liouville operator is a special type of linear operator called self-adjoint, which is the deeper mathematical reason why the eigenvalues are real and the eigenfunctions are orthogonal.

The Crucial Requirement of Homogeneity

It is absolutely critical to note that Sturm-Liouville theory—and the Principle of Superposition that allows us to sum up eigenfunctions—requires both the differential equation and the boundary conditions to be **homogeneous** (equal to zero).

Why? Because our final solution is a linear combination of many eigenfunctions: $y = \sum A_n \phi_n$. If our boundary condition was non-homogeneous, say $y(0) = 5$, then adding two valid eigenfunctions together would yield $y_1(0) + y_2(0) = 10$. The boundary condition would be instantly violated by superposition!

If you are faced with a transport problem with non-homogeneous boundary conditions (e.g., a wall held at a fixed temperature of 100°C), you *cannot* directly apply Sturm-Liouville theory. You must first transform the problem into a homogeneous one, typically by solving for the steady-state profile and defining a new variable as the deviation from that steady state. We will use this approach later when discussing separation of variables solutions to PDEs.

Example: Vibrating String

A vibrating string has a spatial part governed by:

$$y'' + \lambda y = 0$$

with boundary conditions $y(0) = 0$ and $y(1) = 0$.

For arbitrary λ, the only solution satisfying both boundary conditions is $y = 0$! This is known as the trivial solution. But for specific values of λ, there exist non-trivial solutions. These are the eigenvalues and eigenfunctions.

The general solution is:

$$y = A \sin(\sqrt{\lambda} x) + B \cos(\sqrt{\lambda} x)$$

Applying the boundary condition at $x = 0$:

$$y(0) = A \sin(0) + B \cos(0) = 0 \quad \Longrightarrow \quad B = 0$$

Applying the boundary condition at $x = 1$:

$$y(1) = A \sin(\sqrt{\lambda}) = 0 \quad \Longrightarrow \quad \sqrt{\lambda} = n\pi \quad \text{for } n = 1, 2, \ldots$$

Note that to obtain a non-trivial solution for satisfying this boundary condition we require λ to take on discrete values, the eigenvalues of the problem. The corresponding solutions are the eigenfunctions.

Because the eigenfunctions are complete, we can represent any function over $[0, 1]$ by a linear combination of the ϕ_n:

$$f(x) = \sum_{n=1}^{\infty} A_n \sin(n\pi x)$$

We get the coefficients A_n by invoking orthogonality! Simply multiply both sides by $\sin(m\pi x)$ and the weighting function $w(x)$ (which is just 1 in this case), and integrate:

$$\int_0^1 f(x)\sin(m\pi x)w(x)\,dx = \sum_{n=1}^{\infty}\int_0^1 A_n \sin(n\pi x)\sin(m\pi x)w(x)\,dx$$

Because of orthogonality, all terms in the infinite sum where $n \neq m$ integrate to exactly zero. The only surviving term is when $n = m$:

$$= \int_0^1 A_m(\sin(m\pi x))^2 w(x)\,dx$$

So, solving for A_m:

$$A_m = \frac{\int_0^1 f(x)\sin(m\pi x)w(x)\,dx}{\int_0^1 (\sin(m\pi x))^2 w(x)\,dx}$$

Now, in this specific case, $w(x) = 1$ and the denominator evaluates to $\int_0^1 \sin^2(m\pi x)\,dx = \frac{1}{2}$. Thus, swapping our index back to n, we get:

$$A_n = 2\int_0^1 f(x)\sin(n\pi x)\,dx$$

This is exactly the Fourier Expansion of $f(x)$!

$$f(x) = \sum_{n=1}^{\infty} A_n \sin(n\pi x)$$

where A_n is given above.

Operator Theory and the Inner Product

While we have used the practical term "eigenfunction expansion" to describe our solution method above, it is worth noting how modern mathematical physics frames this exact same process using **Operator Theory**.

In this viewpoint, the differential equation defines a linear spatial operator, $\mathcal{L}$. Because the Sturm-Liouville operator is self-adjoint, the Sturm-Liouville theorem guarantees that its eigenfunctions $\phi_n(x)$ form a **complete** and **orthogonal** basis. You can think of these eigenfunctions as the fundamental, perpendicular "axes" of a functional workspace (a Hilbert space).

When we multiplied both sides by the eigenfunction and the weight $w(x)$ and integrated, we were formally taking what is known as the **inner product**. The inner product of two functions $u(x)$ and $v(x)$ is commonly denoted by angle brackets:

$$\langle u, v\rangle = \int_0^1 u(x)v(x)w(x)\,dx$$

Using this unified notation, the calculation for the Fourier coefficients A_n that we just performed can be written more compactly as:

$$A_n = \frac{\langle f, \phi_n \rangle}{\langle \phi_n, \phi_n \rangle}$$

This operator perspective illustrates that multiplying by $w(x)$ and integrating is the formal geometric *projection* of your initial condition $f(x)$ onto the basis function $\phi_n(x)$, analogous to using a dot product to find the x-component of a geometric vector.

Converting to Sturm-Liouville Form

You might look at the standard Sturm-Liouville equation, $[p(x)y']' - q(x)y + \lambda w(x)y = 0$, and think it looks nothing like the differential equations you normally encounter. However, almost *any* linear, second-order ODE can be forced into this exact form using a clever integrating factor trick.

Suppose you have a general second-order ODE:

$$y'' + P(x)y' + Q(x)y + \lambda R(x)y = 0$$

To convert this, we multiply the entire equation by an integrating factor, $I(x)$, defined as:

$$I(x) = \exp\left(\int P(x)\, dx\right)$$

Because of the chain rule, the derivative of $I(x)$ is simply $I'(x) = I(x)P(x)$. When we multiply our ODE by $I(x)$, the first two terms become:

$$I(x)y'' + I(x)P(x)y' = I(x)y'' + I'(x)y' = [I(x)y']'$$

Substituting this back into the multiplied ODE, we get:

$$[I(x)y']' + I(x)Q(x)y + \lambda I(x)R(x)y = 0$$

This is exactly the Sturm-Liouville form! By matching terms, we see that $p(x) = I(x)$, $q(x) = -I(x)Q(x)$, and $w(x) = I(x)R(x)$. Consequently, many linear transport problems can be analyzed using the powerful machinery of Sturm-Liouville theory. To apply the standard theorem, however, the resulting weight function must remain positive and the associated boundary conditions must be homogeneous.

Convective Boundaries and Transcendental Eigenvalues

In the vibrating string example, our boundary conditions were perfectly fixed ($y = 0$), which yielded clean, analytical eigenvalues: $\sqrt{\lambda} = n\pi$. In heat and mass transfer, however, we usually deal with convective boundaries (e.g., Newton's Law of Cooling).

Suppose we have the same spatial equation $y'' + \lambda y = 0$, but a convective (Robin) boundary condition at $x = 1$:

$$\begin{aligned} y(0) &= 0 \\ y'(1) + hy(1) &= 0 \quad \text{(where } h \text{ is a heat transfer coefficient)} \end{aligned}$$

From the first boundary condition, $y(0) = 0$, we again find that $B = 0$, so our solution is $y = A\sin(\sqrt{\lambda}x)$.

Now, we evaluate the derivative $y' = A\sqrt{\lambda}\cos(\sqrt{\lambda}x)$ and apply the convective boundary condition at $x = 1$:

$$A\sqrt{\lambda}\cos(\sqrt{\lambda}) + hA\sin(\sqrt{\lambda}) = 0$$

Dividing by $A\cos(\sqrt{\lambda})$, we get:

$$\sqrt{\lambda} + h\tan(\sqrt{\lambda}) = 0 \quad \implies \quad \tan(\sqrt{\lambda}) = -\frac{\sqrt{\lambda}}{h}$$

We can no longer solve for λ by simply writing down an integer series! This is a **transcendental equation**. There are still an infinite number of discrete eigenvalues λ_n where the tangent curve intersects the line $-\sqrt{\lambda}/h$, but to find them, you must use a numerical root-finder (like MATLAB's `fzero` from the previous section) to locate each root one by one.

The Physical Meaning of the Weighting Function $w(x)$

In the orthogonality relationship, the integral must include the weighting function $w(x)$. This is not merely a mathematical artifact; it is deeply tied to the physical geometry of your system.

Consider the heat equation in a cylinder. The radial part of the Laplacian in cylindrical coordinates is:

$$\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial T}{\partial r}\right)$$

If you apply separation of variables, the resulting spatial ODE naturally takes the form:

$$[ry']' + \lambda ry = 0$$

By comparing this to the standard SL equation, you can immediately see that $p(r) = r$ and $w(r) = r$.

Why is the weight function exactly r? Because in a cylindrical system, the physical volume of a differential element is $dV = r\,dr\,d\theta\,dz$. The r that appears in the SL weight function is the exact same r that accounts for the increasing volume of material as you move outward from the center of the cylinder.

Because $p(0) = 0$ at the centerline of the cylinder, this is known as a **Singular Sturm-Liouville problem**, which gives rise to Bessel functions. Even though it is singular, the orthogonality theorem still holds, provided you remember to include the physical weight $w(r) = r$ in your integration!

Non-Homogeneous Equations and Green's Functions

As emphasized earlier, Sturm-Liouville theory requires a completely homogeneous problem to work directly. But what if our differential equation has a forcing function (a source term)? Suppose we are solving for the steady-state temperature in a rod with internal heat generation, or the deflection of a string under a distributed load $f(x)$. The equation looks like this:

$$[p(x)y']' - q(x)y = f(x)$$

with homogeneous boundary conditions $y(0) = 0$ and $y(1) = 0$.

To solve this, we use a key concept from linear systems: **The Green's Function**.

Instead of trying to solve for the arbitrary load $f(x)$ all at once, we ask a simpler question: *What is the response of the system to a single, concentrated point source of integrated magnitude 1 located at a specific position $x = \xi$?*

Mathematically, a point source is represented by the **Dirac delta function**, $\delta(x - \xi)$. The response of the system to this point source is called the Green's Function, denoted as $G(x, \xi)$. It is the solution to:

$$[p(x)G']' - q(x)G = \delta(x - \xi)$$

Why do we care about a point source? Because of the principle of superposition! If we know the response $G(x, \xi)$ to a single point source at ξ, we can find the response to *any* distributed load $f(x)$ by simply adding up (integrating) the point-source responses weighted by the actual load $f(\xi)$ at every location ξ:

$$y(x) = \int_0^1 G(x, \xi) f(\xi)\, d\xi$$

This is a tremendous simplification: it transforms the complicated differential equation into a straightforward integral! But how do we find $G(x, \xi)$? We build it out of Sturm-Liouville eigenfunctions!

Eigenfunction Expansion of the Green's Function

Let the differential operator be $L[y] = [p(x)y']' - q(x)y$. The associated Sturm-Liouville eigenvalue problem is $L[\phi_n] + \lambda_n w(x)\phi_n = 0$, which means $L[\phi_n] = -\lambda_n w(x)\phi_n$.

Because the eigenfunctions $\phi_n(x)$ form a complete set, we can express the Green's function as an infinite series of these eigenfunctions:

$$G(x, \xi) = \sum_{n=1}^{\infty} a_n \phi_n(x)$$

We plug this series into our point-source equation $L[G] = \delta(x - \xi)$:

$$L\left[\sum_{n=1}^{\infty} a_n \phi_n(x)\right] = \delta(x - \xi)$$

$$\sum_{n=1}^{\infty} a_n L[\phi_n(x)] = \delta(x - \xi)$$

$$\sum_{n=1}^{\infty} a_n \left(-\lambda_n w(x) \phi_n(x)\right) = \delta(x - \xi)$$

Now, we use the property of orthogonality to isolate a_n. We multiply both sides by $\phi_m(x)$ and integrate from 0 to 1:

$$-\sum_{n=1}^{\infty} a_n \lambda_n \int_0^1 \phi_n(x) \phi_m(x) w(x)\, dx = \int_0^1 \delta(x - \xi) \phi_m(x)\, dx$$

Because of orthogonality, the integral on the left is zero for all terms except when $n = m$. On the right side, a fundamental property of the Dirac delta function is that integrating it against any function simply "sifts out" the value of that function at the location of the spike ($x = \xi$). Thus:

$$-a_n \lambda_n \int_0^1 \phi_n^2(x) w(x)\, dx = \phi_n(\xi)$$

Solving for our unknown coefficients a_n:

$$a_n = \frac{-\phi_n(\xi)}{\lambda_n \int_0^1 \phi_n^2(x) w(x)\, dx}$$

Substituting this back into our original series yields the **Bilinear Expansion of the Green's Function**:

$$G(x, \xi) = \sum_{n=1}^{\infty} \frac{-\phi_n(\xi) \phi_n(x)}{\lambda_n \int_0^1 \phi_n^2(x) w(x)\, dx}$$

The minus sign arises from our choice of operator definition. Many texts define the operator with the opposite sign.

Example: A Loaded String

Consider a taut string fixed at both ends, subjected to an arbitrary distributed load $f(x)$. The governing equation is:

$$y'' = f(x) \quad \text{with} \quad y(0) = 0,\ y(1) = 0$$

The associated homogeneous SL problem is $y'' + \lambda y = 0$. As we derived previously, the eigenvalues are $\lambda_n = (n\pi)^2$ and the eigenfunctions are $\phi_n(x) = \sin(n\pi x)$. The weight function is $w(x) = 1$.

First, we compute the integral in the denominator:

$$\int_0^1 \sin^2(n\pi x)(1)\,dx = \frac{1}{2}$$

Now, we simply plug our ϕ_n, λ_n, and the integral value into the Bilinear Expansion formula to get the Green's function for the string:

$$G(x,\xi) = \sum_{n=1}^{\infty} \frac{-\sin(n\pi\xi)\sin(n\pi x)}{(n\pi)^2(1/2)} = -\frac{2}{\pi^2}\sum_{n=1}^{\infty}\frac{\sin(n\pi\xi)\sin(n\pi x)}{n^2}$$

Finally, the exact displacement of the string $y(x)$ under *any* load $f(x)$ is found by simply evaluating the integral:

$$y(x) = \int_0^1 \left(-\frac{2}{\pi^2}\sum_{n=1}^{\infty}\frac{\sin(n\pi\xi)\sin(n\pi x)}{n^2}\right) f(\xi)\,d\xi$$

This elegantly demonstrates how the homogeneous solutions (eigenfunctions) are the fundamental building blocks used to solve entirely non-homogeneous problems!

Final Note: The solutions to many Sturm-Liouville problems are available in classical references such as Carslaw and Jaeger, Conduction of Heat in Solids, since transient heat-conduction problems naturally lead to Sturm-Liouville eigenvalue problems. The text also treats related problems involving finite thermal or mass reservoirs, for which the eigenvalue appears in a boundary condition. In such cases the usual Sturm-Liouville orthogonality relation is modified by the addition of a boundary term in the inner product. In general, Sturm-Liouville problems are quite easy to solve numerically in MATLAB using an eigenvalue solver. We will use this again when discussing PDE solutions via separation of variables.

Practice Problems

Test your skills on these problems. Recognizing Sturm-Liouville forms, understanding their physical geometric origins, and utilizing Green's functions are critical steps before attempting to solve any partial differential equation.

Problem 1: Converting to SL Form

Hermite's differential equation frequently arises in quantum mechanics (e.g., when modeling the quantum harmonic oscillator). The ODE is given by:

$$y'' - 2xy' + \lambda y = 0$$

Use the integrating factor method to convert this into standard Sturm-Liouville form. Explicitly identify the functions $p(x)$, $q(x)$, and $w(x)$.

Problem 2: The Physical Weight Function

In spherically symmetric heat conduction problems, the radial spatial equation takes the following form after separation of variables:

$$y'' + \frac{2}{r}y' + \lambda y = 0$$

(a) Use the integrating factor trick to put this differential equation into standard Sturm-Liouville form.

(b) Identify the weight function $w(r)$. Based on the section text, explain the physical/geometric significance of this specific weight function.

Problem 3: Using a Green's Function

In the notes, we derived the Bilinear Expansion of the Green's function for a string fixed at $x = 0$ and $x = 1$:

$$G(x, \xi) = -\frac{2}{\pi^2} \sum_{n=1}^{\infty} \frac{\sin(n\pi\xi)\sin(n\pi x)}{n^2}$$

Suppose this string is subjected to a constant downward uniform load (like gravity), such that $f(x) = -g$, where g is a positive constant.

Use the Green's function integral $y(x) = \int_0^1 G(x, \xi) f(\xi)\, d\xi$ to find the infinite series solution for the string's displacement $y(x)$.

Solutions to Practice Problems

Solution 1: Converting to SL Form

We start by identifying the coefficient of the y' term, which is $P(x) = -2x$.

Next, we calculate the integrating factor $I(x)$:

$$I(x) = \exp\left(\int P(x)\,dx\right) = \exp\left(\int -2x\,dx\right) = e^{-x^2}$$

Multiply the entire original ODE by this integrating factor:

$$e^{-x^2}y'' - 2xe^{-x^2}y' + \lambda e^{-x^2}y = 0$$

Recognize that the first two terms are the exact result of the product rule applied to $(e^{-x^2}y')'$. We can rewrite the equation as:

$$[e^{-x^2}y']' + \lambda e^{-x^2}y = 0$$

Comparing this to the standard SL form $[p(x)y']' - q(x)y + \lambda w(x)y = 0$, we can clearly identify:

$$p(x) = e^{-x^2}$$
$$q(x) = 0$$
$$w(x) = e^{-x^2}$$

Because the weighting function is $w(x) = e^{-x^2}$, any eigenfunctions found for this equation will be orthogonal with respect to the weight e^{-x^2}!

