Chapter 14

Partial Differential Equations

Our intuition for ordinary differential equations generally stems from the time evolution of physical systems. Equations like Newton's second law determining the motion of physical objects over time dominate the literature on such initial value problems; additional examples come from chemical concentrations reacting over time, populations of predators and prey interacting from season to season, and so on. In each case, the initial configuration—e.g. the positions and velocities of particles in a system at time zero—are known, and the task is to predict behavior as time progresses.

In this chapter, however, we entertain the possibility of *coupling* relationships between different derivatives of a function. It is not difficult to find examples where this coupling is necessary. For instance, when simulating smoke or gases quantities like "pressure gradients," the derivative of the pressure of a gas in *space*, figure into how the gas moves over *time*; this structure is reasonable since gas naturally diffuses from high-pressure regions to low-pressure regions. In image processing, derivatives couple even more naturally, since measurements about images tend to occur in the x and y directions simultaneously.

Equations coupling together derivatives of functions are known as *partial differential equations*. They are the subject of a rich but strongly nuanced theory worthy of larger-scale treatment, so our goal here will be to summarize key ideas and provide sufficient material to solve problems commonly appearing in practice.

14.1 Motivation

Partial differential equations (PDEs) arise when the unknown is some function $f : \mathbb{R}^n \to \mathbb{R}^m$. We are given one or more relationship between the partial derivatives of f, and the goal is to find an f that satisfies the criteria. PDEs appear in nearly any branch of applied mathematics, and we list just a few below.

As an aside, before introducing specific PDEs we should introduce some notation. In particular, there are a few combinations of partial derivatives that appear often in the world of PDEs. If $f : \mathbb{R}^3 \to \mathbb{R}$ and $\vec{v} : \mathbb{R}^3 \to \mathbb{R}^3$, then the following operators are worth remembering:

$$\text{Gradient: } \nabla f \equiv \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3} \right)$$

$$\text{Divergence: } \nabla \cdot \vec{v} \equiv \frac{\partial v_1}{\partial x_1} + \frac{\partial v_2}{\partial x_2} + \frac{\partial v_3}{\partial x_3}$$
$$\text{Curl: } \nabla \times \vec{v} \equiv \left(\frac{\partial v_3}{\partial x_2} - \frac{\partial v_2}{\partial x_3}, \frac{\partial v_1}{\partial x_3} - \frac{\partial v_3}{\partial x_1}, \frac{\partial v_2}{\partial x_1} - \frac{\partial v_1}{\partial x_2} \right)$$
$$\text{Laplacian: } \nabla^2 f \equiv \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2} + \frac{\partial^2 f}{\partial x_3^2}$$

Example 14.1 (Fluid simulation)**.** *The flow of fluids like water and smoke is governed by the* Navier-Stokes equations*, a system of PDEs in many variables. In particular, suppose a fluid is moving in some region $\Omega \subseteq \mathbb{R}^3$. We define the following variables, illustrated in Figure NUMBER:*

- $t \in [0, \infty)$*: Time*
- $\vec{v}(t) : \Omega \to \mathbb{R}^3$*: The velocity of the fluid*
- $\rho(t) : \Omega \to \mathbb{R}$*: The density of the fluid*
- $p(t) : \Omega \to \mathbb{R}$*: The pressure of the fluid*
- $\vec{f}(t) : \Omega \to \mathbb{R}^3$*: External forces like gravity on the fluid*

If the fluid has viscosity μ, then if we assume it is incompressible *the Navier-Stokes equations state:*

$$\rho \left(\frac{\partial \vec{v}}{\partial t} + \vec{v} \cdot \nabla \vec{v} \right) = -\nabla p + \mu \nabla^2 \vec{v} + \vec{f}$$

Here, $\nabla^2 \vec{v} = \partial^2 v_1 / \partial x_1^2 + \partial^2 v_2 / \partial x_2^2 + \partial^2 v_3 / \partial x_3^2$; we think of the gradient ∇ as a gradient in space rather than time, i.e. $\nabla f = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3})$. This system of equations determines the time dynamics of fluid motion and actually can be constructed by applying Newton's second law to tracking "particles" of fluid. Its statement, however, involves not only derivatives in time $\frac{\partial}{\partial t}$ but also derivatives in space ∇, making it a PDE.

