Hyperbolic and elliptic equations

There are multiple types of partial differential equations (PDEs). Tackling one equation differs from solving another one. So first we need to look at what kind of equations there are. Then we will try to solve them using numerical techniques.

1 Types of partial differential equations

1.1 Linear, quasi-linear and nonlinear equations

Let's examine a partial differential operator $L(u)$. We say that it is **linear** if

$$L\left(\sum c_i u_i\right) = \sum c_i L(u_i), \tag{1.1}$$

where c_i are constants. If $L(u)$ is not linear, it can be **quasi-linear**. Just look for the highest derivatives. Replace all other terms and coefficients by constants. If the remaining operator satisfies equation (1.1) (and is thus linear), then the original PDE is quasi-linear. In every other case, it is **nonlinear**. Solving these kind of equations is usually hardest.

1.2 Hyperbolic, parabolic and elliptic equations

We can also classify PDEs in hyperbolic, parabolic and elliptic equations. **Hyperbolic** PDEs usually describe phenomena in which features propagate in preffered directions, while keeping its strength (like supersonic flow). **Elliptic** PDEs usually describe phenomena in which features propagate in all directions, while decaying in strength (like subsonic flow). **Parabolic** PDEs are just a limit case of hyperbolic PDEs. We will therefore not consider those.

There is a way to check whether a PDE is hyperbolic or elliptic. For that, we have first have to rewrite our PDE as a system of first-order PDEs. If we can then transform it to a system of ODEs, then the original PDE is hyperbolic. Otherwise it is elliptic.

The above method might sounds a bit vague. S so we'll demonstrate it with an example. Let's consider the equation

$$\frac{\partial^2 u}{\partial x^2} + c\frac{\partial^2 u}{\partial y^2} = 0, \tag{1.2}$$

where c is an unknown constant. We want to know for what c the above equation is hyperbolic, and for which it is elliptic.

First we need to split the equation up into a system of first-order PDEs. So we define $p = \partial u/\partial x$ and $q = \partial u/\partial y$. We can then find two equations. One follows from our differential equation, and the second one is almost per definition true.

$$\frac{\partial p}{\partial x} + c\frac{\partial q}{\partial y} = 0 \qquad \text{and} \qquad \frac{\partial q}{\partial x} - \frac{\partial p}{\partial y} = 0. \tag{1.3}$$

We now examine the linear combinations of the above two equations. In other words, we consider

$$\left(\frac{\partial p}{\partial x} + c\frac{\partial q}{\partial y}\right) + \alpha\left(\frac{\partial q}{\partial x} - \frac{\partial p}{\partial y}\right) = 0. \tag{1.4}$$

We can rewrite this to

$$\left(\frac{\partial}{\partial x} - \alpha\frac{\partial}{\partial y}\right)p + \left(\frac{\partial}{\partial x} + \frac{c}{\alpha}\frac{\partial}{\partial y}\right)(\alpha q) = 0. \tag{1.5}$$

So, we want to know whether we can transform the system of PDEs to a system of ODEs. If we can, then there should be some (real) direction s, in which we can differentiate both p and q (or equivalently, both p and αq). In other words, we should have

$$\frac{\partial}{\partial x} - \alpha\frac{\partial}{\partial y} = \frac{\partial}{\partial x} + \frac{c}{\alpha}\frac{\partial}{\partial y}, \qquad \text{which implies that} \qquad \alpha = \pm\sqrt{-c}. \tag{1.6}$$

What can we conclude from this? If $c > 0$, then there is no real direction in which we can differentiate p and q. So the equation is elliptic. If $c < 0$, then the equation is hyperbolic. We can differentiate p and q in two directions. To find these directions s_1 and s_2, we just insert $\alpha = \sqrt{-c}$ and $\alpha = -\sqrt{-c}$ in equation (1.5). We then find

$$\frac{d}{ds1}(p + \sqrt{c}q) = 0, \qquad \text{where} \qquad \frac{d}{ds1} = \frac{\partial}{\partial x} - \sqrt{-c}\frac{\partial}{\partial y}, \tag{1.7}$$

$$\frac{d}{ds2}(p - \sqrt{-c}q) = 0, \qquad \text{where} \qquad \frac{d}{ds2} = \frac{\partial}{\partial x} + \sqrt{-c}\frac{\partial}{\partial y}. \tag{1.8}$$

Now what does this mean? To find that out, we look at the so-called **Riemann invariants** $p + \sqrt{-c}q$ and $p - \sqrt{-c}q$. These values are constant (invariant) along the lines where $\sqrt{-c}x + y$ and $\sqrt{-c}x - y$ are constant, respectively. By the way, these lines are called the **characteristic lines**.