Solution 2: The Physical Weight Function

(a) The coefficient of the y' term is $P(r) = \dfrac{2}{r}$. We calculate the integrating factor:

$$I(r) = \exp\left(\int \frac{2}{r}\,dr\right) = \exp(2\ln r) = r^2$$

Multiplying the entire ODE by r^2 yields:

$$r^2y'' + 2ry' + \lambda r^2 y = 0$$

The first two terms collapse via the product rule into $(r^2y')'$, giving the final Sturm-Liouville form:

$$[r^2y']' + \lambda r^2 y = 0$$

(b) Comparing this to the standard SL form, the weight function is $w(r) = r^2$.
Physically, this represents the geometry of a sphere. The differential volume element in spherical coordinates is $dV = r^2 \sin\theta\, dr\, d\theta\, d\phi$. The r^2 term exactly accounts for the quadratically increasing volume of spherical shells as you move radially outward from the origin.

Solution 3: Using a Green's Function

We are given $f(\xi) = -g$. We substitute this and our Green's function into the superposition integral:

$$y(x) = \int_0^1 G(x,\xi) f(\xi)\, d\xi$$
$$y(x) = \int_0^1 \left(-\frac{2}{\pi^2} \sum_{n=1}^{\infty} \frac{\sin(n\pi\xi)\sin(n\pi x)}{n^2} \right) (-g)\, d\xi$$

Because the integral is with respect to ξ, we can pull the constants, the summation, and the $\sin(n\pi x)$ term outside the integral:

$$y(x) = \frac{2g}{\pi^2} \sum_{n=1}^{\infty} \frac{\sin(n\pi x)}{n^2} \int_0^1 \sin(n\pi\xi)\, d\xi$$

Now, we evaluate the definite integral:

$$\int_0^1 \sin(n\pi\xi)\, d\xi = \left[-\frac{\cos(n\pi\xi)}{n\pi} \right]_0^1 = -\frac{\cos(n\pi) - \cos(0)}{n\pi} = \frac{1 - \cos(n\pi)}{n\pi}$$

Recall that $\cos(n\pi) = (-1)^n$. Thus, the integral evaluates to $\frac{1-(-1)^n}{n\pi}$. Note that this value is 0 for even n, and $\frac{2}{n\pi}$ for odd n.

Substituting this back into our expression for $y(x)$, we get our final series solution:

$$y(x) = \frac{2g}{\pi^2} \sum_{n=1}^{\infty} \frac{\sin(n\pi x)}{n^2} \left(\frac{1 - (-1)^n}{n\pi} \right)$$
$$y(x) = \frac{2g}{\pi^3} \sum_{n=1}^{\infty} \frac{1 - (-1)^n}{n^3} \sin(n\pi x)$$

Note: This represents the Fourier sine series of the parabolic deflection of a string under a constant load. You wouldn't solve the problem this way of course - the differential equation is directly integrable - but the same principle applies to much more complicated and difficult problems which aren't!

21 Solving PDEs via Separation of Variables

Some PDEs can be solved via separation of variables. The PDE must be linear in the dependent variable, and it also needs a separable structure. Many transport problems (particularly transients) qualify!

Key Steps

1. **Determine the asymptotic solution.** This should satisfy any inhomogeneities in the Differential Equation (DE) or the Boundary Conditions (BCs).
2. **Subtract off this solution to get the decaying part.** Substitute into the DE to get a homogeneous DE (with homogeneous BCs) for the decaying part. This bit now has an inhomogeneous Initial Condition (IC) created by subtracting off the asymptotic solution.
3. **Assume the solution to the decaying part is a product** of a spatial part and a time-dependent part. Substitute this into the DE and divide out.
4. If one side of the equation is a function only of time and the other side is a function only of the spatial variable, **both must be constant**: you get two ODEs.
5. **Solve the time-dependent part.** This is usually just an exponential.
6. **The spatial part becomes a Sturm-Liouville eigenvalue problem** once the BCs have been rendered homogeneous.
7. **Determine the eigenvalues and eigenfunctions** of the Sturm-Liouville (SL) problem.
8. **Get the coefficients of your new Series Solution** (one term for each eigenvalue) using orthogonality and the IC created by subtracting off the asymptotic solution.
9. **The complete solution is the sum** of the asymptotic and decaying solutions.

Example: Start-Up Flow in a Tube

Let's see how this works! A vertical straw of radius a is filled with a fluid of viscosity μ and density ρ. At $t = 0$, you take your finger off the end and flow starts.

Question: What is the centerline velocity as a function of time?

Governing Equation (DE):

$$\rho\frac{\partial u}{\partial t} = \mu\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right) + \rho g$$

Boundary Conditions (BCs):

$$u|_{r=a} = 0 \quad \text{and} \quad \left.\frac{\partial u}{\partial r}\right|_{r=0} = 0 \quad \text{(symmetry)}$$

Initial Condition (IC):

$$u|_{t=0} = 0$$

Step 0: Scaling

Note that you don't *have* to render a problem dimensionless before solving it by this or other techniques, but it is always a *very* good idea! Thus, let's call this **Step 0**.

We start by scaling. Let $r^* = \frac{r}{a}$, $u^* = \frac{u}{U_c}$, and $t^* = \frac{t}{t_c}$. Substitute into the DE:

$$\left[\frac{\rho U_c}{t_c}\right]\frac{\partial u^*}{\partial t^*} = \left[\frac{\mu U_c}{a^2}\right]\frac{1}{r^*}\frac{\partial}{\partial r^*}\left(r^*\frac{\partial u^*}{\partial r^*}\right) + \rho g$$

Divide by the viscous group, as viscosity is expected to matter:

$$\left[\frac{\rho a^2}{\mu t_c}\right]\frac{\partial u^*}{\partial t^*} = \frac{1}{r^*}\frac{\partial}{\partial r^*}\left(r^*\frac{\partial u^*}{\partial r^*}\right) + \left[\frac{\rho g a^2}{\mu U_c}\right]$$

To balance the physical mechanisms, we set both the bracketed groups equal to 1:

$$\implies U_c = \frac{\rho g a^2}{\mu} \quad \text{and} \quad t_c = \frac{\rho a^2}{\mu}$$

So our dimensionless PDE is:

$$\frac{\partial u^*}{\partial t^*} = \frac{1}{r^*}\frac{\partial}{\partial r^*}\left(r^*\frac{\partial u^*}{\partial r^*}\right) + 1$$

with $u^*|_{r^*=1} = 0$, $\left.\frac{\partial u^*}{\partial r^*}\right|_{r^*=0} = 0$, and $u^*|_{t^*=0} = 0$.

Now the BCs are already homogeneous, but the DE isn't (due to the +1 gravity term)!

Step 1 & 2: Asymptotic Solution and Decaying Part

At long times ($t^* \gg 1$), the flow is steady (note: this is not always true for all problems, but it is here). Let this steady asymptotic solution be u^*_∞.

$$0 = \frac{1}{r^*}\frac{\partial}{\partial r^*}\left(r^*\frac{\partial u^*_\infty}{\partial r^*}\right) + 1 \qquad \implies \qquad \frac{1}{r^*}\frac{\partial}{\partial r^*}\left(r^*\frac{\partial u^*_\infty}{\partial r^*}\right) = -1$$

Integrating once:

$$r^* \frac{\partial u^*_\infty}{\partial r^*} = -\frac{1}{2} r^{*2} + C_1$$

From the symmetry BC at $r^* = 0$, we find $C_1 = 0$. Integrating again:

$$u^*_\infty = -\frac{1}{4} r^{*2} + C_2$$

From the no-slip BC at $r^* = 1$, $C_2 = \frac{1}{4}$. Thus, the asymptotic solution is the classic Poiseuille flow:

$$u^*_\infty = \frac{1}{4}(1 - r^{*2})$$

Now subtract off u^*_∞ to find the decaying part u^*_d:

$$u^* = u^*_\infty + u^*_d$$

Substituting this into the dimensionless PDE yields a homogeneous DE for u^*_d:

$$\frac{\partial u^*_d}{\partial t^*} = \frac{1}{r^*} \frac{\partial}{\partial r^*} \left(r^* \frac{\partial u^*_d}{\partial r^*} \right)$$

with homogeneous BCs: $u^*_d|_{r^*=1} = 0$ and $\left.\dfrac{\partial u^*_d}{\partial r^*}\right|_{r^*=0} = 0$.

However, our Initial Condition for u^*_d is now inhomogeneous!

$$u^*_d|_{t^*=0} = u^*|_{t^*=0} - u^*_\infty|_{t^*=0} = 0 - \frac{1}{4}(1 - r^{*2}) = -\frac{1}{4}(1 - r^{*2})$$

Steps 3-7: Separation of Variables & Sturm-Liouville

Let $u^*_d(r^*, t^*) = F(r^*)G(t^*)$. Substitute this into the homogeneous DE:

$$FG' = G \frac{1}{r^*} (r^* F')'$$

Divide out by FG:

$$\frac{G'}{G} = \frac{(r^* F')'}{r^* F} = \text{constant} = -\sigma^2 \quad (\text{or } -\lambda)$$

(We choose a negative constant $-\sigma^2$ because we are looking for a decaying solution!)

This gives two ODEs:

1. $G' = -\sigma^2 G \implies G(t^*) = e^{-\sigma^2 t^*}$

2. $(r^* F')' + \sigma^2 r^* F = 0$ with $F'(0) = 0$ and $F(1) = 0$.

This second ODE is a Sturm-Liouville eigenvalue problem! Specifically, it is Bessel's differential equation of order zero:

$$F(r^*) = J_0(\sigma r^*)$$

The second solution, the Bessel function of the second kind $Y_0(\sigma r^*)$, is eliminated by the condition at $r^* = 0$.

To satisfy the boundary condition $F(1) = 0$, the eigenvalues σ_n must be the roots of the Bessel function:

$$J_0(\sigma_n) = 0$$

Step 8 & 9: Series Solution and Orthogonality

Because this is a linear system, the total decaying solution is the infinite sum of all valid modes:

$$u_d^* = \sum_{n=1}^{\infty} A_n e^{-\sigma_n^2 t^*} J_0(\sigma_n r^*)$$

Now we use the initial condition to find the constants A_n:

$$u_d^*|_{t^*=0} = \sum_{n=1}^{\infty} A_n J_0(\sigma_n r^*) = -\frac{1}{4}(1 - r^{*2})$$

By the property of orthogonality (using the weight function r^* from the SL DE!):

$$A_n = \frac{\int_0^1 \left[-\frac{1}{4}(1 - r^{*2})\right] J_0(\sigma_n r^*) r^* \, dr^*}{\int_0^1 \left[J_0(\sigma_n r^*)\right]^2 r^* \, dr^*} = \frac{-2}{\sigma_n^3 J_1(\sigma_n)}$$

Note that in the "old days" you would look up this integral in the classic tome by Gradshteyn and Ryzhik. Now, of course, you would just plug it into Wolfram Alpha on your iPhone...

Thus, the complete velocity profile is:

$$u^*(r^*, t^*) = \frac{1}{4}(1 - r^{*2}) + \sum_{n=1}^{\infty} \frac{-2}{\sigma_n^3 J_1(\sigma_n)} e^{-\sigma_n^2 t^*} J_0(\sigma_n r^*)$$

Example 2: Separation of Variables for Elliptic PDEs

While we usually employ this method for parabolic PDEs (like the transient tube flow above), it is perfectly suited for other separable problems, including steady **elliptic PDEs**. The solution procedure is exactly the same: render the equation dimensionless, determine an asymptotic solution, subtract it off to render the problem homogeneous, and solve the resulting Sturm-Liouville problem!

Example: Consider steady, fully developed pressure-driven flow of a fluid through a rectangular duct of dimension $2a$ in the x-direction and $2b$ in the y-direction. Assume $a \gg b$ (the duct is much wider than it is tall). This is a fundamental problem in microfluidic Lab-on-a-Chip design.

Governing Equation (DE):

$$\mu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) = \frac{dp}{dz}$$

Boundary Conditions (BCs):

$$u(\pm a, y) = 0 \quad \text{and} \quad u(x, \pm b) = 0$$

Step 0: Scaling

Let $x^* = x/a$, $y^* = y/b$, and $u^* = u/U_c$. Substituting into the PDE yields:

$$\frac{\mu U_c}{a^2}\frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\mu U_c}{b^2}\frac{\partial^2 u^*}{\partial y^{*2}} = \frac{dp}{dz}$$

Because the duct is very wide ($a \gg b$), the y-derivative will dominate the bulk flow. We balance it with the pressure gradient:

$$\frac{\mu U_c}{b^2} = -\frac{dp}{dz} \qquad \Longrightarrow \qquad U_c = \frac{b^2}{\mu}\left(-\frac{dp}{dz}\right)$$

Substituting this velocity scale back into the equation, we obtain our dimensionless PDE:

$$\left(\frac{b}{a}\right)^2 \frac{\partial^2 u^*}{\partial x^{*2}} + \frac{\partial^2 u^*}{\partial y^{*2}} = -1$$

Step 1 & 2: Asymptotic Solution and Decaying Part

In this geometry, our "asymptotic" solution isn't found at $t \to \infty$, but rather far away from the side walls (near the center of the very wide duct). If we ignore the presence of the side walls, the flow is purely 1-D between parallel plates:

$$0 + \frac{d^2 u^*_\infty}{dy^{*2}} = -1 \qquad \Longrightarrow \qquad u^*_\infty(y^*) = \frac{1}{2}(1 - y^{*2})$$

(This satisfies the BCs $u^*_\infty(\pm 1) = 0$ at the top and bottom walls).

We subtract this off to find the decaying part: $u^*_d(x^*, y^*) = u^* - u^*_\infty$.
The PDE becomes homogeneous:

$$\left(\frac{b}{a}\right)^2 \frac{\partial^2 u^*_d}{\partial x^{*2}} + \frac{\partial^2 u^*_d}{\partial y^{*2}} = 0$$

The top and bottom BCs are homogeneous: $u^*_d(x^*, \pm 1) = 0$.
The side-wall BCs ($x^* = \pm 1$) absorb the inhomogeneity, acting like our "initial condition" from the transient problem:

$$u^*_d(\pm 1, y^*) = 0 - u^*_\infty(y^*) = -\frac{1}{2}(1 - y^{*2})$$

Steps 3-7: Separation of Variables

Let $u^*_d = X(x^*)Y(y^*)$. Substitute into the homogeneous PDE and divide by XY:

$$\left(\frac{b}{a}\right)^2 \frac{X''}{X} = -\frac{Y''}{Y} = \lambda^2$$

This yields two ODEs. Because the homogeneous BCs are at $y^* = \pm 1$, the SL eigenvalue problem is in the y-direction:

1. $Y'' + \lambda^2 Y = 0 \implies Y(y^*) = \cos(\lambda y^*)$ (keeping only the even function due to symmetry). Evaluating at the wall, $Y(1) = 0 \implies \cos(\lambda) = 0 \implies \lambda_n = \frac{(2n-1)\pi}{2}$.
2. $X'' - \left(\lambda_n \frac{a}{b}\right)^2 X = 0 \implies X(x^*) = \cosh\left(\lambda_n \frac{a}{b} x^*\right)$ (again, keeping the symmetric solution).

Step 8 & 9: Series Solution and Orthogonality

The total decaying solution is:

$$u_d^* = \sum_{n=1}^{\infty} A_n \cosh\left(\lambda_n \frac{a}{b} x^*\right) \cos(\lambda_n y^*)$$

We evaluate this at the side wall $x^* = 1$ to find our constants using the inhomogeneous BC:

$$u_d^*(1, y^*) = \sum_{n=1}^{\infty} \left[A_n \cosh\left(\lambda_n \frac{a}{b}\right)\right] \cos(\lambda_n y^*) = -\frac{1}{2}(1 - y^{*2})$$

Using orthogonality to isolate the bracketed constant:

$$A_n \cosh\left(\lambda_n \frac{a}{b}\right) = \frac{\int_{-1}^{1} \left[-\frac{1}{2}(1 - y^{*2})\right] \cos(\lambda_n y^*) dy^*}{\int_{-1}^{1} \cos^2(\lambda_n y^*) dy^*} = \frac{2(-1)^n}{\lambda_n^3}$$

Solving for A_n and adding the asymptotic solution back in, we get our final complete velocity profile:

$$u^*(x^*, y^*) = \frac{1}{2}(1 - y^{*2}) + \sum_{n=1}^{\infty} \frac{2(-1)^n}{\lambda_n^3 \cosh\left(\lambda_n \frac{a}{b}\right)} \cosh\left(\lambda_n \frac{a}{b} x^*\right) \cos(\lambda_n y^*)$$

Note on Elliptic PDEs: In the rectangular duct example, the "decay" happens in **space** rather than time. The x direction acts as the time-like variable. Because of the cosh terms, the influence of the side walls decays exponentially as you move inward toward the center of the duct. If the duct is extremely wide ($a/b \gg 1$), the side-wall effects die out rapidly, and the bulk of the duct closely follows the plane-Poiseuille result without side walls. It is interesting to note that even though the deviation is small far from the side walls, the slow-down there is actually very important! In microfluidic chip design, for example, the effect of the side walls dominates the dispersion of solute slugs in Lab-on-a-Chip systems - by a factor of about 7 for large aspect ratio channels. This is often the limiting factor in the separation of solute slugs for high throughput screening designs. We will look at this again when we discuss numerical solutions to PDEs.

What Do We Actually Care About?

In both of the examples above, you can get everything analytically (which is nice), but often there is no analytic solution! No problem: just solve the Sturm-Liouville problem numerically.

In engineering practice, what do we actually care about? In order of importance:

1. **Asymptotic solution:** What is left at the end!

2. **Lead eigenvalue:** Tells us how fast the asymptotic solution is reached!
3. **Lead eigenfunction & coefficient:** This is the dominant part of the decaying solution at long times!
4. **Everything else...** (Usually negligible after a short transient period).

For the transient tube flow problem, the first root is $J_0(\sigma_1) = 0 \implies \sigma_1 = 2.4048$. Thus, the velocity decays as $e^{-\sigma_1^2 t^*} = e^{-5.78 t^*}$.

- It is mostly gone at $t^* \sim \frac{1}{\sigma_1^2} = 0.173$.
- It is $\sim 90\%$ gone at $t^* = 0.4$! (And $\sim 99\%$ gone by $t^* = 0.8$).