Example 14.2 (Maxwell's equations)**.** *Maxwell's equations determine the interaction of electric fields $\vec{E}$ and magnetic fields $\vec{B}$ over time. As with the Navier-Stokes equations, we think of the gradient, divergence, and curl as taking partial derivatives in space (and not time t). Then, Maxwell's system (in "strong" form) can be written:*

$$\text{Gauss's law for electric fields: } \nabla \cdot \vec{E} = \frac{\rho}{\varepsilon_0}$$
$$\text{Gauss's law for magnetism: } \nabla \cdot \vec{B} = 0$$
$$\text{Faraday's law: } \nabla \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$
$$\text{Ampère's law: } \nabla \times \vec{B} = \mu_0 \left(\vec{J} + \varepsilon_0 \frac{\partial \vec{E}}{\partial t} \right)$$

Here, ε_0 and μ_0 are physical constants and $\vec{J}$ encodes the density of electrical current. Just like the Navier-Stokes equations, Maxwell's equations related derivatives of physical quantities in time t to their derivatives in space (given by curl and divergence terms).

Example 14.3 (Laplace's equation)**.** *Suppose Ω is a domain in $\mathbb{R}^2$ with boundary $\partial\Omega$ and that we are given a function $g : \partial\Omega \to \mathbb{R}$, illustrated in Figure NUMBER. We may wish to interpolate g to the interior of Ω. When Ω is an irregular shape, however, our strategies for interpolation from Chapter 11 can break down.*

Suppose we define $f(\vec{x}) : \Omega \to \mathbb{R}$ to be the interpolating function. Then, one strategy inspired by our approach to least-squares is to define an energy functional:

$$E[f] = \int_\Omega \|\nabla f(\vec{x})\|_2^2 \, d\vec{x}$$

That is, $E[f]$ measures the "total derivative" of f measured by taking the norm of its gradient and integrating this quantity over all of Ω. Wildly fluctuating functions f will have high values of $E[f]$ since the slope ∇f will be large in many places; smooth and low-frequency functions f, on the other hand, will have small $E[f]$ since their slope will be small everywhere.[1] *Then, we could ask that f interpolates g while being as smooth as possible in the interior of Ω using the following optimization:*

$$\begin{aligned} &\text{minimize}_f \; E[f] \\ &\text{such that } f(\vec{x}) = g(\vec{x}) \; \forall x \in \partial\Omega \end{aligned}$$

This setup looks *like optimizations we have solved in previous examples, but now our unknown is a function f rather than a point in $\mathbb{R}^n$!*

If f minimizes E, then $E[f + h] \geq E[f]$ for all functions $h(\vec{x})$. This statement is true even for small perturbations $E[f + \varepsilon h]$ as $\varepsilon \to 0$. Dividing by ε and taking the limit as $\varepsilon \to 0$, we must have $\frac{d}{d\varepsilon} E[f + \varepsilon h]|_{\varepsilon=0} = 0$; this is just like setting directional derivatives of a function equal to zero to find its minima. We can simplify:

$$\begin{aligned} E[f + \varepsilon h] &= \int_\Omega \|\nabla f(\vec{x}) + \varepsilon \nabla h(\vec{x})\|_2^2 \, d\vec{x} \\ &= \int_\Omega (\|\nabla f(\vec{x})\|_2^2 + 2\varepsilon \nabla f(\vec{x}) \cdot \nabla h(\vec{x}) + \varepsilon^2 \|\nabla h(\vec{x})\|_2^2) \, d\vec{x} \end{aligned}$$

Differentiating shows:

$$\begin{aligned} \frac{d}{d\varepsilon} E[f + \varepsilon h] &= \int_\Omega (2\nabla f(\vec{x}) \cdot \nabla h(\vec{x}) + 2\varepsilon \|\nabla h(\vec{x})\|_2^2) \, d\vec{x} \\ \frac{d}{d\varepsilon} E[f + \varepsilon h]|_{\varepsilon=0} &= 2 \int_\Omega [\nabla f(\vec{x}) \cdot \nabla h(\vec{x})] \, d\vec{x} \end{aligned}$$

This derivative must equal zero for all h, so in particular we can choose $h(\vec{x}) = 0$ for all $\vec{x} \in \partial\Omega$. Then, applying integration by parts, we have:

$$\frac{d}{d\varepsilon} E[f + \varepsilon h]|_{\varepsilon=0} = -2 \int_\Omega h(\vec{x}) \nabla^2 f(\vec{x}) \, d\vec{x}$$

This expression must equal zero for all *(all!) perturbations h, so we must have $\nabla^2 f(\vec{x}) = 0$ for all $\vec{x} \in \Omega \backslash \partial\Omega$ (a formal proof is outside of the scope of our discussion). That is, the interpolation problem above*

[1] The notation $E[\cdot]$ used here does not stand for "expectation" as it might in probability theory, but rather simply is an "energy" functional; it is standard notation in areas of functional analysis.

can be solved using the following PDE:

$$\begin{aligned}\nabla^2 f(\vec{x}) &= 0 \\ f(\vec{x}) &= g(\vec{x}) \; \forall \vec{x} \in \partial\Omega\end{aligned}$$