1.3 Boundary conditions for elliptic and hyperbolic PDEs

Let's suppose we have a rectangle, spanning from $x = 0$ to $x = w$ and from $y = 0$ to $y = h$. Also suppose that this rectangle is subject to the PDE of equation (1.2). When solving for u, we will need boundary conditions. We want enough boundaries to have a unique solution. If this is indeed the case, then we say that the problem is **well-posed**.

First let's ask ourselves, how many boundary conditions do we need? There is a second-order derivative with respect to x, so we need two boundary conditions for given x. This can be something like $u(0, y) = f_1(y)$, $u(w, y) = f_2(y)$ or $\frac{\partial u}{\partial x}(0, y) = f_3(y)$. Similarly, we need two boundary conditions for given y.

Now let's ask ourselves, where should we apply these boundary conditions? This differs for hyperbolic and elliptic PDEs. For hyperbolic equations we should have one side with two boundary conditions. So there usually are two boundary conditions at the line $x = 0$. For elliptic PDEs things are different. If we have an elliptic PDE, then we should have exactly one boundary condition at every side of our rectangle.

2 Numerical methods for hyperbolic equations

2.1 Finite-difference schemes

Let's suppose we have, as an example equation, the PDE

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \tag{2.1}$$

with the constant a being bigger than zero. We want to solve this equation over a certain interval using a finite-difference approximation. To do that, we first have to turn our interval into a **grid**: We divide it in small rectangles. These rectangles span Δt in t-direction and Δx in x-direction. We will now approximate $u(x, t)$ by u^l_m (with l and m integers), where $l = \frac{x}{\Delta x}$ and $m = \frac{t}{\Delta t}$. So we simply find u^l_m, and then we say that $u(x, t) \approx u^l_m$.

But how do we find u^l_m? Well, to do that, we have to use a **finite-difference scheme**. Such a scheme expresses u^l_m in its neighbouring cells. There are many finite-difference schemes for equation (2.1). One

example is

$$\frac{u_m^{l+1} - u_m^l}{\Delta t} + a\frac{u_{m+1}^l - u_{m-1}^l}{\Delta x} = 0. \tag{2.2}$$

It is often handy to visualize which points are used by a finite-difference scheme. A good way to do that, is by making a stencil of the scheme. A stencil of the scheme above can be seen in figure 1.

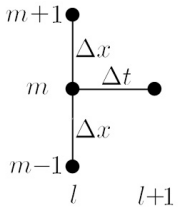

Figure 1: Stencil of a finite difference scheme.

There are multiple types of schemes. If you can express the next time level u_m^{l+1} in previous time levels, we say that the scheme if **explicit (forward) in time**. Otherwise it is **implicit (backward) in time**. (You then need to have an equation with matrices to solve the problem.) If the expressions in x-direction are symmetric with respect to point m (or some other point), then we say the scheme is **central in space**. If this is not the case, then we use data from only one direction. We then call the scheme **upwind in space**. We can see that our example scheme is forward in time. (We can express u_m^{l+1} as a function of $u_{\ldots}^l$.) It is, however, central in space. (The scheme is symmetric about m.)

Once we have selected a scheme, we will use our boundary conditions. We usually know the value of u at the boundaries $x = 0$ and $t = 0$. (So we usually know u_0^l and u_m^0 for all l, m.) Using our finite-difference scheme, we can then find the values of u for $l = 1$, $l = 2$, and so on. At least, this works for a hyperbolic PDE.

You may be wondering, why doesn't it work for an elliptic PDE? Well, we have previously seen that hyperbolic PDEs often don't have a boundary condition at the boundary $x = w$. So, by applying our finite-difference scheme we can just work from left (where $x = 0$) to right (up to $x = w$) without a problem. When we deal with an elliptic PDE, there is also a boundary condition at $x = w$. And we actually need to combine the boundary condition at $x = 0$ and $x = w$! We will take a look on that difficult problem at the end of this chapter. For now we will just stick to our hyperbolic example PDE.