Higher order terms decay much faster (and in this case have smaller coefficients). The lead coefficient is $A_1 \approx -0.277$, which is very close to the steady centerline velocity $u^*_\infty|_{r^*=0} = 0.25$. So, to a very good approximation, the centerline velocity is simply:

$$u^*|_{r^*=0} \approx 0.25 - 0.277 e^{-5.78 t^*}$$

provided $t^* > 0.1$ (as the higher modes decay incredibly fast!). This example is typical: in most cases you can approximate your transient function with only the asymptotic solution and the first eigenfunction. While the full expansion is necessary at very short times (before the higher eigenfunctions have decayed away), short times are usually amenable to boundary layer-type analysis which gives a better picture of the short time physics than does the infinite series solution of the SL problem.

Example 3. Separation of Variables Solutions to Hyperbolic PDEs

A **hyperbolic PDE** is superficially quite similar to the elliptic PDE (e.g., the Laplace equation) which we just considered, yet it has very different mathematical characteristics. It is, however, often amenable to solution via the same separation of variables technique. Consider the two-dimensional Laplace equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

In contrast, the equivalent hyperbolic PDE (the 1D wave equation) is:

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

The difference between these two becomes clear if you move the spatial derivative to the left side—it is just a negative sign! The consequence of this is profound. When we solved the elliptic PDE via separation of variables, we had sines and cosines in one direction, forcing us to have exponentials (or hyperbolic sines/cosines) in the other. For the hyperbolic PDE, however, the negative sign flips the eigenvalue problem. The separation of variables solution leads to oscillatory solutions in *both* of the separated parts: in other words, the solution is oscillatory in both space and time.

This sort of equation occurs throughout physics, chemistry, and engineering because it is the fundamental mathematical model governing the propagation of waves. In electromagnetism, Maxwell's equations decouple into hyperbolic wave equations that govern the propagation of light and other electromagnetic radiation through space. In acoustics, it governs the propagation of sound waves through gases and liquids. In structural mechanics, it determines the transient vibrations of taut strings and elastic membranes. In fluid systems, it describes pressure transients such as the acoustic surge associated with water hammer in a pipeline. Here we shall confine ourselves to the simplest example: the harp string.

The Harp String

If you pluck a harp string, you get a nice, pure tone—the dominant mode of the vibration. As any musician knows, the tone depends on the string's mass per unit length (gut yields a higher pitch than heavy metal), the length (a longer string gives a lower pitch), and the tension (what you actually turn the pegs to fine-tune).

The governing equation for the transverse displacement $u(x, t)$ is derived from a simple momentum balance: $F = ma$. The mass m of a differential string element is the mass per unit length ρ_L times dx. The net restoring force F is the tension T times the local curvature of the string. This yields the classic 1D wave equation:

$$\rho_L \frac{\partial^2 u}{\partial t^2} = T \frac{\partial^2 u}{\partial x^2} \quad \Longrightarrow \quad \frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}$$

where the wave speed is $c = \sqrt{T/\rho_L}$.

We have fixed boundary conditions at both ends of the string: $u(0, t) = 0$ and $u(L, t) = 0$. The problem is already homogeneous, but we want to render it dimensionless by scaling it (Step 0!). Let's define our dimensionless spatial variable as $\xi = x/L$, our dimensionless displacement as $\theta = u/u_0$ (where u_0 is the initial pluck amplitude), and our dimensionless time as $\tau = t/(L/c)$. Substituting these in yields the beautifully clean dimensionless wave equation:

$$\frac{\partial^2 \theta}{\partial \tau^2} = \frac{\partial^2 \theta}{\partial \xi^2}$$

with boundary conditions $\theta(0, \tau) = 0$ and $\theta(1, \tau) = 0$.

We separate the function into the product of the spatial part and the time-dependent part:

$$\theta(\xi, \tau) = X(\xi)\Gamma(\tau)$$

Substituting this into our dimensionless PDE and dividing by $X\Gamma$ gives:

$$\frac{1}{\Gamma}\frac{d^2\Gamma}{d\tau^2} = \frac{1}{X}\frac{d^2X}{d\xi^2} = -\lambda^2$$

The spatial part is the exact same Sturm-Liouville eigenvalue problem we considered earlier:

$$\frac{d^2X}{d\xi^2} + \lambda^2 X = 0$$

Applying the boundary conditions $X(0) = 0$ and $X(1) = 0$, we find the eigenvalues are $\lambda_n = n\pi$ for $n = 1, 2, 3, \ldots$, and the corresponding eigenfunctions are:

$$X_n(\xi) = \sin(n\pi\xi)$$

Now for the time-dependent part. Unlike the elliptic problem which gave us exponentials, the negative sign in the hyperbolic PDE gives us another oscillatory ODE:

$$\frac{d^2\Gamma}{d\tau^2} + (n\pi)^2\Gamma = 0$$

The general solution for this is:

$$\Gamma_n(\tau) = A_n \cos(n\pi\tau) + B_n \sin(n\pi\tau)$$

Applying the principle of superposition, the complete dimensionless solution is the infinite series:

$$\theta(\xi, \tau) = \sum_{n=1}^{\infty} \sin(n\pi\xi) \left[A_n \cos(n\pi\tau) + B_n \sin(n\pi\tau)\right]$$

If we map this back to our dimensional variables (x and t), the actual physical solution looks like this:

$$u(x, t) = \sum_{n=1}^{\infty} \sin\left(\frac{n\pi x}{L}\right) \left[A_n \cos\left(\frac{n\pi ct}{L}\right) + B_n \sin\left(\frac{n\pi ct}{L}\right)\right]$$

The specific constants A_n and B_n would be determined from the initial condition (the exact shape of the string when it was plucked at $t = 0$ and its initial velocity). However, the key point of interest for a musician is the frequency of vibration of the dominant ($n = 1$) mode.

The argument inside the time-dependent cosine is $2\pi\nu t$, where ν is the frequency in Hertz. Setting the physical argument equal to our solution's time dependence gives:

$$2\pi\nu_n t = \frac{n\pi ct}{L}$$
$$\nu_n = \frac{nc}{2L}$$

Plugging our wave speed $c = \sqrt{T/\rho_L}$ back in, the frequency of the n-th harmonic is:

$$\nu_n = \frac{n}{2L}\sqrt{\frac{T}{\rho_L}}$$

Higher modes ($n = 2, 3, \ldots$) produce higher frequencies (overtones), however, fluid damping from the air and internal friction within the string (which were not included in our ideal equation) would cause these higher-frequency modes to rapidly decay, leaving the pure fundamental tone ($n = 1$) of the string.

Practice Problems

Test your skills on these problems. Remember the core sequence: render the problem dimensionless, find the asymptotic (steady) solution, subtract it off to create a decaying problem with homogeneous boundary conditions, solve the spatial ODE as a Sturm-Liouville eigenvalue problem, and use orthogonality to find the series coefficients!

Problem 1: Start-Up of Couette Flow

Consider a viscous fluid between two parallel infinite plates separated by a distance h. Initially, the fluid is completely at rest. At $t = 0$, the top plate ($y = h$) suddenly begins moving horizontally with a constant velocity U_0, while the bottom plate ($y = 0$) remains stationary. The velocity $u(y, t)$ is governed by:

$$\frac{\partial u}{\partial t} = \nu \frac{\partial^2 u}{\partial y^2}$$

with BCs $u(0, t) = 0$ and $u(h, t) = U_0$, and IC $u(y, 0) = 0$.

(a) Find the steady asymptotic solution $u_\infty(y)$.

(b) Define the decaying solution $u_d(y, t)$. What are its PDE, BCs, and IC?

(c) Use separation of variables ($u_d = F(y)G(t)$) to find the eigenvalues λ_n and eigenfunctions $F_n(y)$.

(d) Use orthogonality to find the series coefficients and write the final complete solution $u(y, t)$. *(Hint:* $\int x \sin(ax)\, dx = \frac{\sin(ax)}{a^2} - \frac{x\cos(ax)}{a}$*)*

Problem 2: Heat Conduction with Generation

A flat wall of thickness $2L$ (spanning from $x = -L$ to $x = L$) has an initial uniform temperature T_0. At $t = 0$, a uniform internal heat generation begins throughout the wall at a volumetric rate $\dot{q}$ (W/m^3). The surfaces at $x = \pm L$ are perfectly maintained at T_0.

$$\rho \hat{C}_p \frac{\partial T}{\partial t} = k \frac{\partial^2 T}{\partial x^2} + \dot{q}$$

Find the complete transient temperature profile $T(x, t)$.

Problem 3: Steady Heat Conduction in an Asymmetric Semi-Infinite Slab (Elliptic)

Consider steady-state heat conduction in a solid slab of thermal conductivity k and width $2b$ (spanning from $y = -b$ to $y = b$ in the cross-direction). The slab is bounded by a base wall at $x = 0$ and extends to infinity in the positive x-direction. The base at $x = 0$ is maintained at a constant baseline temperature T_0. However, the side walls are asymmetric: the right wall at $y = b$ is kept at

$T_0 + \Delta T$, while the left wall at $y = -b$ is kept at $T_0 - \Delta T$. The temperature field $T(x, y)$ satisfies Laplace's equation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

First, render the problem dimensionless by defining $x^* = x/b$, $y^* = y/b$, and a shifted, scaled dimensionless temperature $\theta^* = \frac{T-T_0}{\Delta T}$. Find the complete dimensionless temperature profile $\theta^*(x^*, y^*)$.

(Hint: Because the domain is semi-infinite in x^*, use exponentials $e^{\pm\lambda x^*}$ rather than hyperbolic functions for the x-dependence, throwing out the unphysical growing term. Pay close attention to how the asymmetry affects both your asymptotic solution and your choice of spatial eigenfunctions!)

Problem 4: The Clarinet (Mixed Boundary Conditions)

We saw in the text that a harp string, fixed at both ends, produces all integer harmonics of its fundamental frequency. But what happens if we change the boundary conditions?

Consider the acoustic pressure wave inside a clarinet or a stopped organ pipe. The pipe has a length L. One end of the pipe ($x = 0$) is blocked by a solid mouthpiece or plug, while the other end ($x = L$) is open to the room. The governing equation for the gauge pressure $p(x, t)$ is the 1D wave equation:

$$\frac{\partial^2 p}{\partial t^2} = c^2 \frac{\partial^2 p}{\partial x^2}$$

where c is the speed of sound in air.

The boundary conditions are driven by the physics at the ends of the pipe: At the closed end ($x = 0$), the air cannot move. This implies a zero pressure gradient: $\frac{\partial p}{\partial x}(0, t) = 0$. At the open end ($x = L$), the pressure must equalize with the surrounding room, meaning the gauge pressure is zero: $p(L, t) = 0$.

Use the method of separation of variables to find the general series solution for $p(x, t)$. Then, determine the resonance frequencies of the pipe. How do these frequencies relate to the fundamental frequency compared to the harp string?

Solutions to Practice Problems

Solution 1: Start-Up of Couette Flow

(a) Define $u^* = u/U_0$, $y^* = y/h$, and $t^* = t/t_c$. Substitute into the PDE:

$$\left[\frac{U_0}{t_c}\right]\frac{\partial u^*}{\partial t^*} = \nu\left[\frac{U_0}{h^2}\right]\frac{\partial^2 u^*}{\partial y^{*2}} \quad \Longrightarrow \quad \frac{\partial u^*}{\partial t^*} = \left[\frac{\nu t_c}{h^2}\right]\frac{\partial^2 u^*}{\partial y^{*2}}$$

Balance the equation by setting the group to 1: $t_c = h^2/\nu$.

The dimensionless PDE is $\dfrac{\partial u^*}{\partial t^*} = \dfrac{\partial^2 u^*}{\partial y^{*2}}$ with BCs $u^*(0,t^*) = 0$ and $u^*(1,t^*) = 1$, and IC $u^*(y^*,0) = 0$.

(b) The asymptotic solution is steady ($\frac{\partial u^*}{\partial t^*} = 0$):

$$0 = \frac{d^2 u^*_\infty}{dy^{*2}} \quad \Longrightarrow \quad u^*_\infty(y^*) = C_1 y^* + C_2$$

Applying the BCs $u^*_\infty(0) = 0$ and $u^*_\infty(1) = 1$, we find the linear Couette profile: $u^*_\infty(y^*) = y^*$.

(c) The decaying solution is $u^*_d = u^* - u^*_\infty$.

The PDE is $\dfrac{\partial u^*_d}{\partial t^*} = \dfrac{\partial^2 u^*_d}{\partial y^{*2}}$. The BCs become homogeneous: $u^*_d(0,t^*) = 0$ and $u^*_d(1,t^*) = 0$.

The IC absorbs the inhomogeneity: $u^*_d(y^*,0) = 0 - y^* = -y^*$.

(d) Let $u^*_d = F(y^*)G(t^*)$. Substitute into the PDE and divide by FG:

$$\frac{G'}{G} = \frac{F''}{F} = -\lambda^2$$

The time ODE is $G' = -\lambda^2 G \implies G(t^*) = e^{-\lambda^2 t^*}$.

The spatial ODE is $F'' + \lambda^2 F = 0 \implies F(y^*) = A\sin(\lambda y^*) + B\cos(\lambda y^*)$.

Apply $F(0) = 0 \implies B = 0$. Apply $F(1) = 0 \implies \sin(\lambda) = 0$.

Thus, the eigenvalues are $\lambda_n = n\pi$ and the eigenfunctions are $F_n(y^*) = \sin(n\pi y^*)$.

(e) The total decaying solution is $u^*_d = \sum_{n=1}^{\infty} A_n e^{-(n\pi)^2 t^*}\sin(n\pi y^*)$.

Apply the IC: $u^*_d(y^*,0) = \sum_{n=1}^{\infty} A_n \sin(n\pi y^*) = -y^*$.

Using orthogonality (weight function $w = 1$):

$$A_n = \frac{\int_0^1 (-y^*)\sin(n\pi y^*)dy^*}{\int_0^1 \sin^2(n\pi y^*)dy^*} = \frac{\left[-\frac{\sin(n\pi y^*)}{(n\pi)^2} + \frac{y^*\cos(n\pi y^*)}{n\pi}\right]_0^1}{1/2} = 2\left(\frac{\cos(n\pi)}{n\pi}\right) = \frac{2(-1)^n}{n\pi}$$

The complete dimensionless velocity profile is:

$$u^*(y^*, t^*) = y^* + \sum_{n=1}^{\infty} \frac{2(-1)^n}{n\pi} e^{-(n\pi)^2 t^*} \sin(n\pi y^*)$$

Solution 2: Heat Conduction with Generation

Substitute $\theta = \Delta T_c \theta^*$, $x = Lx^*$, and $t = t_c t^*$ into the PDE $\rho \hat{C}_p \dfrac{\partial \theta}{\partial t} = k \dfrac{\partial^2 \theta}{\partial x^2} + \dot{q}$:

$$\left[\frac{\rho \hat{C}_p \Delta T_c}{t_c}\right] \frac{\partial \theta^*}{\partial t^*} = \left[\frac{k \Delta T_c}{L^2}\right] \frac{\partial^2 \theta^*}{\partial x^{*2}} + \dot{q} \quad \Longrightarrow \quad \left[\frac{\rho \hat{C}_p L^2}{k t_c}\right] \frac{\partial \theta^*}{\partial t^*} = \frac{\partial^2 \theta^*}{\partial x^{*2}} + \left[\frac{\dot{q} L^2}{k \Delta T_c}\right]$$

Balance the terms: $t_c = \dfrac{\rho \hat{C}_p L^2}{k} = \dfrac{L^2}{\alpha}$ and $\Delta T_c = \dfrac{\dot{q} L^2}{k}$.

The dimensionless PDE is $\dfrac{\partial \theta^*}{\partial t^*} = \dfrac{\partial^2 \theta^*}{\partial x^{*2}} + 1$. The BCs are $\theta^*(\pm 1, t^*) = 0$ and IC is $\theta^*(x^*, 0) = 0$.

The steady asymptotic solution θ^*_∞ satisfies $0 = \theta^{*''}_\infty + 1$. Integrating and applying the symmetric BCs $\theta^*_\infty(\pm 1) = 0$ yields: $\theta^*_\infty(x^*) = \frac{1}{2}(1 - x^{*2})$.

The decaying solution $\theta^*_d = \theta^* - \theta^*_\infty$ satisfies the homogeneous PDE $\dfrac{\partial \theta^*_d}{\partial t^*} = \dfrac{\partial^2 \theta^*_d}{\partial x^{*2}}$ with homogeneous BCs $\theta^*_d(\pm 1) = 0$.

The IC is $\theta^*_d(x^*, 0) = -\theta^*_\infty(x^*) = -\frac{1}{2}(1 - x^{*2})$.

Separation of variables ($F'' + \lambda^2 F = 0$) with symmetric BCs at ± 1 dictates we only keep the even cosine terms: $F(x^*) = \cos(\lambda x^*)$. $F(1) = 0 \implies \cos(\lambda) = 0 \implies \lambda_n = \frac{(2n-1)\pi}{2}$.

The series is $\theta^*_d = \sum_{n=1}^{\infty} A_n e^{-\lambda_n^2 t^*} \cos(\lambda_n x^*)$. Using orthogonality:

$$A_n = \frac{\int_{-1}^{1} \left[-\frac{1}{2}(1 - x^{*2})\right] \cos(\lambda_n x^*) dx^*}{\int_{-1}^{1} \cos^2(\lambda_n x^*) dx^*}$$

The denominator is 1. The numerator evaluates to $-\frac{2(-1)^{n-1}}{\lambda_n^3} = \frac{2(-1)^n}{\lambda_n^3}$.