This equation is known as Laplace's equation*, and it can be solved using sparse positive definite linear methods like what we covered in Chapter 10. As we have seen, it can be applied to interpolation problems for irregular domains Ω; furthermore, $E[f]$ can be augmented to measure other properties of f, e.g. how well f approximates some noisy function f_0, to derive related PDEs by paralleling the argument above.*

Example 14.4 (Eikonal equation)**.** *Suppose $\Omega \subseteq \mathbb{R}^n$ is some closed region of space. Then, we could take $d(\vec{x})$ to be a function measuring the distance from some point $\vec{x}_0$ to $\vec{x}$ completely within Ω. When Ω is convex, we can write d in closed form:*

$$d(\vec{x}) = \|\vec{x} - \vec{x}_0\|_2.$$

As illustrated in Figure NUMBER, however, if Ω is non-convex or a more complicated domain like a surface, distances become more complicated to compute. In this case, distance functions d satisfy the localized condition known as the eikonal equation*:*

$$\|\nabla d\|_2 = 1.$$

If we can compute it, d can be used for tasks like planning paths of robots by minimizing the distance they have to travel with the constraint that they only can move in Ω.

Specialized algorithms known as fast marching methods *are used to find estimates of d given $\vec{x}_0$ and Ω by integrating the eikonal equation. This equation is nonlinear in the derivative ∇d, so integration methods for this equation are somewhat specialized, and proof of their effectiveness is complex. Interestingly but unsurprisingly, many algorithms for solving the eikonal equation have structure similar to Dijkstra's algorithm for computing shortest paths along graphs.*

Example 14.5 (Harmonic analysis)**.** *Different objects respond differently to vibrations, and in large part these responses are functions of the* geometry *of the objects. For example, cellos and pianos can play the same note, but even an inexperienced musician easily can distinguish between the sounds they make. From a mathematical standpoint, we can take $\Omega \subseteq \mathbb{R}^3$ to be a shape represented either as a surface or a volume. If we clamp the edges of the shape, then its* frequency spectrum *is given by solutions of the following differential eigenvalue problem:*

$$\begin{aligned}\nabla^2 f &= \lambda f \\ f(x) &= 0 \; \forall x \in \partial\Omega,\end{aligned}$$

where ∇^2 is the Laplacian of Ω and $\partial\Omega$ is the boundary of Ω. Figure NUMBER shows examples of these functions on different domains Ω.

It is easy to check that $\sin kx$ *solves this problem when Ω is the interval $[0, 2\pi]$, for $k \in \mathbb{Z}$. In particular, the Laplacian in one dimension is $\partial^2/\partial x^2$, and thus we can check:*

$$\begin{aligned}\frac{\partial^2}{\partial x^2}\sin kx &= \frac{\partial}{\partial x} k\cos kx \\ &= -k^2 \sin kx \\ \sin k \cdot 0 &= 0 \\ \sin k \cdot 2\pi &= 0\end{aligned}$$

Thus, the eigenfunctions are $\sin kx$ *with eigenvalues $-k^2$.*

14.2 Basic definitions

Using the notation of CITE, we will assume that our unknown is some function $f : \mathbb{R}^n \to \mathbb{R}$. For equations of up to three variables, we will use subscript notation to denote partial derivatives:

$$f_x \equiv \frac{\partial f}{\partial x},$$
$$f_y \equiv \frac{\partial f}{\partial y},$$
$$f_{xy} \equiv \frac{\partial^2 f}{\partial x \partial y},$$

and so on.

Partial derivatives usually are stated as relationships between two or more derivatives of f, as in the following:

- Linear, homogeneous: $f_{xx} + f_{xy} - f_y = 0$
- Linear: $f_{xx} - y f_{yy} + f = xy^2$
- Nonlinear: $f_{xx}^2 = f_{xy}$

Generally, we really wish to find $f : \Omega \to \mathbb{R}$ for some $\Omega \subseteq \mathbb{R}^n$. Just as ODEs were stated as initial value problems, we will state most PDEs as *boundary value problems.* That is, our job will be to fill in f in the interior of Ω given values on its boundary. In fact, we can think of the ODE initial value problem this way: the domain is $\Omega = [0, \infty)$, with boundary $\partial\Omega = \{0\}$, which is where we provide input data. Figure NUMBER illustrates more complex examples. Boundary conditions for these problems take many forms:

- *Dirichlet conditions* simply specify the value of $f(\vec{x})$ on $\partial\Omega$
- *Neumann conditions* specify the derivatives of $f(\vec{x})$ on $\partial\Omega$
- *Mixed* or *Robin conditions* combine these two

14.3 Model Equations

Recall from the previous chapter that we were able to understand many properties of ODEs by examining a *model equation* $y' = ay$. We can attempt to pursue a similar approach for PDEs, although we will find that the story is more nuanced when derivatives are linked together.