2.2 Testing for consistency

We want our approximation to be **consistent**. This means that, as $\Delta t \to 0$ and $\Delta x \to 0$, our finite-difference scheme converges to the actual PDE. To test for consistency, we use Taylor series expansions. The general equations for Taylor expansions in this case are

$$u_m^{l+a} = u_m^l + a\Delta t\left(\frac{\partial u}{\partial t}\right)_m^l + \frac{a^2}{2}\Delta t^2\left(\frac{\partial^2 u}{\partial t^2}\right)_m^l + \frac{a^3}{6}\Delta t^3\left(\frac{\partial^3 u}{\partial t^3}\right)_m^l + \frac{a^4}{24}\Delta t^4\left(\frac{\partial^4 u}{\partial t^4}\right)_m^l + O(\Delta t^5), \tag{2.3}$$

$$u_{m+a}^l = u_m^l + a\Delta x\left(\frac{\partial u}{\partial x}\right)_m^l + \frac{a^2}{2}\Delta x^2\left(\frac{\partial^2 u}{\partial x^2}\right)_m^l + \frac{a^3}{6}\Delta x^3\left(\frac{\partial^3 u}{\partial x^3}\right)_m^l + \frac{a^4}{24}\Delta x^4\left(\frac{\partial^4 u}{\partial x^4}\right)_m^l + O(\Delta x^5). \tag{2.4}$$

By the way, usually the indices l and m are omitted. We can apply these relations to our finite-difference scheme. We then find

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + \frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2} + a\frac{\Delta x^2}{6}\frac{\partial^3 u}{\partial x^3} + O(\Delta t^2, \Delta x^4) = 0. \tag{2.5}$$

So, as $\Delta t \to 0$ and $\Delta x \to 0$, our finite-difference scheme does converge to our PDE. So it is consistent! By the way, the above equation is called the **modified equation** belonging to the finite-difference scheme.

2.3 Testing for stability and convergence

We also want our scheme to be **stable**. This means that our solution remains bounded. One method to see whether our solution remains bounded is the **Von Neumann stability analysis**. For this we assume that we can write the exact solution to our PDE as

$$u(x,t) = Ue^{\sigma t}e^{i\omega x} \qquad \Rightarrow \qquad u_m^l = Ue^{\sigma l\Delta t}e^{i\omega m\Delta x} = U\rho^l e^{ikm}. \tag{2.6}$$

Here U and ω are given real constants, and σ is still to be determined. We have also defined the **amplification factor** $\rho = e^{\sigma\Delta t}$ and the **grid frequency** $k = \omega\Delta x$.

To apply the stability analysis, we should find ρ, or, more specifically, $|\rho|$. If $|\rho| > 1$, our solution is unstable. If $|\rho| < 1$ it is stable. If $|\rho| = 1$, our solution has neutral stability.

Let's try to find ρ for our example scheme. We insert our new relation for u_m^l in the finite-difference scheme (2.2). After we divide by common terms, we find

$$\rho = 1 - \frac{a\Delta t}{2\Delta x}\left(e^{ik} - e^{-ik}\right) = 1 - i\frac{a\Delta t}{\Delta x}\sin k. \tag{2.7}$$

So ρ is a complex number! (Note that we have used the relations $e^{ik} = \cos k + i\sin k$ and $e^{-ik} = \cos k - i\sin k$.) To find the length $|\rho|$ of ρ, we have to sum up the squares of the real and the complex part, and take the square root of that. We then get

$$|\rho| = \sqrt{(1)^2 + \left(\frac{a\Delta t}{\Delta x}\sin k\right)^2}. \tag{2.8}$$

It turns out that $|\rho|$ is always bigger than 1! So our example scheme isn't stable at all! How annoying. Luckily there are plenty other schemes that are stable. We'll look at those in the next paragraph.

Finally, we also want our finite-difference scheme to be **convergent**. This means that our solution converges to the continuous solution as $\Delta x \to 0$ and $\Delta t \to 0$. Luckily, the **Lax equivalence theorem** states that any scheme that is both consistent and stable is also convergent. This means we don't have to worry about convergence. Great!

2.4 Numerical schemes for our example equation

There are a lot more numerical schemes belonging to our example PDE. And they all got names too! Let's examine a few.