Thus, the final dimensionless temperature profile is:

$$\theta^*(x^*, t^*) = \frac{1}{2}(1 - x^{*2}) + \sum_{n=1}^{\infty} \frac{2(-1)^n}{\lambda_n^3} e^{-\lambda_n^2 t^*} \cos(\lambda_n x^*)$$

Solution 3: Steady Heat Conduction in an Asymmetric Semi-Infinite Slab (Elliptic)

Substitute $x = bx^*$, $y = by^*$, and $T = T_0 + \Delta T\theta^*$ into Laplace's equation:

$$\left[\frac{\Delta T}{b^2}\right]\frac{\partial^2\theta^*}{\partial x^{*2}} + \left[\frac{\Delta T}{b^2}\right]\frac{\partial^2\theta^*}{\partial y^{*2}} = 0 \quad \Longrightarrow \quad \frac{\partial^2\theta^*}{\partial x^{*2}} + \frac{\partial^2\theta^*}{\partial y^{*2}} = 0$$

Let's look at our transformed boundary conditions:

- Right side wall: $T(x, b) = T_0 + \Delta T \implies \theta^*(x^*, 1) = 1$
- Left side wall: $T(x, -b) = T_0 - \Delta T \implies \theta^*(x^*, -1) = -1$
- Base wall: $T(0, y) = T_0 \implies \theta^*(0, y^*) = 0$
- Boundedness at infinity: $\lim_{x\to\infty} T < \infty \implies \lim_{x^*\to\infty} \theta^*(x^*, y^*) < \infty$

Step 1 & 2: Asymptotic Solution and Decaying Part

Far down the slab ($x^* \to \infty$), the temperature profile becomes independent of x^* but must satisfy the asymmetric side walls. This gives our 1-D asymptotic steady solution $\theta^*_\infty(y^*)$:

$$0 + \frac{d^2\theta^*_\infty}{dy^{*2}} = 0 \quad \Longrightarrow \quad \theta^*_\infty(y^*) = C_1 y^* + C_2$$

Applying $\theta^*_\infty(1) = 1$ and $\theta^*_\infty(-1) = -1$ yields $C_1 = 1$ and $C_2 = 0$. Thus, $\theta^*_\infty(y^*) = y^*$.

Now subtract this off to define the decaying part: $\theta^*_d(x^*, y^*) = \theta^* - y^*$.

The PDE remains homogeneous: $\dfrac{\partial^2\theta^*_d}{\partial x^{*2}} + \dfrac{\partial^2\theta^*_d}{\partial y^{*2}} = 0.$

The side-wall BCs become nicely homogeneous: $\theta^*_d(x^*, \pm 1) = (\pm 1) - (\pm 1) = 0.$

The base-wall BC absorbs the shift, acting like our inhomogeneous "initial condition":

$$\theta^*_d(0, y^*) = \theta^*(0, y^*) - y^* = 0 - y^* = -y^*$$

Steps 3-7: Separation of Variables

Let $\theta^*_d = X(x^*)Y(y^*)$. Substituting and dividing by XY yields:

$$\frac{X''}{X} = -\frac{Y''}{Y} = \lambda^2$$

The spatial Sturm-Liouville problem is in the y-direction: $Y'' + \lambda^2 Y = 0$ with $Y(\pm 1) = 0$.

Because our driving base condition ($\theta^*_d = -y^*$) is an **odd** (anti-symmetric) function, it will only excite anti-symmetric modes. Therefore, we discard the cosine terms and keep only the sines:

$$Y(y^*) = \sin(\lambda y^*)$$

Applying the boundary condition at the wall: $Y(1) = \sin(\lambda) = 0 \implies \lambda_n = n\pi$ for $n = 1, 2, 3, \ldots$

Next, solve the x-direction ODE: $X'' - (n\pi)^2 X = 0$. The general solution is:

$$X(x^*) = C_1 e^{n\pi x^*} + C_2 e^{-n\pi x^*}$$

As $x^* \to \infty$, the growing exponential $e^{n\pi x^*} \to \infty$, which violates physical boundedness. We must set $C_1 = 0$. Only the decaying spatial exponential $e^{-n\pi x^*}$ is physical!

Step 8 & 9: Series Solution and Orthogonality
Constructing our total decaying series solution:

$$\theta_d^*(x^*, y^*) = \sum_{n=1}^{\infty} A_n e^{-n\pi x^*} \sin(n\pi y^*)$$

We isolate the coefficients A_n by evaluating the series at the base wall ($x^* = 0$) where $\theta_d^* = -y^*$:

$$\theta_d^*(0, y^*) = \sum_{n=1}^{\infty} A_n \sin(n\pi y^*) = -y^*$$

Using orthogonality (with a weight function $w = 1$ across the domain $y^* \in [-1, 1]$):

$$A_n = \frac{\int_{-1}^{1} (-y^*) \sin(n\pi y^*) dy^*}{\int_{-1}^{1} \sin^2(n\pi y^*) dy^*}$$

The denominator integrates to 1. Evaluating the numerator via integration by parts yields:

$$A_n = \left[\frac{y^* \cos(n\pi y^*)}{n\pi} - \frac{\sin(n\pi y^*)}{(n\pi)^2} \right]_{-1}^{1} = \frac{2\cos(n\pi)}{n\pi} = \frac{2(-1)^n}{n\pi}$$

Adding our linear asymptotic solution back in, we arrive at the complete dimensionless temperature profile:

$$\theta^*(x^*, y^*) = y^* + \sum_{n=1}^{\infty} \frac{2(-1)^n}{n\pi} e^{-n\pi x^*} \sin(n\pi y^*)$$

Solution 4: The Clarinet (Mixed Boundary Conditions)

Step 0: Non-Dimensionalization Before we do any calculus, it is always a good idea to render the problem dimensionless! Let's define our dimensionless spatial variable as $\xi = x/L$, our dimensionless gauge pressure as $\theta = p/p_0$ (where p_0 is a characteristic initial pressure amplitude), and our dimensionless time as $\tau = t/(L/c)$.

Using the chain rule to substitute these into the wave equation gives:

$$\frac{p_0}{(L/c)^2} \frac{\partial^2 \theta}{\partial \tau^2} = c^2 \frac{p_0}{L^2} \frac{\partial^2 \theta}{\partial \xi^2} \quad \implies \quad \frac{\partial^2 \theta}{\partial \tau^2} = \frac{\partial^2 \theta}{\partial \xi^2}$$

Our boundary conditions also transform perfectly. The closed end (zero gradient) at $x = 0$ becomes $\frac{\partial \theta}{\partial \xi}(0, \tau) = 0$. The open end (zero pressure) at $x = L$ becomes $\theta(1, \tau) = 0$.

Step 1: Separation of Variables We assume the dimensionless pressure can be separated into a spatial function and a time function:

$$\theta(\xi, \tau) = X(\xi)\Gamma(\tau)$$

Substituting this into our dimensionless PDE and dividing by $X\Gamma$ yields:

$$\frac{1}{\Gamma}\frac{d^2\Gamma}{d\tau^2} = \frac{1}{X}\frac{d^2X}{d\xi^2} = -\lambda^2$$

Step 2: The Spatial Problem (Sturm-Liouville) We extract the spatial ODE:

$$\frac{d^2X}{d\xi^2} + \lambda^2 X = 0$$

The general solution is our familiar combination of sines and cosines:

$$X(\xi) = A\cos(\lambda\xi) + B\sin(\lambda\xi)$$

Now we apply the boundary conditions. First, the closed end (the Neumann condition). We need the derivative of $X(\xi)$:

$$\frac{dX}{d\xi} = -A\lambda\sin(\lambda\xi) + B\lambda\cos(\lambda\xi)$$

Setting $\xi = 0$ gives:

$$\frac{dX}{d\xi}(0) = B\lambda = 0 \quad \Longrightarrow \quad B = 0$$

Unlike the harp string where the sine terms survived, the Neumann boundary condition kills the sine term. Our spatial solution is now strictly a cosine profile: $X(\xi) = A\cos(\lambda\xi)$.

Next, apply the open end (Dirichlet) boundary condition at $\xi = 1$:

$$X(1) = A\cos(\lambda) = 0$$

For this to be true without the trivial solution ($A = 0$), the argument λ must be a root of the cosine function. Cosine equals zero at $\pi/2, 3\pi/2, 5\pi/2$, etc. We can express this mathematically as:

$$\lambda_n = \frac{(2n-1)\pi}{2} \quad \text{for } n = 1, 2, 3, \ldots$$

Our spatial eigenfunctions are therefore:

$$X_n(\xi) = \cos\left(\frac{(2n-1)\pi\xi}{2}\right)$$

Step 3: The Time-Dependent Problem The time ODE is:

$$\frac{d^2\Gamma}{d\tau^2} + \lambda_n^2\Gamma = 0$$

Which gives the standard oscillatory solution:

$$\Gamma_n(\tau) = C_n \cos(\lambda_n \tau) + D_n \sin(\lambda_n \tau)$$

Step 4: The General Solution Combining the spatial and temporal parts using superposition gives the full dimensionless series:

$$\theta(\xi, \tau) = \sum_{n=1}^{\infty} \cos\left(\frac{(2n-1)\pi\xi}{2}\right)\left[C_n \cos\left(\frac{(2n-1)\pi\tau}{2}\right) + D_n \sin\left(\frac{(2n-1)\pi\tau}{2}\right)\right]$$

To answer physical questions, we map this back to our dimensional variables (x and t):

$$p(x, t) = \sum_{n=1}^{\infty} \cos\left(\frac{(2n-1)\pi x}{2L}\right)\left[C_n \cos\left(\frac{(2n-1)\pi ct}{2L}\right) + D_n \sin\left(\frac{(2n-1)\pi ct}{2L}\right)\right]$$

Step 5: Finding the Frequencies The physical frequency ν_n in Hertz is found by matching $2\pi\nu_n t$ to the time argument in our dimensional solution:

$$2\pi\nu_n t = \frac{(2n-1)\pi ct}{2L}$$
$$\nu_n = \frac{(2n-1)c}{4L}$$

The fundamental frequency ($n = 1$) is $\nu_1 = c/(4L)$.

If we plug in $n = 2$, the next frequency is $3c/(4L)$, which is exactly $3\nu_1$. The next is $5\nu_1$. Because of the mixed boundary conditions, the pipe entirely skips the even harmonics (the octaves) and only produces the **odd harmonics**. This mathematical result directly explains the distinctive, hollow timbre of a clarinet. In contrast, a flute (where you blow over an open hole) has a Dirichlet boundary condition ($p = 0$) at both ends, preserves the even harmonics, and sounds quite different!

Part V

Numerical & Stochastic Solutions for PDEs

22 Numerical Solutions via Separation of Variables

While analytical solutions to Sturm-Liouville problems are elegant, many engineering problems involve variable physical properties or complex geometries where an exact solution is impossible to find. In these cases, we solve the Sturm-Liouville problem numerically.

To do this, we discretize the spatial domain and use second-order finite difference approximations to convert the continuous differential operator into a matrix eigenvalue problem of the form:

$$\mathbf{A}\mathbf{y} = -\lambda \mathbf{W}\mathbf{y}$$

where $\mathbf{A}$ is the discretized matrix representation of our spatial differential operators, $\mathbf{W}$ is a diagonal weight matrix, and $\mathbf{y}$ is the eigenvector representing the spatial shape of the eigenfunction.

The Generalized Sturm-Liouville Solver: `slsolve.m`

The following MATLAB function solves a generalized regular Sturm-Liouville eigenvalue problem defined on the domain $0 < x < 1$:

$$\frac{d}{dx}\left[p(x)\frac{dy}{dx}\right] + q(x)y + \lambda w(x)y = 0$$

subject to the linear boundary conditions:

$$bc(1)y(0) + bc(2)y'(0) = 0$$
$$bc(3)y(1) + bc(4)y'(1) = 0$$

To maintain second-order accuracy ($O(1/n^2)$) throughout the entire domain, the function employs a midpoint grid for the internal derivatives and one-sided, three-point second-order finite differences at the boundaries.

```
function [lambda, eigenvec] = slsolve(varargin)
% SLSOLVE Solves the Sturm-Liouville eigenvalue problem:
% [p(x) y']' + q(x) y + lambda * w(x) y = 0
%
% Subject to the boundary conditions:
% bc(1)*y(0) + bc(2)*y'(0) = 0
% bc(3)*y(1) + bc(4)*y'(1) = 0
% over the domain 0 < x < 1.
%
% Call syntax:
% [lambda, eigenvec] = slsolve(p, q, w, bc, n);
%
% p, q, w : Anonymous functions or function handles able to accept arrays.
% bc      : Array containing [bc(1), bc(2), bc(3), bc(4)].
% n       : Degree of discretization (number of intervals). Default is 50.

p  = varargin{1};
q  = varargin{2};
w  = varargin{3};
bc = varargin{4};

if nargin < 5
    n = 50;
else
    n = varargin{5};
end

h = 1/n;          % Set spatial step size
x = (0:h:1)';     % Column array of node locations

% Initialize coefficient arrays
pp = zeros(1, n+1);
pm = zeros(1, n+1);
ww = zeros(1, n+1);
qq = zeros(1, n+1);

% Evaluate functions at node midpoints and node centers
for i = 2:n
    pp(i) = feval(p, x(i) + h/2);
    pm(i) = feval(p, x(i) - h/2);
    ww(i) = feval(w, x(i));
    qq(i) = feval(q, x(i));
end

% The weight matrix W is easy:
weight = -diag(ww);
```

```
% Assemble the main tridiagonal matrix A
a = diag(-pp - pm - qq * h^2);      % Main diagonal

a = a + diag(pp(1:n), 1);           % Superdiagonal

a = a + diag(pm(2:n+1), -1);        % Subdiagonal

a = a / h^2;                        % Divide out cell area width scale

% Enforce second-order boundary conditions using 3-point formulations
% Left boundary (x = 0):
a(1,1) = bc(1) - bc(2) * 1.5 / h;
a(1,2) = bc(2) * 2.0 / h;
a(1,3) = -bc(2) / (2*h);

% Right boundary (x = 1):
a(n+1, n+1) = bc(3) + bc(4) * 1.5 / h;
a(n+1, n)   = -bc(4) * 2.0 / h;
a(n+1, n-1) = bc(4) / (2*h);

% Calculate the eigenvalues and eigenvectors
[v, d] = eig(a, weight);

evals = diag(d);
i = find(isfinite(evals));          % Keep only finite valid modes
evals = evals(i);
evecs = v(:, i);

% Sort modes by ascending absolute real value of eigenvalues
[~, i] = sort(abs(real(evals)));
lambda = evals(i);
evecs  = evecs(:, i);

% Normalize eigenfunctions by infinity norm and enforce positive starting slope
eigenvec = zeros(size(evecs));
for j = 1:length(lambda)
    eigenvec(:,j) = evecs(:,j) / norm(evecs(:,j), inf) / sign(evecs(2,j));
end
```

Note on numerical accuracy: Because this script utilizes second-order derivative approximations, the truncation error in your results scales as $O(1/n^2)$. Consequently, while the first few lead eigenvalues will be highly reliable, accuracy deteriorates significantly for higher-order modes.

The code provided here is in MATLAB, however it should run natively in Octave and can easily be translated by any AI into other languages as desired. Surprisingly, at the time of this writing MATLAB does not have a native Sturm-Liouville solving function. More accurate estimates of the eigenvalues and eigenfunctions can be determined via standard boundary value solvers such as `bvp4c.m` and iteration on the eigenvalue, however these would need to be determined one-by-one

starting from a particular initial guess for the eigenvalue. The code provided here is convenient because it determines all of them simultaneously and gets the first ones with high accuracy (which can be improved by increasing the discretization parameter n).

Example: Startup of Flow in a Tube (Revisited)

Let's apply our numerical solver to the fluid startup problem analyzed analytically in Section 16. Recall that the steady-state asymptotic velocity profile is given by the classic Poiseuille solution:

$$u_\infty^*(r^*) = \frac{1}{4}(1 - r^{*2})$$

The spatial operators resulting from separating variables in cylindrical coordinates map onto the Sturm-Liouville parameters as follows:

$$p(r^*) = r^*, \quad q(r^*) = 0, \quad w(r^*) = r^*$$

The centerline symmetry requires $u_r^*(0) = 0$, and no-slip at the wall requires $u^*(1) = 0$, yielding the boundary condition array `bc = [0, 1, 1, 0]`.

The following script initializes these conditions, runs our solver with $n = 100$ spatial intervals, and applies the Trapezoidal Rule to determine the series coefficients (A_n) via numerical orthogonality:

```
% Define the asymptotic steady-state solution
uinf = @(r) (1 - r.^2) / 4;
% Define Sturm-Liouville function handles
p = @(x) x;
q = @(x) zeros(size(x));
w = @(x) x;
bc = [0, 1, 1, 0]; % Centerline symmetry and no-slip wall
n = 100; % Grid discretization density
% Execute numerical solver
[lambda, eigenvecs] = slsolve(p, q, w, bc, n);
% Initialize independent grid and Trapezoidal Rule integration weights
r = (0:1/n:1)';
weights = ones(1, n+1);
weights(1) = 0.5;
weights(n+1) = 0.5;
weights = weights / n;
% Calculate series coefficients via numerical orthogonality integration
a = zeros(length(lambda), 1);
for i = 1:length(lambda)
    numerator = -weights * (w(r) .* uinf(r) .* eigenvecs(:, i));
    denominator = weights * (w(r) .* (eigenvecs(:, i).^2));
    a(i) = numerator / denominator;
end
```

```
% Display the lead 5 eigenvalues and corresponding coefficients
firsteigenvals = lambda(1:5)
firstcoefficients = a(1:5)
% Plot the first five eigenfunctions
figure(1)
plot(r, eigenvecs(:, 1:5), LineWidth=1.5)
xlabel('Dimensionless Radius (r^*)')
ylabel('Eigenfunction Magnitude (y)')
title('First Five Spatial Eigenfunctions')
legend("n = " + string(1:5))
grid on
```

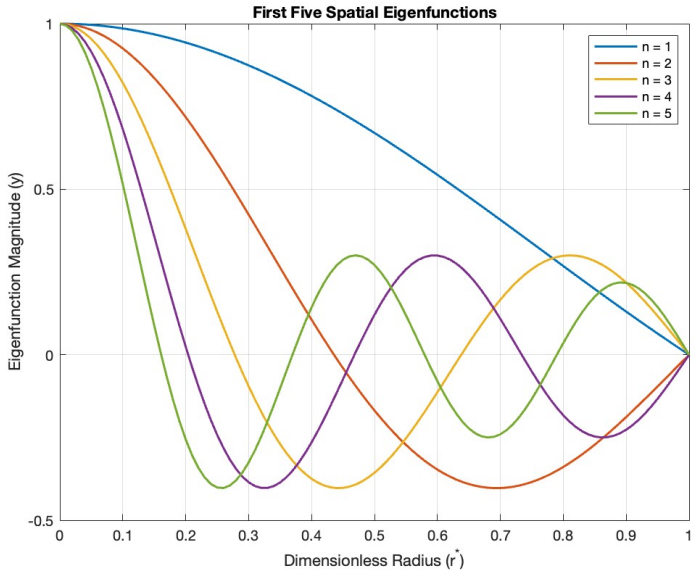


Figure 22.1: The first five spatial eigenfunctions computed numerically for transient tube flow. Note the increase in zero crossings for the higher modes.