As with the model equation for ODEs, we will study the single-variable *linear* case. We also will restrict ourselves to *second-order* systems, that is, systems containing at most the second derivative of u; the model ODE was first-order but here we need at least two orders to study how derivatives interact in a nontrivial way.

A linear second-order PDE has the following general form:

$$\sum_{ij} a_{ij} \frac{\partial f}{\partial x_i \partial x_j} + \sum_i b_i \frac{\partial f}{\partial x_i} + c = 0$$

Formally, we might define the "gradient operator" as:

$$\nabla \equiv \left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n} \right).$$

You should check that this operator is reasonable notation in that expressions like ∇f, $\nabla \cdot \vec{v}$, and $\nabla \times \vec{v}$ all provide the proper expressions. In this notation, the PDE can be thought of as taking a matrix form:

$$(\nabla^\top A \nabla + \nabla \cdot \vec{b} + c) f = 0.$$

This form has much in common with our study of quadratic forms in conjugate gradients, and in fact we usually characterize PDEs by the structure of A:

- If A is *positive or negative definite*, system is *elliptic*.
- If A is *positive or negative semidefinite*, the system is *parabolic*.
- If A has only one eigenvalue of different sign from the rest, the system is *hyperbolic*.
- If A satisfies none of the criteria, the system is *ultrahyperbolic.*

These criteria are listed approximately in order of the difficulty level of solving each type of equation. We consider the first three cases below and provide examples of corresponding behavior; ultrahyperbolic equations do not appear as often in practice and require highly specialized techniques for their solution.

TODO: Reduction to canonical form via eigenstuff of A (not in 205A)

14.3.1 Elliptic PDEs

Just as positive definite matrices allow for specialized algorithms like Cholesky decomposition and conjugate gradients that simplify their inversion, elliptic PDEs have particularly strong structure that leads to effective solution techniques.

The model elliptic PDE is the *Laplace equation*, given by $\nabla^2 f = g$ for some given function g as in Example 14.3. For instance, in two variables the Laplace equation becomes

$$f_{xx} + f_{yy} = g.$$

Figure NUMBER illustrates some solutions of the Laplace equation for different choices of u and f on a two-dimensional domain.

Elliptic equations have many important properties. Of particular theoretical and practical importance is the idea of *elliptic regularity*, that solutions of elliptic PDEs automatically are smooth functions in $C^\infty(\Omega)$. This property is not immediately obvious: a second-order PDE in f only requires that f be twice-differentiable to make sense, but in fact under weak conditions they automatically are infinitely differentiable. This property lends to the physical intuition that elliptic equations represent stable physical equilbria like the rest pose of a stretched out rubber sheet. Second-order elliptic equations in the form above also are guaranteed to admit solutions, unlike PDEs in some other forms.

Example 14.6 (Poisson in one variable)**.** *The Laplace equation with $g = 0$ is given the special name* Poisson's equation. *In one variable, it can be written $f''(x) = 0$, which trivially is solved by $f(x) = \alpha x + \beta$. This equation is sufficient to examine possible boundary conditions on $[a, b]$:*

- *Dirichlet conditions for this equation simply specify $f(a)$ and $f(b)$; there is obviously a unique line that goes through $(a, f(a))$ and $(b, f(b))$, which provides the solution to the equation.*

- *Neumann conditions would specify $f'(a)$ and $f'(b)$. But, $f'(a) = f'(b) = \alpha$ for $f(x) = \alpha x + \beta$. In this way, boundary values for Neumann problems can be subject to* compatibility conditions *needed to admit admit a solution. Furthermore, the choice of β does not affect the boundary conditions, so when they are satisfied the solution is not unique.*

14.3.2 Parabolic PDEs

Continuing to parallel the structure of linear algebra, positive *semi*definite systems of equations are only slightly more difficult to deal with than positive definite ones. In particular, positive semidefinite matrices admit a null space which must be dealt with carefully, but in the remaining directions the matrices behave the same as the definite case.

The heat equation is the model parabolic PDE. Suppose $f(0; x, y)$ is a distribution of heat in some region $\Omega \subseteq \mathbb{R}^2$ at time $t = 0$. Then, the heat equation determines how the heat diffuses over time t as a function $f(t; x, y)$:

$$\frac{\partial f}{\partial t} = \alpha \nabla^2 f,$$

where $\alpha > 0$ and we once again think of ∇^2 as the Laplacian in the *space variables* x and y, that is, $\nabla^2 = \partial^2/\partial x^2 + \partial^2/\partial y^2$. This equation must be parabolic, since there is the same coefficient α in front of f_{xx} and f_{yy}, but f_{tt} does not figure into the equation.