First, there is the **upwind scheme**, reading

$$\frac{u_m^{l+1} - u_m^l}{\Delta t} + a\frac{u_m^l - u_{m-1}^l}{\Delta x} = 0. \tag{2.9}$$

This scheme is stable for $a\frac{\Delta t}{\Delta x} \leq 1$. It is even exact for $a\frac{\Delta t}{\Delta x} = 1$. (This usually means that the finite-difference solution coincides with the actual solution.)

Second, there is the **Lax-Friedrichs scheme**. This one is defined as

$$\frac{2u_m^{l+1} - (u_{m+1}^l - u_{m-1}^l)}{2\Delta t} + a\frac{u_{m+1}^l - u_{m-1}^l}{2\Delta x} = 0. \tag{2.10}$$

This scheme is stable if $-1 \leq a\frac{\Delta t}{\Delta x} \leq 1$.

Third, we have the **Leapfrog scheme**, defined as

$$\frac{u_m^{l+1} - u_m^{l-1}}{2\Delta t} + a\frac{u_{m+1}^l - u_{m-1}^l}{2\Delta x} = 0. \tag{2.11}$$

This scheme has $|\rho| = 1$ if $-1 \leq a\frac{\Delta t}{\Delta x} \leq 1$. So it is neutrally stable. Theoretically this is great. But in practice this often poses difficulties.

There is also another disadvantage to the Leapfrog scheme. To calculate u_m^{l+1} we need data from both the l layer and the $l-1$ layer. Usually this data isn't present in just our boundary conditions. Instead, we first need to user another scheme to acquire enough data. The Leapfrog scheme is thus often combined with the Lax-Friedrichs scheme. Together, they form the **Lax-Wendroff two-step scheme**. First, this scheme uses the Lax-Friedrichs scheme to find the points u_{m+1}^{l+1} and u_{m-1}^{l+1}. It then continous with the Leapfrog scheme to find u_m^{l+2}. It can be derived that we then have

$$\frac{u^{l+2} - u_m^l}{2\Delta t} + \frac{a}{4\Delta x}\left(u_{m+2}^l - u_{m-2}^l\right) - \frac{a^2\Delta t}{4\Delta x^2}\left(u_{m+2}^l - 2u_m^l + u_{m-2}^l\right) = 0. \tag{2.12}$$

3 Numerical methods for elliptic equations

3.1 Implicitly solving the system

Elliptic differential equations are much harder to solve then hyperbolic differential equations. This is because elliptic equations have boundary conditions at all boundaries of the interval. So any finite-difference scheme has to combine multiple boundary conditions at different points. To see how we deal with this, we examine the **Laplace equation** $\nabla^2 u = 0$. (Note that we now don't have a time-derivate, but a derivative w.r.t. y.) A possible finite-difference scheme for this equation is

$$\frac{u_{m+1,n} - 2u_{m,n} + u_{m-1,n}}{2\Delta x} + \frac{u_{m,n+1} - 2u_{m,n} + u_{m,n-1}}{2\Delta y}, \tag{3.1}$$

where $m\Delta x = x$ and $n\Delta y = y$. If we choose $\Delta x = \Delta y = h$, we get

$$u_{m-1,n} + u_{m,n-1} - 4u_{m,n} + u_{m+1,n} + u_{m,n+1} = 0. \tag{3.2}$$

We can't just use four points to find the fifth now. So how do we solve it? One way is by solving for all points simultaneously (we call this **implicit solving**). We do that using a matrix. We first put all (unknown) data points in a vector

$$\mathbf{x} = \begin{bmatrix} \cdots & u_{m,n-1} & \cdots & u_{m-1,n} & u_{m,n} & u_{m+1,n} & \cdots & u_{m,n+1} & \cdots \end{bmatrix}^T. \tag{3.3}$$

Then we can write our system of equations, being

$$A\mathbf{x} = \begin{bmatrix} & & 1 & & & 1 & -4 & 1 & & & 1 & & & \\ & & & 1 & & & 1 & -4 & 1 & & & 1 & & \\ & & & & 1 & & & 1 & -4 & 1 & & & 1 & \\ & & & & & & & & & & & & & \end{bmatrix} \mathbf{x} = \mathbf{0}. \tag{3.4}$$

We then need to solve this equation. Although it is possible, it is quite a lot of work. So usually other methods are prefered.