When executed, the script outputs the following numeric values for the dominant parameters:

```
firsteigenvals =
    5.7829
   30.4633
   74.8397
  138.8784
  222.5174

firstcoefficients =
   -0.2770
    0.0350
   -0.0113
    0.0054
   -0.0028
```

Comparing these to our analytical roots from Chapter 16, the lead eigenvalue closely approaches $\sigma_1^2 = (2.4048)^2 = 5.7831$, confirming excellent agreement between numerical discretization and analytical theory.

Centerline Velocity Tracking

With our numerical eigenvalues and coefficients in hand, we can track the evolution of the centerline velocity ($r^* = 0$) over time. We can also evaluate how well the simple analytical one-term approximation holds up against the full numerical summation.

```
t = 0.0005:0.001:1;
ucenter = zeros(size(t)); % Pre-allocate tracking array

% Evaluate full numerical series summation across time steps
for i = 1:length(t)
    ucenter(i) = uinf(0) + sum(a .* exp(-lambda * t(i)) .* eigenvecs(1, :)');
end

% Evaluate the one-term dominant mode approximation
ucenter1term = uinf(0) + a(1) * exp(-lambda(1) * t) * eigenvecs(1, 1);

figure(2)
plot(t, ucenter, 'k-', LineWidth=2); hold on;
plot([0, 0.25], [0, 0.25], 'r--'); % Linear short-time asymptote
plot(t, ucenter1term, 'b:', LineWidth=1.5);
xlabel('Dimensionless Time (t^*)')
ylabel('Centerline Velocity (u^*|_{r^*=0})')
legend(['Full Numerical Solution'],['Short-Time Asymptote'],['One-Term Approx'])
grid on
```

As shown in the transient results, the one-term approximation matches the full solution with near-perfect alignment for all times $t^* > 0.1$. This mathematically illustrates why engineers prioritize calculating the lead eigenvalue and coefficient over all others—higher-order terms decay away almost instantly

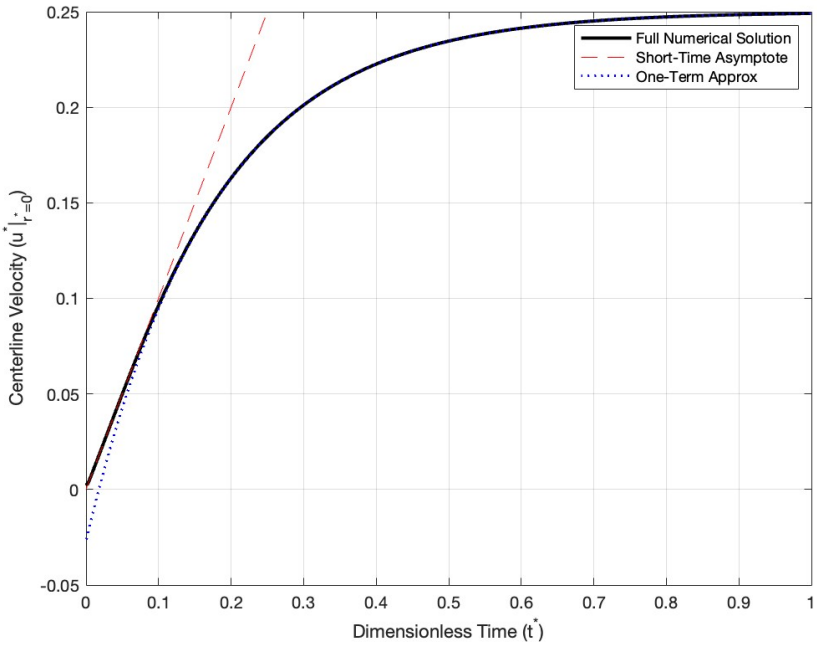


Figure 22.2: Centerline velocity profile development comparing the full numerical calculation against the single lead-mode prediction. The short time asymptote of $u^*|_{r^*=0} = t^*$ is also included.

Velocity Profiles and Numerical Errors

Finally, we can visualize the spatial distribution of fluid velocity across the tube radius at discrete chronological checkpoints.

```
tplot = [0.0002, 0.05, 0.1, 0.2, 0.4, 0.8, 1.6, 3.2]';
uprofile = zeros(length(r), length(tplot));

for j = 1:length(tplot)
    for i = 1:length(r)
      uprofile(i, j) = uinf(r(i)) + sum(a .* exp(-lambda * tplot(j)) .* eigenvecs(i, :)');
    end
end

figure(3)
plot(r, uprofile, LineWidth=1.2); hold on;
plot(r, uinf(r), 'k--', LineWidth=1.5); % Steady asymptote profile
legend([string(tplot); "Steady Asymptote"])
xlabel('Dimensionless Radius (r^*)')
ylabel('Velocity Profile (u^*)')
title('Velocity Distribution at Different Times')
grid on
```

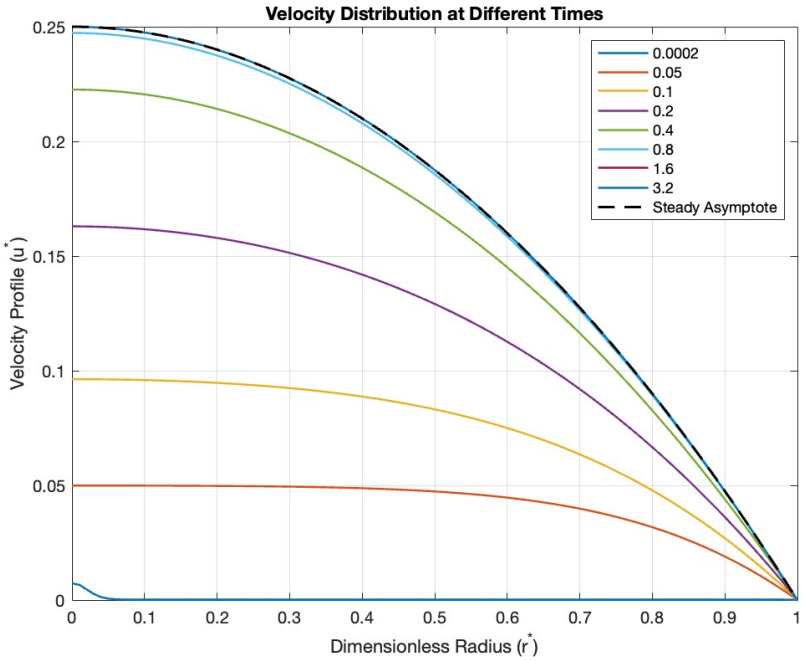


Figure 22.3: Fluid velocity profile development across the radius of the tube at various time snapshots.

A Note on Numerical Ringing: If you inspect the velocity profile at the earliest calculated timestamp ($t^* = 0.0002$), you will see a non-physical wavy oscillation near the centerline origin ($r^* = 0$). While visually similar to the Gibbs Ringing phenomenon which occurs when a discontinuous function is approximated with a Fourier series, this is actually a **numerical artifact**. Because our matrix discretization and numerical integration are inherently $O(1/n^2)$ accurate, they impose a structural noise floor. The true analytical series coefficients decay away very rapidly, but our computed higher-order coefficients eventually hit this $O(1/n^2)$ error limit imposed by the Trapezoidal Rule quadrature and stop decaying. Consequently, the summation becomes polluted with erroneous high-frequency modes. This error is exacerbated at the origin because all the cylindrical eigenfunctions peak and are in phase at $r^* = 0$ (for this problem), causing the noise to constructively compound. At very short times, these artificial high frequencies have not yet been damped out by the transient response. As time advances even slightly, the exponential decay terms ($e^{-\lambda t}$) suppress them, and the ringing vanishes entirely. Away from the centerline the eigenfunctions tend to cancel out, so even though the higher coefficients are numerical noise the magnitude of the error is greatly diminished even at $t^* = 0$.

Practice Problems

Problem 1: Truncation Error and Grid Resolution

In the text, it is stated that the numerical solver has a truncation error of $O(1/n^2)$. Using the fluid startup problem from the example, run the `slsolve` function for grid resolutions of $n = 10, 20, 40, 80,$ and 160.

(a) For each value of n, calculate the absolute error between the numerically computed first eigenvalue (λ_1) and the exact analytical value ($\lambda_{1,\text{exact}} = 5.78318$).

(b) Plot the absolute error against n on a log-log plot. Calculate the slope of the resulting line. Does it confirm the theoretical second-order accuracy?

Problem 2: Convective Boundary Conditions (Robin BC)

Consider a transient heat conduction problem in a 1D Cartesian slab of thickness $L = 1$. The governing Sturm-Liouville parameters are $p(x) = 1$, $q(x) = 0$, and $w(x) = 1$. The left boundary is insulated, resulting in $y'(0) = 0$. The right boundary experiences convective cooling, governed by the Biot number (Bi), such that $y'(1) + Bi\, y(1) = 0$.

(a) Determine the correct `bc` array format for `slsolve` to represent these conditions.

(b) Set $n = 100$. Compute and plot the first spatial eigenfunction for $Bi = 0.1$, $Bi = 1.0$, and $Bi = 10.0$. This would be the dominant term of the decaying solution after some time has passed. Briefly explain how the physical shape of the eigenfunction changes as the convective boundary becomes stronger.

Problem 3: Spatially Variable Properties

The true power of `slsolve.m` or an equivalent numerical scheme is its ability to handle variable physical properties where exact analytical solutions involve Bessel or Airy functions or there is no analytic solution at all. Consider a composite fin where the thermal conductivity increases linearly from the base to the tip: $p(x) = 1 + x$. Let $q(x) = 0$ and $w(x) = 1$. Both the base ($x = 0$) and the tip ($x = 1$) are held at a fixed temperature ($y = 0$).

(a) Use `slsolve` with $n = 200$ to find the first three eigenvalues for this functionally graded material.

(b) Plot the first eigenfunction. Is it perfectly symmetric about $x = 0.5$? Explain physically why it leans toward one side of the domain based on the thermal conductivity.

Solutions to Practice Problems

Solution 1: Truncation Error and Grid Resolution

```
% Setup base parameters
p = @(x) x; q = @(x) 0; w = @(x) x;
bc = [0, 1, 1, 0];
exact_lambda = 5.78318;
n_vals = [10, 20, 40, 80, 160];
errors = zeros(size(n_vals));

% Loop through grid resolutions
for i = 1:length(n_vals)
    [lambda, ~] = slsolve(p, q, w, bc, n_vals(i));
    errors(i) = abs(lambda(1) - exact_lambda);
end

% Calculate slope of log-log plot
coeffs = polyfit(log10(n_vals), log10(errors), 1);
slope = coeffs(1)

figure(1)
loglog(n_vals, errors, 'ko-', LineWidth=1.5, MarkerFaceColor='k')
xlabel('Number of intervals (n)')
ylabel('Absolute Error')
title(["Truncation Error Scaling"; "Slope = " + string(slope)])
grid on
```

Discussion: The code outputs a slope of approximately -2.0. On a log-log plot, a slope of -2 perfectly confirms the theoretical $O(1/n^2)$ second-order truncation error.

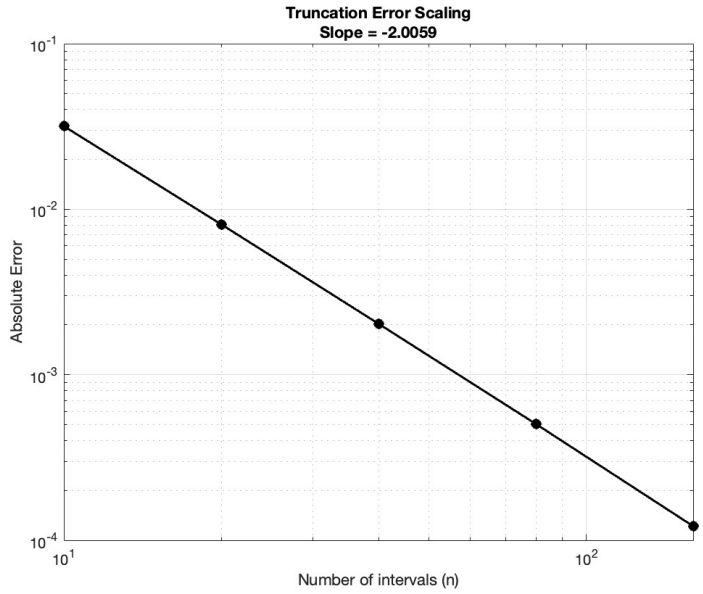


Figure 22.4: Truncation error scaling for the numerical Sturm-Liouville solver, confirming $O(1/n^2)$ accuracy.

Solution 2: Convective Boundary Conditions (Robin BC)

```
p = @(x) 1; q = @(x) 0; w = @(x) 1;
Bi_vals = [0.1, 1.0, 10.0];
n = 100;
figure(2);
for i = 1:length(Bi_vals)
% bc(3)*y(1) + bc(4)*y'(1) = 0 => Bi*y(1) + 1*y'(1) = 0
bc = [0, 1, Bi_vals(i), 1];
[lambda, eigenvecs] = slsolve(p, q, w, bc, n);
plot(linspace(0,1,n+1), eigenvecs(:,1), LineWidth=1.5)
hold on;
end
hold off;
xlabel('Dimensionless Distance (x)')
ylabel('First Eigenfunction Magnitude')
legend("Bi = " + string(Bi_vals))
grid on
```

Discussion: The bc array format is `[0, 1, Bi, 1]`. Physically, as the Biot number increases (stronger convection), the right boundary behaves more like a fixed-temperature (Dirichlet) boundary condition, pinning the eigenfunction closer to zero at $x = 1$. For low Biot number the temperature profile is essentially flat.

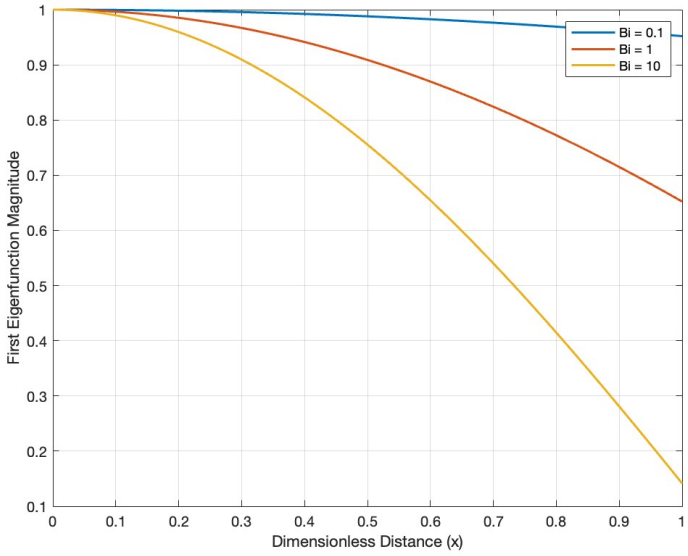


Figure 22.5: Effect of Biot Number on Lead Eigenfunction

Solution 3: Spatially Variable Properties

```
p = @(x) 1 + x; q = @(x) 0; w = @(x) 1;
bc = [1, 0, 1, 0]; % Fixed zero temperature at both ends
n = 200;
[lambda, eigenvecs] = slsolve(p, q, w, bc, n);
first_three_lambdas = lambda(1:3)

figure(4)
plot(linspace(0,1,n+1), eigenvecs(:,1), 'k-', LineWidth=2)
xlabel('Dimensionless Distance (x)')
ylabel('First Eigenfunction')
grid on
```

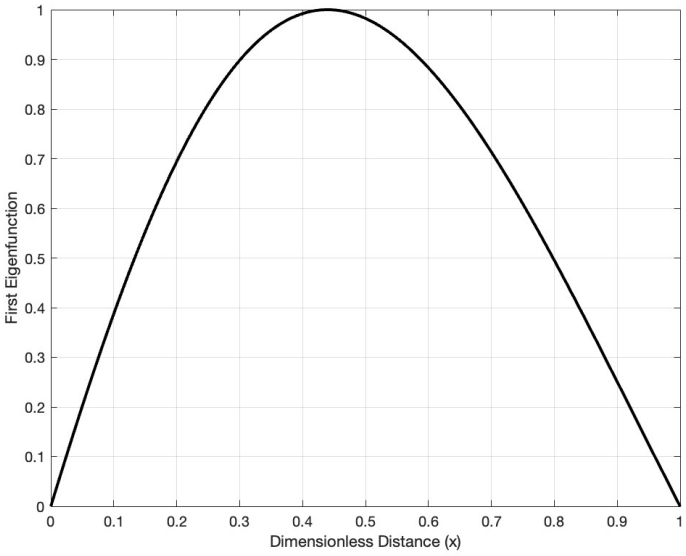


Figure 22.6: Lead Eigenfunction for Non-Constant Conductivity

Discussion:

(a) The first three eigenvalues are 14.3374, 57.4755, and 129.3610

(b) The eigenfunction is not perfectly symmetric; its peak leans toward the left ($x < 0.5$). Physically, the right side has a higher thermal conductivity ($p(x)$ is larger), so heat escapes to the $x = 1$ boundary much easier. The slower heat removal on the left side causes the temperature (or spatial waveform) to "bunch up" or peak closer to $x = 0$.