Figure NUMBER illustrates a phenomenological interpretation of the heat equation. We can think of $\nabla^2 f$ as measuring the convexity of f, as in Figure NUMBER(a). Thus, the heat equation increases u with time when its value is "cupped" upward, and decreases f otherwise. This negative feedback is stable and leads to equilibrium as $t \to \infty$.

There are two boundary conditions needed for the heat equation, both of which come with straightforward physical interpretations:

- The distribution of heat $f(0; x, y)$ at time $t = 0$ at all points $(x, y) \in \Omega$

- Behavior of f when $t > 0$ at points $(x, y) \in \partial\Omega$. These boundary conditions describe behavior at the boundary of the domain. Dirichlet conditions here provide $f(t; x, y)$ for all $t \geq 0$ and $(x, y) \in \partial\Omega$, corresponding to the situation in which an outside agent fixes the temperatures at the boundary of the domain. These conditions might occur if Ω is a piece of foil sitting next to a heat source whose temperature is not significantly affected by the foil like a large refrigerator or oven. Contrastingly, Neumann conditions specify the derivative of f in the direction normal to the boundary $\partial\Omega$, as in Figure NUMBER; they correspond to fixing the *flux* of heat out of Ω caused by different types of insulation.

14.3.3 Hyperbolic PDEs

The final model equation is the wave equation, corresponding to the indefinite matrix case:

$$\frac{\partial^2 f}{\partial t^2} - c^2 \nabla^2 f = 0$$

The wave equation is hyperbolic because the second derivative in time has opposite sign from the two spatial derivatives. This equation determines the motion of waves across an elastic medium like a rubber sheet; for example, it can be derived by applying Newton's second law to points on a piece of elastic, where x and y are positions on the sheet and $f(t;x,y)$ is the height of the piece of elastic at time t.

Figure NUMBER illustrates a one-dimensional solution of the wave equation. Wave behavior contrasts considerably with heat diffusion in that as $t \to \infty$ energy may not diffuse. In particular, waves can bounce back and forth across a domain indefinitely. For this reason, we will see that implicit integration strategies may not be appropriate for integrating hyperbolic PDEs because they tend to damp out motion.

Boundary conditions for the wave equation are similar to those of the heat equation, but now we must specify both $f(0;x,y)$ and $f_t(0;x,y)$ at time zero:

- The conditions at $t = 0$ specify the position and velocity of the wave at the initial time.
- Boundary conditions on Ω determine what happens at the ends of the material. Dirichlet conditions correspond to fixing the sides of the wave, e.g. plucking a cello string, which is held flat at its two ends on the instrument. Neumann conditions correspond to leaving the ends of the wave untouched like the end of a whip.

14.4 Derivatives as Operators

In PDEs and elsewhere, we can think of derivatives as operators acting on functions the same way that matrices act on vectors. Our choice of notation often reflects this parallel: The derivative df/dx looks like the product of an operator d/dx and a function f. In fact, differentiation is a *linear operator* just like matrix multiplication, since for all $f, g : \mathbb{R} \to \mathbb{R}$ and $a, b \in \mathbb{R}$

$$\frac{d}{dx}(af(x) + bg(x)) = a\frac{d}{dx}f(x) + b\frac{d}{dx}g(x).$$

In fact, when we discretize PDEs for numerical solution, we can carry this analogy out completely. For example, consider a function f on $[0,1]$ discretized using $n+1$ evenly-spaced samples, as in Figure NUMBER. Recall that the space between two samples is $h = 1/n$. In Chapter 12, we developed an approximation for the second derivative $f''(x)$:

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h)$$

Suppose our n samples of $f(x)$ on $[0,1]$ are $y_0 \equiv f(0), y_1 \equiv f(h), y_2 \equiv f(2h), \ldots, y_n = f(nh)$. Then, applying our formula above gives a strategy for approximating f'' at each grid point:

$$y_k'' \equiv \frac{y_{k+1} - 2y_k + y_{k-1}}{h^2}$$

That is, the second derivative of a function on a grid of points can be computed using the 1—$-$2—1 stencil illustrated in Figure NUMBER(a).