3.2 Iteration methods

Luckily, there is a slightly easier way to solve this problem. This is by using **iteration methods**. This method uses iterative steps. To apply it, we first choose some values for $u_{m,n}^0$ (for every m, n). We then

refine our initial choice using an iterative equation. One example of such an equation is the **point Jacobi iteration**, given by

$$u^l_{m-1,n} + u^l_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^l_{m,n+1} = 0. \tag{3.5}$$

So during every iteration step the value of $u_{m,n}$ is updated, using the values of its neighbours. This is done for every number $u^{l+1}_{m,n}$, after which the next iteration step commences. And, after a while, the values of $u^l_{m,n}$ will (hopefully) converge.

The point Jacobi iteration methods only 'upgrades' one point every iteration step. There are several other iteration methods, which 'upgrade' more points in one step. For example, there is the **point Gauss-Seidel iteration**. This method has two variants. In the first method a point $u_{m,n}$ and all points to the left-bottom of it are upgraded. (So all $u_{i,j}$ with $i + j \leq m + n$.) In the second method a point $u_{m,n}$ and all points to the right-top are upgraded. (So all $u_{i,j}$ with $i + j \geq m + n$.) This gives us the equations

$$\begin{aligned} u^{l+1}_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^l_{m,n+1} &= 0, & (3.6)\\ u^l_{m-1,n} + u^l_{m,n-1} - 4u^{l+1}_{m,n} + u^{l+1}_{m+1,n} + u^{l+1}_{m,n+1} &= 0. & (3.7) \end{aligned}$$

There is also the **line Jacobi iteration**. Here we iterate over one line every time. This can be either a horizontal or a vertical line. So,

$$\begin{aligned} u^{l+1}_{m-1,n} + u^l_{m,n-1} - 4u^{l+1}_{m,n} + u^{l+1}_{m+1,n} + u^l_{m,n+1} &= 0, & (3.8)\\ u^l_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^{l+1}_{m,n+1} &= 0. & (3.9) \end{aligned}$$

Finally there is the **line Gauss-Seidel iteration**. It is similar to the line Jacobi iteration. But if we iterate over a horizontal line now, we don't only update all points in that line, but also all points below it. Similarly, if we iterate over a vertical line, we also update all points left to it. So we then get the equations

$$\begin{aligned} u^{l+1}_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^{l+1}_{m+1,n} + u^l_{m,n+1} &= 0, & (3.10)\\ u^{l+1}_{m-1,n} + u^{l+1}_{m,n-1} - 4u^{l+1}_{m,n} + u^l_{m+1,n} + u^{l+1}_{m,n+1} &= 0. & (3.11) \end{aligned}$$

3.3 Testing for convergence

The above iteration techniques are nice, if they actually cause $u_{m,n}$ to converge to their actual value. But that isn't always the case. To investigate when this happens, we introduce the **iteration error** $\epsilon^l_{m,n} = u^l_{m,n} - u^*_{m,n}$, where $u^*_{m,n}$ is the exact (converged) solution in point m, n. We assume that

$$\epsilon^l_{m,n} = E\rho^l e^{i(k_1 m + k_2 n)}, \tag{3.12}$$

where k_1 and k_2 are the **grid frequencies** in x- and y-direction, respectively. Also E is a constant and ρ is the amplication factor. We are interested in ρ. Because, if we have found it, we can find the **rate of convergence** R_c using

$$R_c = \ln \frac{1}{|\rho|}. \tag{3.13}$$

So how do we find ρ? We just insert ϵ in our iteration scheme. We can do this for the point Jacobi iteration. We then get

$$E\rho^l e^{i(k_1(m-1)+k_2 n)} + E\rho^l e^{i(k_1 m + k_2(n-1))} - 4E\rho^l e^{i(k_1 m + k_2 n)} + E\rho^l e^{i(k_1(m+1)+k_2 n)} + E\rho^l e^{i(k_1 m + k_2(n+1))} = 0. \tag{3.14}$$

Solving this equation for $|\rho|$ will give

$$|\rho| = \left| \frac{e^{ik_1} + e^{-ik_1} + e^{ik_2} + e^{-ik_2}}{4} \right| = \frac{|\cos k_1 + \cos k_2|}{2}. \tag{3.15}$$

So we see that we always have $|\rho| \leq 1$ for this iteration method, which is nice. It means that eventually the values of $u^l_{m,n}$ will converge to $u^*_{m,n}$.