23 Finite Difference Marching Solutions

Many transport problems cannot be solved analytically, but we still need the answer! As a result, numerical solutions have been developed. Many approaches have been used historically: implicit/explicit finite differences, finite element, spectral methods, lattice Boltzmann, etc. It is a vast literature!

Here, we introduce the simplest method: **Explicit Euler Method / Finite Difference Marching Solutions**. This is useful for many problems, but particularly for parabolic PDEs (first-order derivative in time, second or higher-order derivative in the spatial variable). It is good for both linear and non-linear problems, and it is extremely easy to code up!

Key Idea: Discretize the spatial domain, develop finite difference approximations for the spatial derivatives at each point, and use the Euler Method to march forward in time.

Example: Startup Flow in a Pipe

Let's apply this to the startup flow in a pipe. It is better to do this analytically (as seen in earlier sections), but this numerical method would even work for non-linear, non-Newtonian rheology.

Always render the equations dimensionless (and scale properly) before doing any numerical solution! The dimensionless governing PDE is:

$$\frac{\partial u^*}{\partial t^*} = \frac{1}{r^*}\frac{\partial}{\partial r^*}\left(r^*\frac{\partial u^*}{\partial r^*}\right) + 1$$

Subject to the boundary and initial conditions:

$$\begin{aligned} u^*(1, t^*) &= 0 \quad \text{(No-slip at the wall)} \\ \left.\frac{\partial u^*}{\partial r^*}\right|_{r^*=0} &= 0 \quad \text{(Centerline symmetry)} \\ u^*(r^*, 0) &= 0 \quad \text{(Fluid initially at rest)} \end{aligned}$$

Spatial Discretization

We discretize the domain in r into n intervals of width Δr, resulting in $n + 1$ node locations:

$$r = [0 : \Delta r : 1]$$

Let $u_i = u(r_i)$. Apart from the boundary conditions, the differential equation at any interior node i is:

$$\frac{\partial u_i}{\partial t} = \left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)\right]_{r_i} + 1$$

We need a finite difference representation for the right-hand side. We want a center difference approximation so the truncation error remains $O(\Delta r^2)$. We evaluate the inner derivative at the midpoints of the intervals between nodes:

$$\left.\frac{\partial u}{\partial r}\right|_{r_i+\Delta r/2} \approx \frac{u_{i+1} - u_i}{\Delta r}$$
$$\left.\frac{\partial u}{\partial r}\right|_{r_i-\Delta r/2} \approx \frac{u_i - u_{i-1}}{\Delta r}$$

Applying the outer derivative across these midpoints yields:

$$\left[\frac{1}{r}\frac{\partial}{\partial r}\left(r\frac{\partial u}{\partial r}\right)\right]_{r_i} \approx \frac{1}{r_i}\left[\frac{\left(r_i + \frac{\Delta r}{2}\right)\frac{u_{i+1}-u_i}{\Delta r} - \left(r_i - \frac{\Delta r}{2}\right)\frac{u_i-u_{i-1}}{\Delta r}}{\Delta r}\right]$$

Marching Forward in Time

Marching forward is easy using the explicit Euler Method. Let u_i^k be the estimate of the velocity at node i at time step t_k. The velocity at the next time step is simply:

$$u_i^{k+1} = u_i^k + \Delta t\left\{\frac{1}{r_i \Delta r^2}\left[\left(r_i + \frac{\Delta r}{2}\right)(u_{i+1}^k - u_i^k) - \left(r_i - \frac{\Delta r}{2}\right)(u_i^k - u_{i-1}^k)\right] + 1\right\}$$

This works for all interior nodes, but not for the first and last nodes! Those are determined by our boundary conditions. After updating the interior nodes, we apply the BCs:

Outer Edge ($r = 1$): The velocity is identically zero.

$$u_{n+1}^{k+1} = 0$$

Inner Edge ($r = 0$): The derivative must be zero. To maintain $O(\Delta r^2)$ accuracy, we estimate the derivative at the midpoint $\Delta r/2$ and shift it back to $r = 0$ using a Taylor series approximation incorporating the second derivative:

$$0 = \left.\frac{\partial u}{\partial r}\right|_{r=0} \approx \underbrace{\frac{u_2 - u_1}{\Delta r}}_{\text{Deriv est. at } \Delta r/2} \underbrace{- \frac{\Delta r}{2}\left(\frac{u_3 + u_1 - 2u_2}{\Delta r^2}\right)}_{\text{Shift back to } r=0}$$

Multiplying through by Δr and rearranging:

$$0 = (u_2 - u_1) - \frac{1}{2}(u_3 + u_1 - 2u_2)$$
$$0 = 2u_2 - \frac{3}{2}u_1 - \frac{1}{2}u_3$$
$$\frac{3}{2}u_1 = 2u_2 - \frac{1}{2}u_3$$

Which yields the explicit update for the centerline node:

$$u_1^{k+1} = \frac{1}{3}(4u_2^{k+1} - u_3^{k+1})$$

Numerical Stability

Like any explicit numerical integration method, this approach is numerically unstable if Δt is too large (the eigenvalues of the Jacobian are large and negative, leading the system of equations to be very stiff). For this method, there is a von Neumann condition for stability that strictly limits the time step size relative to the spatial grid:

$$\Delta t \lesssim \frac{1}{2}\Delta r^2$$

Anything bigger than this blows up! This requires a severely small Δt when using a fine spatial grid (small Δr). You can avoid this restriction by using implicit methods (e.g., Crank-Nicolson), but these require setting up and solving a matrix system at every time step and are much harder to code up. For highly complicated problems, it is best to use canned PDE solvers—they have a bit of a learning curve but are incredibly useful.

MATLAB Implementation

The entire explicit marching scheme can be coded up in just a few lines of MATLAB. The following script calculates the transient velocity profile and generates a movie comparing the numerical results to the exact analytical series solution. Note that the actual numerical execution of this technique is very fast - if you comment out the graphics which dominates the execution time. Because the time step is so small you usually adjust the code so that it only plots things up at specific intervals or times. At least some speedup can be obtained if you use the `drawnow limitrate` command at the bottom of the loop, which cuts the refresh rate down to 20 times per second, important for extremely small time steps and finer discretization.

```
% This script produces a movie of startup flow in a pipe. It uses the Euler
% method marching forward in time.

n = 20; % The spatial discretization.
dr = 1/n;
r = (0:dr:1);
u = zeros(size(r));
udot = zeros(size(r));

i = 2:n; % The interior nodes
dt = 0.49*dr^2; % We choose a time discretization for stability.
% Increase the coefficient to 0.51 and watch it blow up!

t = 0;
```

```
while t < 0.5
    t = t + dt;

    % The derivative for the interior nodes
    udot(i) = 1 + ((u(i+1) - u(i))/dr .* (r(i) + dr/2) ...
        - (u(i) - u(i-1))/dr .* (r(i) - dr/2)) / dr ./ r(i);

    u(i) = u(i) + udot(i)*dt; % We update interior nodes

    u(n+1) = 0; % The velocity at the outer wall remains zero.

    % We use a zero derivative condition at the center.
    u(1) = 2/3 * (2*u(2) - 0.5*u(3));

    figure(1)
    plot(u, r, 'x', 0.25*(1-r.^2), r, [0 0], [0 1])
    axis([0 0.3 0 1])
    xlabel('Dimensionless Velocity (u^*)')
    ylabel('Dimensionless Radius (r^*)')
    title(['Velocity Profile in a Pipe for t* = ', num2str(t)])

    % Let's compare this to the exact solution! We have the eigenvalues
    % (roots of J0, we keep ten):
    sigma = [2.40482555769577;  5.52007811028631;  8.65372791291101;
            11.79153443901428; 14.93091770848779; 18.07106396791092;
            21.21163662987925; 24.35247153074930; 27.49347913204025;
            30.63460646843198];

    % Which yields the coefficients:
    an = -2.0 ./ sigma.^3.0 ./ besselj(1, sigma);

    % and the solution:
   uexact = 0.25*(1-r.^2) + (an .* exp(-sigma.^2 * t))' * besselj(0, sigma * r);

    % and we plot it up:
    hold on
    plot(uexact, r, 'g')
    hold off

    % We can also plot up a movie of the deviation:
    figure(2)
    plot(r, u - uexact)
    xlabel('Dimensionless Radius (r^*)')
    ylabel('Deviation')
   title(['Deviation from exact solution (10 eigenvalues) for t* = ', num2str(t)])
    drawnow limitrate
end
```

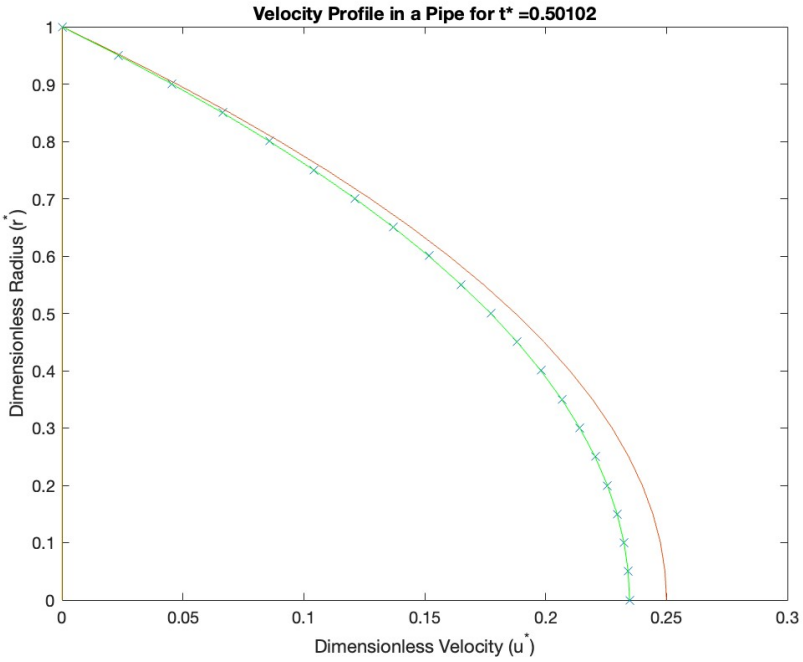


Figure 23.1: Numerical finite-difference solution (symbols) compared with the analytical transient solution (solid line). The steady-state Poiseuille profile is also shown.

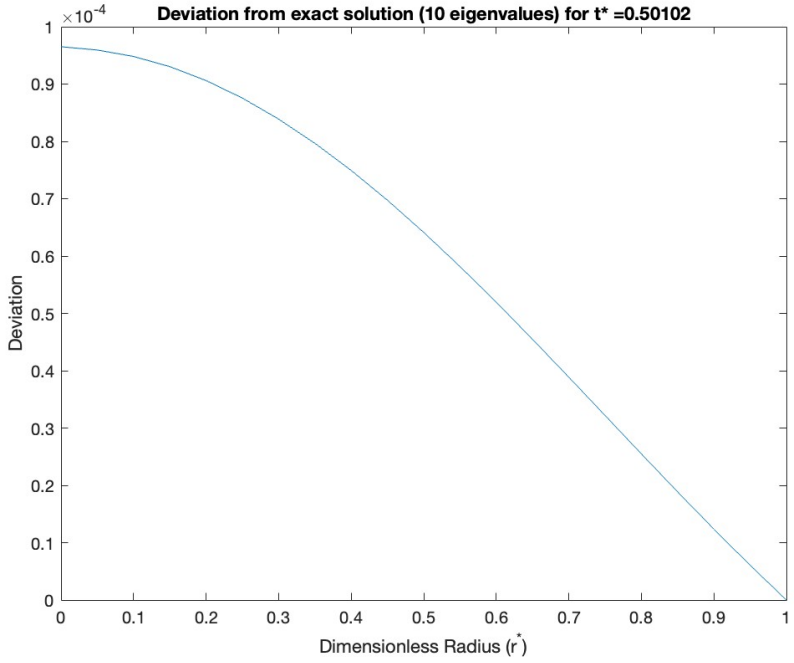


Figure 23.2: The absolute deviation between the explicit finite difference calculation and the 10-term analytical series solution at $t^* \approx 0.5$.

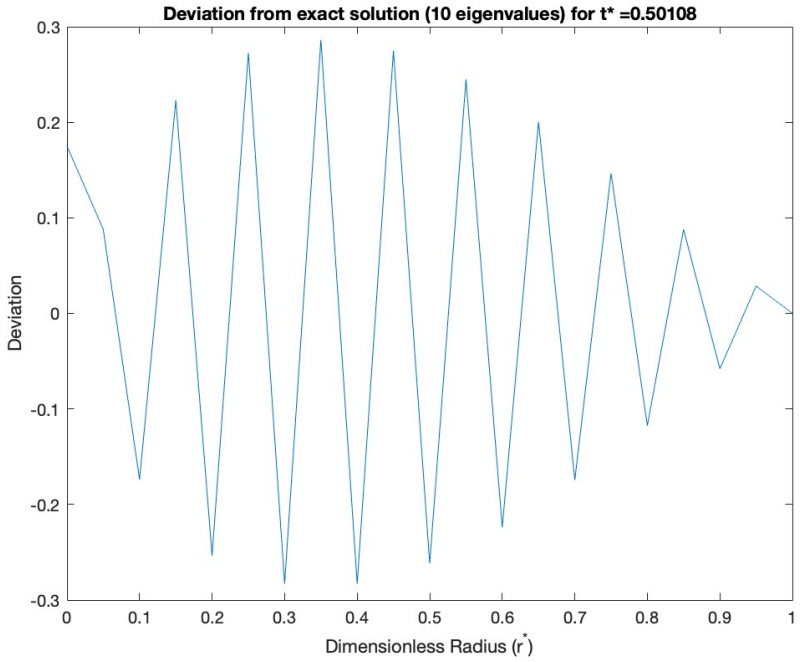


Figure 23.3: The absolute deviation between the explicit finite difference calculation and the 10-term analytical series solution at $t^* \approx 0.5$ with a time step just above stability limits.

Practice Problems

Problem 1: Reaction-Diffusion with Non-Linear Kinetics

The true power of numerical finite difference methods is their ability to handle non-linear differential equations where exact analytical solutions are mathematically impossible.

Consider a long cylindrical catalyst pellet of radius R bathed in a chemical reactant. The reactant diffuses radially inward and is consumed by a chemical reaction. Because the catalyst active sites can saturate, the reaction follows non-linear Michaelis-Menten kinetics.

The dimensionless governing PDE for the concentration profile $C^*(r^*, t^*)$ is:

$$\frac{\partial C^*}{\partial t^*} = \frac{1}{r^*}\frac{\partial}{\partial r^*}\left(r^*\frac{\partial C^*}{\partial r^*}\right) - \frac{\Phi^2 C^*}{1+\beta C^*}$$

where Φ^2 is the Thiele modulus (ratio of reaction rate to diffusion rate) and β is the dimensionless saturation constant.

The boundary and initial conditions are:

$$
\begin{aligned}
C^*(1, t^*) &= 1 \quad \text{(Constant surface concentration)} \\
\left.\frac{\partial C^*}{\partial r^*}\right|_{r^*=0} &= 0 \quad \text{(Centerline symmetry)} \\
C^*(r^*, 0) &= 0 \quad \text{(Pellet initially empty)}
\end{aligned}
$$

(a) Modify the explicit marching script developed in this chapter to solve this non-linear PDE. Set $\Phi^2 = 10$, $\beta = 2$, and $n = 20$.

(b) Run the simulation until the concentration profile reaches a steady state (around $t^* = 0.5$). Plot the transient concentration profiles over time.

(c) Explain physically why the concentration does not reach 1.0 at the centerline ($r^* = 0$).

Note that while you can "pretty it up", this intrinsically requires modification of only a few lines of the provided code!

Solutions to Practice Problems

Solution 1: Reaction-Diffusion with Non-Linear Kinetics

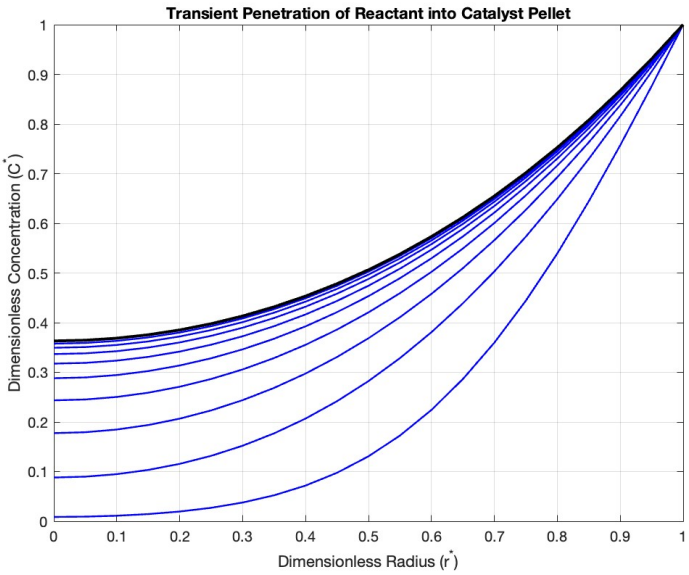


Figure 23.4: Concentration profiles marching toward a non-linear steady state at dimensionless time intervals of 0.05.

Discussion:

(a,b) Because the explicit Euler method evaluates the right-hand side using the **current** known time step, handling the non-linear reaction requires no special algebraic maneuvering. We simply substitute the non-linear term into the `Cdot` evaluation. Finding a numerical solution to a PDE that has no simple analytical solution required changing only two lines of code.