One subtlety we did not address is what happens at y_0'' and y_n'', since the formula above would require y_{-1} and y_{n+1}. In fact, this decision encodes the *boundary conditions* introduced in §14.2. Keeping in mind that $y_0 = f(0)$ and $y_n = f(1)$, examples of possible boundary conditions for f' include:

- Dirichlet boundary conditions: $y_{-1} = y_{n+1} = 0$, that is, simply fix the value of y beyond the endpoints
- Neumann boundary conditions: $y_{-1} = y_0$ and $y_{n+1} = y_n$, encoding the boundary condition $f'(0) = f'(1) = 0$.
- Periodic boundary conditions: $y_{-1} = y_n$ and $y_{n+1} = y_0$, making the identification $f(0) = f(1)$

Suppose we stack the samples y_k into a vector $\vec{y} \in \mathbb{R}^{n+1}$ and the samples y_k'' into a second vector $\vec{w} \in \mathbb{R}^{n+1}$. Then, our construction above it is easy to see that $h^2\vec{w} = L_1\vec{y}$, where L_1 is one of the choices below:

$$\begin{pmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 \end{pmatrix} \quad \begin{pmatrix} -1 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -1 \end{pmatrix} \quad \begin{pmatrix} -2 & 1 & & & & & 1 \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ 1 & & & & 1 & -2 \end{pmatrix}$$

Dirichlet **Neumann** **Periodic**

That is, the matrix L can be thought of as a discretized version of the operator $\frac{d^2}{dx^2}$ acting on $\vec{y} \in \mathbb{R}^{n+1}$ rather than functions $f : [0,1] \to \mathbb{R}$.

We can write a similar approximation for $\nabla^2 f$ when we sample $f : [0,1] \times [0,1] \to \mathbb{R}$ with a grid of values, as in Figure NUMBER. In particular, recall that in this case $\nabla^2 f = f_{xx} + f_{yy}$, so in particular we can sum up x and y second derivatives as we did in the one-dimensional example above. This leads to a doubled-over 1—$-$2—1 stencil, as in Figure NUMBER. If we number our samples as $y_{k,\ell} \equiv f(kh, \ell h)$, then our formula for the Laplacian of f is in this case:

$$(\nabla^2 y)_{k,\ell} \equiv \frac{1}{h^2}(y_{(k-1),\ell} + y_{k,(\ell-1)} + y_{(k+1),\ell} + y_{k,(\ell+1)} - 4y_{k,\ell})$$

If we once again combine our samples of y and y'' into $\vec{y}$ and $\vec{w}$, then using a similar construction and choice of boundary conditions we can once again write $h^2\vec{w} = L_2\vec{y}$. This two-dimensional grid Laplacian L_2 appears in many image processing applications, where (k, ℓ) is used to index pixels on an image.

A natural question to ask after the discussion above is why we jumped to the second derivative Laplacian in our discussion above rather than discretizing the first derivative $f'(x)$. In principle, there is no reason why we could not make similar matrices D implementing forward, backward, or symmetric difference approximations of f'. A few technicalities, however, make this task somewhat more difficult, as detailed below.

Most importantly, we must decide which first derivative approximation to use. If we write y_k' as the forward difference $\frac{1}{h}(y_{k+1} - y_k)$, for example, then we will be in the unnaturally asymmetric position of needing a boundary condition at y_n but not at y_0. Contrastingly, we could use the symmetric difference $\frac{1}{2h}(y_{k+1} - y_{k-1})$, but this discretization suffers from a more subtle *fencepost problem* illustrated in Figure NUMBER. In particular, this version of y_k' ignores the value of y_k and only looks at its neighbors y_{k-1} and y_{k+1}, which can create artifacts since each row of D only involves y_k for either even or odd k but not both.

If we use forward or backward derivatives to avoid the fencepost problems, we lose an order of accuracy and also suffer from the asymmetries described above. As with the leapfrog integration

algorithm in §13.4.2, one way to avoid these issues is to think of the derivatives as living on *half* gridpoints, illustrated in Figure NUMBER. In the one-dimensional case, this change corresponds to labeling the difference $\frac{1}{h}(y_{k+1} - y_k)$ as $y_{k+1/2}$. This technique of placing different derivatives on vertices, edges, and centers of grid cells is particularly common in fluid simulation, which maintains pressures, fluid velocities, and so on at locations that simplify calculations.

These subtleties aside, our main conclusion from this discussion is that if we discretize a function $f(\vec{x})$ by keeping track of samples (x_i, y_i) then most reasonable approximations of derivatives of f will be computable as a product $L\vec{x}$ for some matrix L. This observation completes the analogy: "Derivatives act on functions as matrices act on vectors." Or in standardized exam notation:

Derivatives : Functions :: Matrices : Vectors

14.5 Solving PDEs Numerically

Much remains to be said about the theory of PDEs. Questions of existence and uniqueness as well as the possibility of characterizing solutions to assorted PDEs leads to nuanced discussions using advanced aspects of real analysis. While a complete understanding of these properties is needed to prove effectiveness of PDE discretizations rigorously, we already have enough to suggest a few techniques that are used in practice.

14.5.1 Solving Elliptic Equations

We already have done most of the work for solving elliptic PDEs in §14.4. In particular, suppose we wish to solve a linear elliptic PDE of the form $\mathcal{L}f = g$. Here $\mathcal{L}$ is a differential operator; for example, to solve the Laplace's equation we would take $\mathcal{L} \equiv \nabla^2$, the Laplacian. Then, in §14.4 we showed that if we discretize f by taking a set of samples in a vector $\vec{y}$ with $y_i = f(x_i)$, then a corresponding approximation of $\mathcal{L}f$ can be written $L\vec{y}$ for some matrix L. If we also discretize g using samples in a vector $\vec{b}$, then solving the elliptic PDE $\mathcal{L}f = g$ is approximated by solving the linear system $L\vec{y} = \vec{b}$.

Example 14.7 (Elliptic PDE discretization)**.** *Suppose we wish to approximate solutions to $f''(x) = g(x)$ on $[0, 1]$ with boundary conditions $f(0) = f(1) = 0$. We will approximate $f(x)$ with a vector $\vec{y} \in \mathbb{R}^n$ sampling f as follows:*

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \approx \begin{pmatrix} f(h) \\ f(2h) \\ \vdots \\ f(nh) \end{pmatrix}$$

where $h = 1/n+1$. We do not add samples at $x = 0$ or $x = 1$ since the boundary conditions determine values there. We will use $\vec{b}$ to hold an analogous set of values for $g(x)$.

Given our boundary conditions, we discretize $f''(x)$ as $\frac{1}{h^2}L\vec{y}$, where L is given by:

$$L \equiv \begin{pmatrix} -2 & 1 & & & & & \\ 1 & -2 & 1 & & & & \\ & 1 & -2 & 1 & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & 1 & -2 & 1 & \\ & & & & 1 & -2 \end{pmatrix}$$

Thus, our approximate solution to the PDE is given by $\vec{y} = h^2 L^{-1}\vec{b}$.

Just as elliptic PDEs are the most straightforward PDEs to solve, their discretizations using matrices as in the above example are the most straightforward to solve. In fact, generally the discretization of an elliptic operator $\mathcal{L}$ is a *positive definite* and *sparse* matrix perfectly suited for the solution techniques derived in Chapter 10.

Example 14.8 (Elliptic operators as matrices)**.** *Consider the matrix L from Example 14.7. We can show L is negative definite (and hence the* positive *definite system $-L\vec{y} = -h^2\vec{b}$ can be solved using conjugate gradients) by noticing that $-L = D^\top D$ for the matrix $D \in \mathbb{R}^{(n+1)\times n}$ given by:*

$$D = \begin{pmatrix} 1 & & & & \\ -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \\ & & & & -1 \end{pmatrix}$$

This matrix is nothing more than the finite-differenced first *derivative, so this observation parallels the fact that $d^2f/dx^2 = d/dx(df/dx)$. Thus, $\vec{x}^\top L\vec{x} = -\vec{x}^\top D^\top D\vec{x} = -\|D\vec{x}\|_2^2 \leq 0$, showing L is negative semidefinite. It is easy to see $D\vec{x} = 0$ exactly when $\vec{x} = 0$, completing the proof that L is in fact negative definite.*

14.5.2 Solving Parabolic and Hyperbolic Equations

Parabolic and hyperbolic equations generally introduce a *time* variable into the formulation, which also is differentiated but potentially to lower order. Since solutions to parabolic equations admit many stability properties, numerical techniques for dealing with this time variable often are stable and well-conditioned; contrastingly, more care must be taken to treat hyperbolic behavior and prevent dampening of motion over time.

Semidiscrete Methods Probably the most straightforward approach to solving simple time-dependent equations is to use a *semidiscrete* method. Here, we discretize the domain but not the time variable, leading to an ODE that can be solved using the methods of Chapter 13.

Example 14.9 (Semidiscrete heat equation)**.** *Consider the heat equation in one variable, given by $f_t = f_{xx}$, where $f(t;x)$ represents the heat at position x and time t. As boundary data, the user provides a*

function $f_0(x)$ such that $f(0;x) \equiv f_0(x)$; we also attach the boundary $x \in \{0,1\}$ to a refrigerator and thus enforce $f(t;0) = f(t;1) = 0$.

Suppose we discretize the x variable by defining:

$$\begin{aligned}\bar{f}_1(t) &\equiv f(h;t)\\ \bar{f}_2(t) &\equiv f(2h;t)\\ &\vdots \quad \vdots\\ \bar{f}_n(t) &\equiv f(nh;t),\end{aligned}$$

where as in Example 14.7 we take $h = 1/n+1$ and omit samples at $x \in \{0,1\}$ since they are provided by the boundary conditions.