(c) The concentration at the centerline does not reach 1.0 because the reactant is being consumed by the catalyst as it diffuses inward. At steady state, a balance is struck: the rate at which molecules diffuse toward the center exactly matches the rate at which they are destroyed by the reaction along the way. Because $\Phi^2 = 10$ is relatively high, the reaction is fast enough to create a permanent "depletion zone" near the core. This is essentially the problem investigated in the classic work of Ernest Wilhelm Thiele, although his original analysis assumed first-order kinetics and therefore led to a solvable linear differential equation.

```
% Explicit marching solution for Michaelis-Menten Reaction-Diffusion
n = 20;
dr = 1/n;
r = (0:dr:1);
C = zeros(size(r));
Cdot = zeros(size(r));

i = 2:n; % Interior nodes
dt = 0.2*dr^2; % Safe time step

% Reaction parameters
Phi2 = 10;
beta = 2;

t = 0;

while t < 0.5
    t = t + dt;

    % The spatial derivative + non-linear reaction sink
    Cdot(i) = ((C(i+1) - C(i))/dr .* (r(i) + dr/2) ...
        - (C(i) - C(i-1))/dr .* (r(i) - dr/2)) / dr ./ r(i) ...
        - (Phi2 * C(i)) ./ (1 + beta * C(i));

    C(i) = C(i) + Cdot(i)*dt; % Update interior nodes

    C(n+1) = 1; % Boundary condition at the outer wall

    % Symmetry boundary condition at the center
    C(1) = 2/3 * (2*C(2) - 0.5*C(3));

    % Plot the evolving profile at specific intervals
    if mod(round(t/dt), 100) == 0
        figure(1)
        plot(r, C, 'b-', LineWidth=1.2)
        title(['Concentration Profile for t* = ',num2str(t)])
        hold on
    end
end

plot(r, C, 'k-', LineWidth=2) % Final steady-state profile
xlabel('Dimensionless Radius (r^*)')
ylabel('Dimensionless Concentration (C^*)')
title('Transient Penetration of Reactant into Catalyst Pellet')
grid on; hold off;
```

24 Monte Carlo Solutions to Diffusion Equations

A completely different way to solve transient diffusion problems is via **Monte Carlo Simulations**. It is particularly useful for linear problems with more complex domains. While it can be computationally expensive, it is very easy to code up! It is also a very good way of visualizing the diffusion process.

Key Idea: Diffusion of mass or energy can be approximated by a normally distributed random walk with a step size having a standard deviation of $(2D\Delta t)^{1/2}$. Thus, you just follow the motion of "tracers" (representing discrete packets of mass or energy) and give them a random kick at each time step!

The Computational Cost: While the simulation is simple, the accuracy requires a *very* large number of tracers. That is because the accuracy is governed by statistics rather than algorithm error: to decrease the error by a factor of 10 you require 100 times more tracers. On the other hand, this also applies to very high dimensionality calculations where the ordinary domain discretization required by conventional solution approaches may be impractical.

Example: The Quenched Cube

Let's look at a simple example: a cube with sides of length $2a$ and a uniform initial temperature T_0 is suddenly cooled with a constant surface temperature T_s. What is the average temperature of the cube as a function of time?

The governing partial differential equation is the 3D heat equation:

$$\frac{\partial T}{\partial t} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} \right)$$

Subject to the initial and boundary conditions:

$$\begin{aligned} T|_{t=0} &= T_0 \\ T|_{x=\pm a} = T|_{y=\pm a} &= T|_{z=\pm a} = T_s \end{aligned}$$

Let's render the equations dimensionless. We define our scaled variables as:

$$T^* = \frac{T - T_s}{T_0 - T_s}, \quad x^* = \frac{x}{a}, \quad y^* = \frac{y}{a}, \quad z^* = \frac{z}{a}, \quad t^* = \frac{t}{t_c}$$

Plugging these into the governing equation yields:

$$\frac{(T_0 - T_s)}{t_c}\frac{\partial T^*}{\partial t^*} = \frac{\alpha(T_0 - T_s)}{a^2}\left(\frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}} + \frac{\partial^2 T^*}{\partial z^{*2}}\right)$$

Dividing out the dimensional temperature scale gives:

$$\left[\frac{a^2}{\alpha t_c}\right]\frac{\partial T^*}{\partial t^*} = \nabla^{*2} T^*$$

To eliminate the bracketed coefficient, we define our characteristic diffusion time as $t_c = \dfrac{a^2}{\alpha}$. This leaves us with the fully dimensionless PDE:

$$\frac{\partial T^*}{\partial t^*} = \frac{\partial^2 T^*}{\partial x^{*2}} + \frac{\partial^2 T^*}{\partial y^{*2}} + \frac{\partial^2 T^*}{\partial z^{*2}}$$

With the updated initial and boundary conditions:

$$T^*|_{t^*=0} = 1$$
$$T^*|_{x^*=\pm 1} = T^*|_{y^*=\pm 1} = T^*|_{z^*=\pm 1} = 0$$

Solving via a Random Walk

We can simplify the simulation domain by utilizing the physical symmetry of the cube. We only need to simulate the positive octant ($0 \le x^*, y^*, z^* \le 1$), which enforces zero-flux symmetry boundary conditions at the origin planes:

$$\left.\frac{\partial T^*}{\partial x^*}\right|_{x^*=0} = \left.\frac{\partial T^*}{\partial y^*}\right|_{y^*=0} = \left.\frac{\partial T^*}{\partial z^*}\right|_{z^*=0} = 0$$

To solve this using the Monte Carlo method, we distribute N tracers uniformly over the domain $0 < x^*, y^*, z^* < 1$. At each time step Δt^*, every tracer gets a random dimensionless "kick" in the x, y, and z directions with a standard deviation of $(2\Delta t^*)^{1/2}$. Note that the random motion in each direction must be independent, and for best results it is useful to employ a normally distributed random number with the required standard deviation. In MATLAB, for example, this would be `(2*dt)^.5*randn(N,1)` for a column vector of length N.

Handling the boundary conditions with discrete tracers is highly intuitive:

- **Symmetry Condition ($x^*, y^*, z^* = 0$):** Simulated by simple reflection. If a tracer wanders into the negative domain, we simply take its absolute value (e.g., $x^* = |x^*|$).
- **Absorbing Condition ($x^*, y^*, z^* = 1$):** This is even easier! You simply remove any tracers that exceed 1.0 in any direction.

Because the dimensionless temperature represents the fraction of initial thermal energy remaining, the average temperature of the cube at any given time is simply the number of tracers left inside the domain divided by the initial number N.

MATLAB Implementation

The following script implements this Monte Carlo integration. Note that the step size in time must be small enough that the spatial jump $(2\Delta t^*)^{1/2}$ remains small with respect to the overall domain size.

```
%% Simulation of a Quenched Cube
% In this simulation we use Monte Carlo integration to determine the
% average temperature of a cube as a function of time.  Its initial
% temperature is 1 and the surface temperature is zero.  We start with N
% tracers distributed over the positive chunk of the cube, reflect at the
% symmetry planes, and remove tracers which exceed 1 in any direction.  The
% average temperature is just the number of tracers remaining in the
% domain!  Note that the step size in time must be small enough that the
% (2*dt)^1/2 jump is small with respect to the domain.

N = 10000; % We start with lots of tracers

dt = 0.00005; % This yields a random walk step of ~1% of the domain

x = rand(N,1); % The initial values of x, y, and z
y = rand(N,1);
z = rand(N,1);

tfinal = .5; % How long we run it for.

tall = 0:dt:tfinal;

t = 0;
i = 1; % A counter.

% This returns the maximum of x, y, and z for each tracer
isout = max(max(x,y),z);
ikeep = find(isout<1);
nleft = N;

Tavgkeep = zeros(size(tall));
Tavg = 1; % Our initial dimensionless temperature
Tavgkeep(1) = Tavg;

while nleft > 0 && t < tfinal
    i = i+1;
    t = tall(i);
```

```
    % Apply the random Brownian kick
    x(ikeep) = x(ikeep) + (2*dt)^.5 * randn(nleft,1);
    y(ikeep) = y(ikeep) + (2*dt)^.5 * randn(nleft,1);
    z(ikeep) = z(ikeep) + (2*dt)^.5 * randn(nleft,1);

    % Enforce symmetry boundary conditions via reflection
    x = abs(x);
    y = abs(y);
    z = abs(z);

    % Enforce absorbing boundary conditions at the walls
    isout = max(max(x,y),z);
    ikeep = find(isout<1);

    % Calculate the new average temperature
    nleft = length(ikeep);
    Tavgkeep(i) = nleft/N;
end

figure(1)
plot(tall, Tavgkeep, 'LineWidth', 1.5)
xlabel('Dimensionless Time (t^*)')
ylabel('Average Dimensionless Temperature')
title('Monte Carlo Simulation: Quenched Cube')
grid on
```

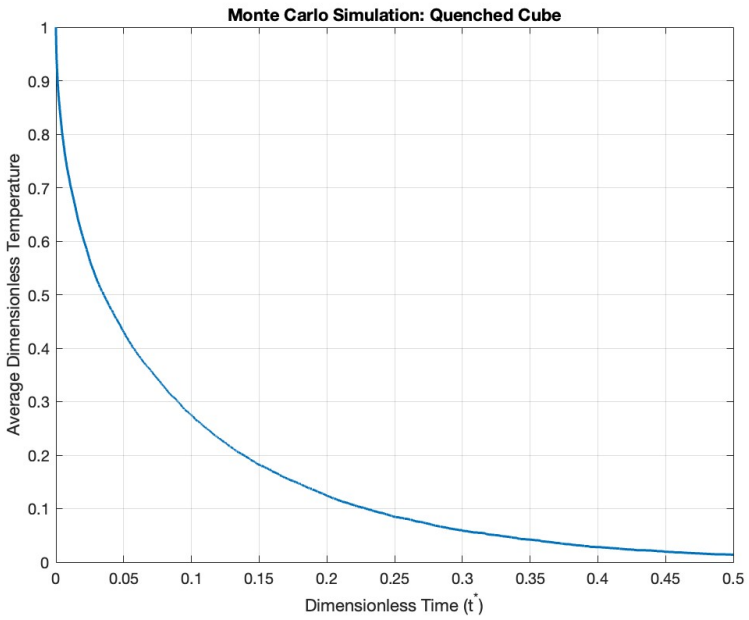


Figure 24.1: Monte Carlo simulation of conduction from a cube.

Extensions of the Method

You can easily extend this methodology to other boundary conditions and physical phenomena:

- **Partial Flux / Robin Conditions:** Simulated via partial reflection/absorption probabilities when a tracer hits a wall.
- **Reaction Sources:** Autocatalytic sources (the "bomb" problem) or first-order sinks can be modeled by having tracers clone themselves or randomly die off at certain rates.
- **Convective Diffusion:** Extremely easy to implement! You simply add a deterministic convective displacement ($u_x \Delta t$) to the random Brownian kick at each time step.

Non-constant (position-dependent) diffusivities are a bit trickier, as you have to mathematically bias your random walk to get it right. This is usually done by adding in a convective velocity equal to the gradient in the diffusivity. For problems such as simulation of an autocatalytic reactor you also have to be careful not to vary array sizes at every time step (e.g., if you are adding particles to track), as this is a memory allocation issue. Still, it is a fairly easy way to solve transient diffusion problems in even complicated geometric domains!

Convective Diffusion: Taylor Dispersion

Where this sort of approach is particularly useful is for the analysis of separation and dispersion in convective techniques such as Field-Flow Fractionation (FFF). Basically, you just follow solute molecules (or particles) in the flow and an applied external field (e.g., electric, gravity, cross-flow, etc.) and add in the Brownian kick. It is highly versatile and very easy to simulate!

Related to this we can simulate convective diffusion (which leads to dispersion) by tracking the motion of tracers undergoing a random walk superimposed on a deterministic velocity field. At each time step, the tracers take a random Brownian step with a standard deviation of $(2D\Delta t)^{1/2}$ and are simultaneously advected by the local fluid velocity.

Effectively, we are integrating the Langevin equation in time and tracking the statistics of the tracer cloud. The growth rate of the variance of the tracer positions in the flow direction directly yields the effective dispersion coefficient (the Taylor dispersivity). While the asymptotic dispersivity can be calculated via other techniques such as the Method of Moments, the simulation approach allows you to see how rapidly the steady state dispersivity is reached - essentially exactly what the solute molecules experience as they flow through a channel of arbitrary cross section. You do, however, have to know what the velocity profile is given by.

For an infinite parallel-plate channel of half-height b, the dimensionless velocity profile $u^* = u_z/U$ is a simple parabola:

$$u_z^* = \frac{3}{2}\left(1 - y^{*2}\right)$$

where $y^* = y/b$. However, for a rectangular channel with side walls at $x = \pm a$, the flow is more complex. As described earlier, we can solve for the exact velocity field using separation of variables by assuming the solution is the infinite parallel-plate parabola plus a 2D deficit function $u_d^*(x^*, y^*)$:

$$u^* = \frac{3}{2}(1 - y^{*2}) + u_d^*(x^*, y^*)$$

where $x^* = x/b$. Because the steady, fully-developed Navier-Stokes equation dictates that $\nabla^2 u^* =$ constant, and the parabola already satisfies the constant pressure gradient, the deficit function must satisfy Laplace's equation:

$$\frac{\partial^2 u_d^*}{\partial x^{*2}} + \frac{\partial^2 u_d^*}{\partial y^{*2}} = 0$$

Subject to the symmetry and no-slip boundary conditions:

$$\left.\frac{\partial u_d^*}{\partial y^*}\right|_{y^*=0} = 0, \quad u_d^*|_{y^*=1} = 0$$
$$\left.\frac{\partial u_d^*}{\partial x^*}\right|_{x^*=0} = 0, \quad u_d^*|_{x^*=a/b} = -\frac{3}{2}(1 - y^{*2})$$

Solving this via separation of variables yields the exact analytical velocity profile:

$$u^* = \frac{3}{2}(1 - y^{*2}) + \sum_{n=1}^{\infty} \frac{6(-1)^n}{\sigma_n^3} \frac{\cosh(\sigma_n x^*)}{\cosh\left(\sigma_n \frac{a}{b}\right)} \cos(\sigma_n y^*)$$

where the eigenvalues are $\sigma_n = \left(n - \frac{1}{2}\right)\pi$.

While exact, computing this infinite series at every single time step for 10,000 Monte Carlo tracers would be computationally prohibitive. Since the flow field is not affected by the simulation, however, a more practical approach would be to evaluate it once on a fine-mesh grid and interpolate at each time step. For large aspect ratios (which are of primary interest) there is an even simpler approach: we can approximate the slowdown caused by the side walls by calculating the total velocity deficit and representing it by an equivalent stagnant layer of thickness δ^*.

$$\delta^* = \int_0^1 \int_0^{a/b} u_d^* \, dx^* \, dy^* = \sum_{n=1}^{\infty} \frac{6}{\sigma_n^5} \left[\tanh\left(\sigma_n \frac{a}{b}\right)\right] \approx 0.63025 \approx \frac{5}{8} \; , a/b >> 1$$

Thus, for our simulation, a surprisingly effective approximation is to represent the side-wall influence by just assuming a stagnant layer of width $5/8$ next to each side where $u_z^* = 0$. It is incredibly easy to add this into our Monte Carlo simulation!

There is an additional effect of the slowdown at the side walls, in that the average velocity of the fluid in the channel is no longer U, but rather is $U^* = U\dfrac{1}{1 - \delta^* \frac{b}{a}} \approx \dfrac{U}{1 - \frac{5}{8}\frac{b}{a}}$. The dispersivity scales as the square of this velocity, so this is a significant correction for finite a/b.

```
%% MC Simulation of Taylor Dispersion
% We construct a simple Monte Carlo simulation of dispersion in a channel,
% first with no effect of the side walls, and then with the no-slip side
% walls causing a velocity reduction.  The effect of the side walls is
% approximated by causing all tracers within 5/8 of the edges to have a
% velocity of zero.  By symmetry, we shall only consider the top right
% quarter of the rectangular channel.  It has a dimensionless half-height
% of 1 and a dimensionless half width of ab (e.g., a/b).  To include the side
% walls remove the comment (%) on the two "find" commands.

n = 1000;
ab = 5;

% We distribute them randomly.
x = ab*rand(n,1);
y = rand(n,1);
z = zeros(n,1);

dt = 0.00005; % Our time step
tquit = 5;
t = [dt:dt:tquit]'; % Our times

uz = @(y) 1.5*(1-y.^2); % our velocity

varz = zeros(size(t)); % we keep track of the variance.

for i = 1:tquit/dt-1
    dz1 = uz(y);

%   k = find(x>ab-5/8); dz1(k)=0; % The side wall fix...

    x = x + randn(n,1)*(2*dt)^.5;
    x = abs(x); % We reflect in the x-direction at the center
    x = min(x, 2*ab-x); % We reflect in from the right side

    y = y + randn(n,1)*(2*dt)^.5;
    y = abs(y);
    y = min(y, 2-y); % We reflect in from the top

    dz2 = uz(y);

%   k = find(x>ab-5/8); dz2(k)=0; % The side wall again.

    z = z + (dz1+dz2)/2*dt; % We update the velocity via Trapezoidal Rule
    % Note: higher accuracy requires separate calculation of reflected points

    varz(i) = var(z);

end
```

```
figure(1)
subplot(2,2,1), plot(x,y,'o')
xlabel('x')
ylabel('y')
title('x-y distribution')

subplot(2,2,2), plot(x,z,'o')
xlabel('x')
ylabel('z')
title('x-z distribution')

subplot(2,2,3), plot(y,z,'o')
xlabel('y')
ylabel('z')
title('y-z distribution')

subplot(2,2,4), plot(t(1:i), varz(1:i), t(1:i), 4/105*t(1:i))
xlabel('t')
ylabel('z variance')
title('variance in the z direction')
legend(['MC simulation'] ,['TA dispersion calculation'],...
    ['Location'],['northwest'])
```

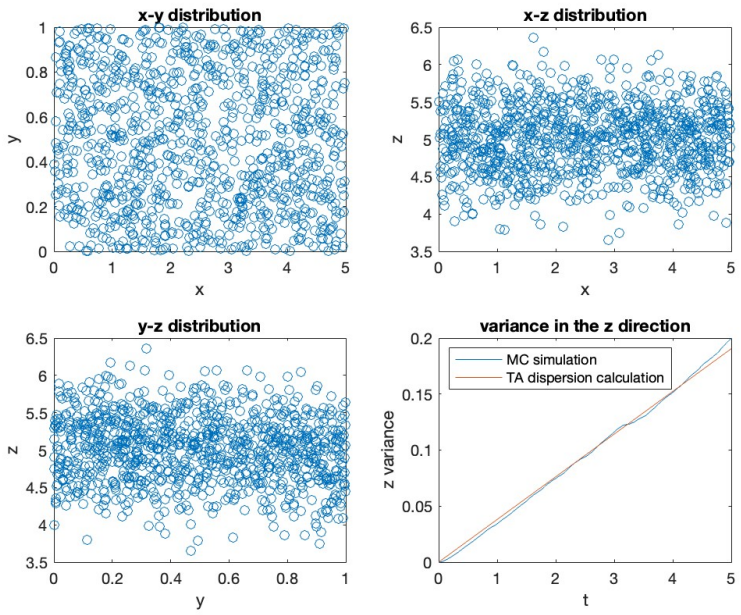


Figure 24.2: Monte Carlo simulation of Taylor Dispersion for plane-Poiseuille flow. The variance in the axial (z) direction grows linearly with time, confirming the relationship $2D_{\text{eff}}t$, plotted against the exact Taylor-Aris dispersion calculation.