Combining these $\bar{f}_i$'s, we can define $\bar{f}(t) : \mathbb{R} \to \mathbb{R}^n$ to be the semidiscrete *version of f where we have sampled in space but not time. By our construction, the semidiscrete PDE approximation is the ODE given by $\bar{f}'(t) = L\bar{f}(t)$.*

The previous example shows an instance of a very general pattern for parabolic equations. When we simulate *continuous* phenomena like heat moving across a domain or chemicals diffusing through a membrane, there is usually one time variable and then several spatial variables that are differentiated in an elliptic way. When we discretize this system semidiscretely, we can then use ODE integration strategies for their solution. In fact, in the same way that the matrix used to solve a linear elliptic equation as in §14.5.1 generally is positive or negative definite, when we write a semidiscrete parabolic PDE $\bar{f}' = L\bar{f}$, the matrix L usually is negative definite. This observation implies that $\bar{f}$ solving this continuous ODE is unconditionally stable, since negative eigenvalues are damped out over time.

As outlined in the previous chapter, we have many choices for solving the ODE in time resulting from a spatial discretization. If time steps are small and limited, explicit methods may be acceptable. Implicit solvers often are applied to solving parabolic PDEs; diffusive behavior of implicit Euler may generate inaccuracy but behaviorally speaking appears similar to diffusion provided by the heat equation and may be acceptable even with fairly large time steps. Hyperbolic PDEs may require implicit steps for stability, but advanced integrators such as "symplectic integrators" can prevent oversmoothing caused by these types of steps.

One contrasting approach is to write solutions of semidiscrete systems $\bar{f}' = L\bar{f}$ in terms of eigenvectors of L. Suppose $\vec{v}_1, \ldots, \vec{v}_n$ are eigenvectors of L with eigenvalues $\lambda_1, \ldots, \lambda_n$ and that we know $\bar{f}(0) = c_1\vec{v}_1 + \cdots + c_n\vec{v}_n$. Then, recall that the solution of $\bar{f}' = L\bar{f}$ is given by:

$$\bar{f}(t) = \sum_i c_i e^{\lambda_i t}\vec{v}_i$$

This formula is nothing new beyond §5.1.2, which we introduced during our discussion of eigenvectors and eigenvalues. The eigenvectors of L, however, may have physical meaning in the case of a semidiscrete PDE, as in Example 14.5, which showed that eigenvectors of Laplacians L correspond to different resonant vibrations of the domain. Thus, this eigenvector approach can be applied to develop, for example, "low-frequency approximations" of the initial value data by truncating the sum above over i, with the advantage that t dependence is known exactly without time stepping.

Example 14.10 (Eigenfunctions of the Laplacian)**.** *Figure NUMBER shows eigenvectors of the matrix L from Example 14.7. Eigenvectors with low eigenvalues correspond to* low-frequency *functions on* $[0, 1]$ *with values fixed on the endpoints and can be good approximations of* $f(x)$ *when it is relatively smooth.*

Fully Discrete Methods Alternatively, we might treat the space and time variables more democratically and discretize them both simultaneously. This strategy yields a system of equations to solve more like §14.5.1. This method is easy to formulate by paralleling the elliptic case, but the resulting linear systems of equations can be large if dependence between time steps has a global reach.

Example 14.11 (Fully-discrete heat diffusion)**.** *Explicit, implicit, Crank-Nicolson. Not covered in CS 205A.*

It is important to note that in the end, even semidiscrete methods can be considered fully discrete in that the time-stepping ODE method still discretizes the t variable; the difference is mostly for classification of how methods were derived. One advantage of semidiscrete techniques, however, is that they can adjust the time step for t depending on the current iterate, e.g. if objects are moving quickly in a physical simulation it might make sense to take more time steps and resolve this motion. Some methods even adjust the discretization of the domain of x values in case more resolution is needed near local discontinuities or other artifacts.

14.6 Method of Finite Elements

Not covered in 205A.

14.7 Examples in Practice

In lieu of a rigorous treatment of all commonly-used PDE techniques, in this section we provide examples of where they appear in practice in computer science.

14.7.1 Gradient Domain Image Processing

14.7.2 Edge-Preserving Filtering

14.7.3 Grid-Based Fluids

14.8 To Do

- More on existence/uniqueness
- CFL conditions
- Lax equivalence theorem
- Consistency, stability, and friends

14.9 Problems

- Show 1d Laplacian can be factored as $D^\top D$ for first derivative matrix D
- Solve first-order PDE