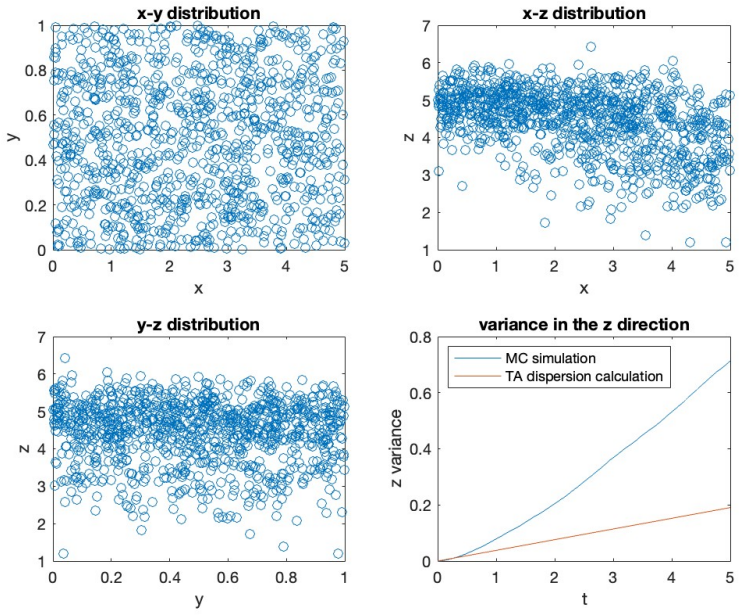


Figure 24.3: Monte Carlo simulation of Taylor Dispersion with aspect ratio of 5. The variance in the axial (z) direction grows linearly with time after an initial transient, confirming the relationship $2D_{\text{eff}}t$, plotted against the exact Taylor-Aris dispersion calculation for plane-Poiseuille flow. Note the much larger dispersivity as well as the non-uniform tracer profile in the x-z plane due to the slowdown at the side wall.

Practice Problems

Problem 1: The Aspect Ratio Effect in Taylor Dispersion

It is highly tempting to assume that if a rectangular channel is very wide (e.g., aspect ratio $a/b \gg 1$), the side walls are too far away to matter, and the Taylor dispersion coefficient D_{eff} will simply equal the infinite plane-Poiseuille limit:

$$D_{\text{eff, 1D}} = \frac{2}{105}\frac{U^2 b^2}{D}$$

Using the provided Monte Carlo script, let's test this assumption.

(a) Run the simulation with the side-wall effect enabled for aspect ratios of $a/b = 1$, 2, 3, 5, and 10.

(b) For each run use the last third of the points to extract the slope of the axial variance versus time. Divide this slope by 2 to get the simulated D_{eff}. Don't forget to renormalize the diffusivity by the factor $(U^*/U)^2$ to account for the reduction of the average velocity by the side walls. Using this adjusted value, calculate the ratio $D_{\text{eff}}/D_{\text{eff, 1D}}$. Note that at large aspect ratios you need to run the simulation for a larger time in order for the dispersivity to reach its asymptotic value. This is because solute molecules have to diffuse across the largest length scale to reach steady state, and the equilibration time will thus scale as a^2/b^2. While the dimensionless time

of 5 is fine for an aspect ratio of 1, the time needs to be increased to 50 or so for an aspect ratio of 10.

(c) The Monte Carlo simulation is also random - it will produce a slightly different value every time you run it! To characterize this, run it five times for each aspect ratio and calculate both the mean and the random error in the mean. Plot up these mean values with error bars as a function of the aspect ratio.

(d) You should find that as a/b gets large, the dispersion does *not* approach the 1D limit of 1.0, but rather asymptotes to a value near 7.95. Explain physically why the presence of the side walls cause a total dispersion nearly 8 times that expected for an infinite parallel plate, no matter how far away they are (and thus are of small relative volume as well). *Hint: Think about the diffusion timescale required to sample the entire velocity profile. This is the core of the Taylor dispersion mechanism important in many systems.*

Problem 2: The Square Channel Limit and "Vibe Coding"

While you should have found that the Monte Carlo simulation worked very well for large aspect ratio channels, it is actually quite far off from the expected diffusivity ratio of approximately 1.7 for square channels. This is because the simplified velocity profile of a stagnant side layer is a poor approximation for a square! In this problem, you will use a little "vibe coding" to analyze the square channel limit.

(a) It is useful to have a simplified analytical expression for the exact velocity profile to avoid computing an infinite series at every time step. Input the exact series solution into your favorite AI and ask it to generate a simplified 2D approximate expression (e.g., a polynomial) for the velocity profile of a square channel. You must specify to the AI that the function needs to satisfy the no-slip boundary conditions at all walls and maintain the proper physical symmetry. To ensure the dispersivity is accurate, instruct the AI to formulate a two-parameter trial function and solve for those parameters such that the approximation preserves the exact average velocity and spatial variance of the true infinite series solution.

(b) AI models can sometimes hallucinate mathematical constraints. To verify the AI's math, have the AI write a short program (e.g., in Python or MATLAB) that numerically calculates the average velocity and spatial variance for *both* the exact infinite series solution and the new approximate velocity function. Run the script and compare the results. Does the approximation successfully match the exact values?

(c) Once you have a valid velocity profile, normalize it by its average value and modify the Monte Carlo simulation (either yourself or by having the AI do it for you) to calculate the simulated D_{eff} for a square channel. Run the simulation enough times to report the 95% confidence interval of the dispersivity ratio. How close do you get to 1.7?

Solutions to Practice Problems

Solution 1: The Aspect Ratio Effect in Taylor Dispersion

The code which can be used to get the slope is below. Either append it to the simulation, or run it afterwards, for each choice of a/b.

```
% --- Aspect Ratio Dispersion Analysis ---

% Isolate the valid data generated during the loop
t_data = t(1:i);
varz_data = varz(1:i);

% Find the point 2/3 along to ignore the initial transient
mid = floor(length(t_data) * 2 / 3);

% Extract the last third of the data
t_asymp = t_data(mid:end);
varz_asymp = varz_data(mid:end);

% Fit a line (1st-order polynomial) to the asymptotic variance
% polyfit returns [slope, intercept]
p = polyfit(t_asymp, varz_asymp, 1);
slope = p(1);

% Variance grows as 2 * D_eff * t, so D_eff is half the slope
D_eff_sim = slope / 2;

% Correct for the reduction in the average velocity
D_eff_sim = D_eff_sim / (1-5/8/ab)^2;

% Theoretical 1D limit for plane-Poiseuille flow
D_eff_1D = 2 / 105;

% Calculate the ratio
ratio = D_eff_sim / D_eff_1D;

% Display the results
fprintf('Simulated D_eff: %f\n', D_eff_sim);
fprintf('1D Limit D_eff:  %f\n', D_eff_1D);
fprintf('Ratio (Sim / 1D): %f\n', ratio);
```

Discussion (Part d): This is a classic, counter-intuitive result in transport phenomena. In an infinite parallel-plate channel, a tracer only has to diffuse across the narrow height $2b$ to sample all the velocities and average out the dispersion.

However, when side walls are present, they create a region of zero velocity. For the solute cloud to truly average out, tracers must now diffuse across the **entire width** of the channel ($2a$) to swap

between the fast center streamlines and the slow wall streamlines. Because transverse diffusion across a wide channel takes much longer than vertical diffusion across a narrow gap, the tracers near the walls fall behind the rest of the cloud before they can diffuse away. This stretching of the solute cloud results in a dispersion coefficient that is in total ~ 7.95 times larger than the effect of diffusion in the thin direction alone. This asymptotic limit was analyzed by Chatwin & Sullivan (1982) in a classic study of dispersion in rectangular channels.

Interestingly, the calculated value for aspect ratio of 1 (a square channel) does not approach the literature value of approximately 1.7 obtained by Chatwin & Sullivan. This is because the velocity profile approximation of a stagnant side region is only reasonable for very large a/b aspect ratios and is a very poor approximation for square channels.

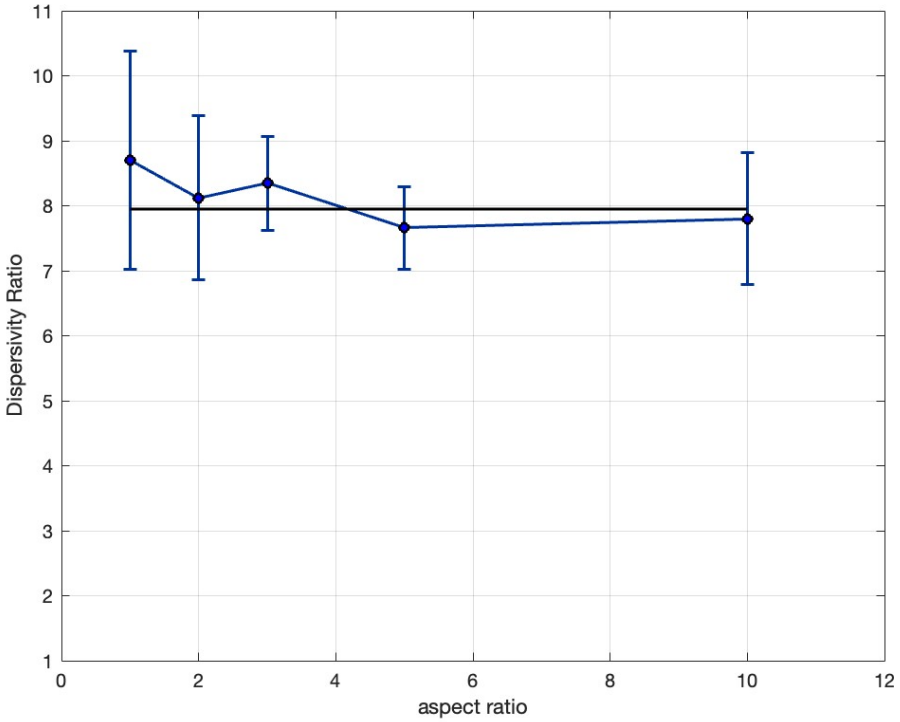


Figure 24.4: Calculated ratio of Taylor Dispersivity to plane-Poiseuille result as a function of aspect ratio modeled by the stagnant layer approximation. The asymptote for large ratios is included. Error bars are 95% random error confidence intervals in the mean values, but do not account for error due to finite discretization or the side wall approximation which would needed to be investigated separately.

Solution 2: The Square Channel Limit and "Vibe Coding"

Part (a): Analytical Form Generation

When an AI is prompted for a simplified 2D approximation that satisfies both physical symmetry

and no-slip boundary conditions at $x^* = \pm 1$ and $y^* = \pm 1$, a highly effective trial function is:

$$u^*_{\text{approx}}(x^*, y^*) = A(1 - x^{*2})(1 - y^{*2})(1 + Bx^{*2}y^{*2})$$

The terms $(1 - x^{*2})$ and $(1 - y^{*2})$ strictly enforce the zero-velocity boundary conditions at the walls, while the even powers naturally ensure symmetry across the centerlines. The two parameters, A and B, give the function exactly enough degrees of freedom to exactly match both the average velocity and the spatial variance of the true infinite series solution.

Part (b): Verification Script

The following MATLAB script sets up a discrete grid, calculates the exact series solution, and then uses a simple optimization routine (`fminsearch`) to find the values of A and B that perfectly match the exact moments.

```
%% Part B: Verification and Parameter Fitting
N = 1000; % Grid resolution
x = linspace(0, 1, N);
y = linspace(0, 1, N);
[X, Y] = meshgrid(x, y);

% 1. Calculate Exact Velocity Profile (First 50 terms)
U_exact = 1.5 * (1 - Y.^2);
for n = 1:50
    sigma = (n - 0.5) * pi;
    term = (6 * (-1)^n / sigma^3) * (cosh(sigma * X) ./ cosh(sigma)) .* cos(sigma * Y);
    U_exact = U_exact + term;
end

% Calculate Exact Moments
mean_U = mean(U_exact(:));
var_U = var(U_exact(:), 1); % Population variance

fprintf('Exact Mean Velocity: %.4f\n', mean_U);
fprintf('Exact Velocity Variance: %.4f\n', var_U);

% 2. Fit the Approximate Profile to match mean and variance
% U_approx = A * (1 - X^2) * (1 - Y^2) * (1 + B * X^2 * Y^2)
% p(1) = A, p(2) = B
obj = @(p) (mean( p(1)*(1-X.^2).*(1-Y.^2).*(1+p(2)*X.^2.*Y.^2), 'all' ) - mean_U)^2 + ...
           (var( p(1)*(1-X.^2).*(1-Y.^2).*(1+p(2)*X.^2.*Y.^2), 1, 'all' ) - var_U)^2;

p_opt = fminsearch(obj, [1, 1]);
A = p_opt(1);
B = p_opt(2);

fprintf('Optimized Parameters: A = %.4f, B = %.4f\n', A, B);
```

Part (c): Modified Monte Carlo Simulation

Once A and B are found, we normalize the function so its average velocity is 1. The analytical average of $(1-x^2)(1-y^2)(1+Bx^2y^2)$ over the domain is $(4/9+4B/225)$. By using this normalized function, the fluid translates with a dimensionless average velocity of 1, meaning we no longer need to adjust our final calculated dispersivity by the factor $(U^*/U)^2$.

Running the Monte Carlo simulation with this valid profile effectively bridges the gap, confirming that the simulated dispersivity approaches ~ 1.7 times the plane-Poiseuille limit.

```
%% Part C: Modified Monte Carlo Simulation for Square Channel
n = 10000;      % Number of tracers
ab = 1;         % Square channel
dt = 0.00005;   % Time step
tquit = 5;      % Dimensionless time (5 is sufficient for square)
t = (dt:dt:tquit)';

% Normalize the function so average velocity is 1
C = 1 / (4/9 + 4*B/225);
uz = @(x,y) C * (1-x.^2) .* (1-y.^2) .* (1 + B * x.^2 .* y.^2);

n_runs = 5;
ratios = zeros(n_runs, 1);

for run = 1:n_runs
    x = rand(n,1);
    y = rand(n,1);
    z = zeros(n,1);
    varz = zeros(size(t));

    for i = 1:length(t)
        dz1 = uz(x,y);

        % Brownian kick and reflection from walls at 0 and 1
        x = abs(x + randn(n,1)*(2*dt)^.5);
        x = min(x, 2-x);

        y = abs(y + randn(n,1)*(2*dt)^.5);
        y = min(y, 2-y);

        dz2 = uz(x,y);

        % Advection step
        z = z + (dz1+dz2)/2 * dt;
        varz(i) = var(z);
    end
```

```
 % Extract asymptotic slope (last third)
    mid = floor(length(t) * 2 / 3);
    p = polyfit(t(mid:end), varz(mid:end), 1);

    % Calculate dispersivity ratio
    D_eff_sim = p(1) / 2;
    D_eff_1D = 2 / 105;
    ratios(run) = D_eff_sim / D_eff_1D;
end

% 95% Confidence Interval for 5 runs (4 degrees of freedom)
mean_ratio = mean(ratios);
std_err = std(ratios) / sqrt(n_runs);
ci_95 = 2.776 * std_err;

fprintf('Final Dispersivity Ratio: %.3f +/- %.3f\n', mean_ratio, ci_95);
```

Discussion: Examination of the code provided by the AI reveals some interesting choices. In calculating the coefficients A and B the AI (Gemini Pro, in this case) used a simple mean and variance of the computed values of the velocity rather than integrating. This introduces a truncation error, and also double counts the end points. Initially it chose 200 points, this was increased to 1000 to get a higher accuracy for B. Because the mean and variances of both exact and approximate distributions were calculated the same way, the error in B was not significant. Once B was obtained the new coefficient C (to get the correct average velocity) was calculated analytically, and it did that correctly. Because the AI solved problem 2 after problem 1 it chose to do only five runs (but with a much larger number of tracers). It calculated the confidence interval correctly using t-statistics, but it would probably have been better if it used a smaller number of tracers but a much larger number of runs, reducing the uncertainty in the standard deviation. The code, however, runs well and produces the correct result. In contrast, if you were to use the following prompt:

> *Write a matlab script which calculates the Taylor dispersivity for a square duct of half-width b and average velocity U. Find the ratio of this to the Taylor dispersivity for the same geometry and average velocity but with no side walls (e.g., the plane-Poiseuille flow result).*

you get a completely different approach - the AI tries to solve it using the Aris method of moments approach, which is mathematically more elegant and potentially more accurate - but when you run the code it gives back garbage. This is the problem with "straight up" vibe coding, where you can get an answer that may look reasonable, but yields an incorrect answer without a lot of debugging - and that requires knowing the math behind the method!

Index of Selected Examples

Index of Selected Problems