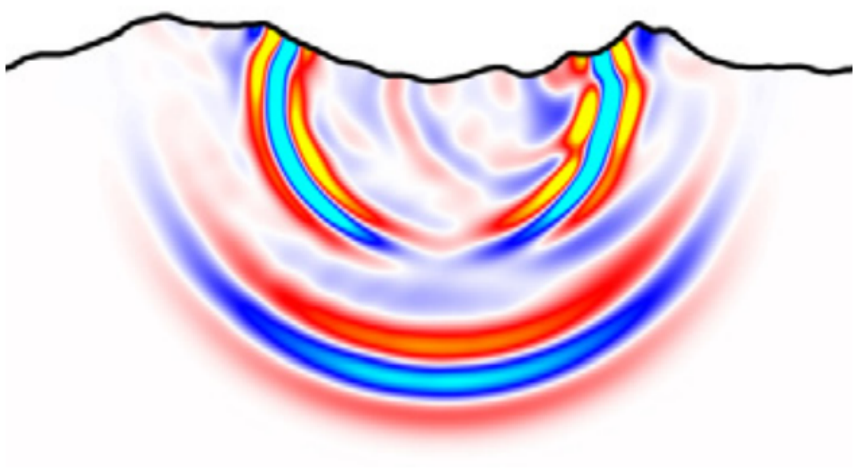
Scientific Computing for Partial Differential Equations

Understanding the models and utilizing efficient numerical methods

Written by

Martin Almquist
Ken Mattsson

Uppsala University
Department of Information Technology
Division of Scientific Computing



February 27, 2024

Contents

1 Well-posedness of scalar PDE in 1D

1.1 The advection equation

One of the simplest examples of a partial differential equation (PDE) is the advection equation:

$$u_t + cu_x = 0, \quad x \in (0, W), \quad t > 0, \tag{1.1}$$

where subscripts denote partial derivatives, $W > 0$ is the domain width, and c is the wave speed (for now assumed to be constant and positive). We will refer to x as a spatial coordinate and t as time, although they could represent other physical quantities. We seek a solution $u = u(x, t)$ such that (1.1) is satisfied.

There are actually *infinitely* many solutions to (1.1). Given any differentiable function f we can construct a solution $u(x, t)$ as

$$u(x, t) = f(x - ct). \tag{1.2}$$

To verify that u satisfies (1.1), we compute its derivatives

$$u_t = -cf'(x - ct), \quad u_x = f'(x - ct) \tag{1.3}$$

and conclude that

$$u_t + cu_x = -cf'(x - ct) + cf'(x - ct) = 0. \tag{1.4}$$

The interpretation of the solution formula (1.2) is that, as time advances, the solution is simply translated (or *advected*) to the right, with speed c.

1.1.1 An introduction to well-posedness

We begin with the following somewhat loose definition of well-posedness, which we will later make more precise:

Definition 1. A PDE-problem is well posed if and only if:

1. A solution *exists*.
2. The solution is *unique*.
3. The solution depends smoothly on provided data (small changes in data lead to small changes in the solution).

Attempting to solve *ill-posed* (not well-posed) problems numerically is almost never worthwile. If no solution exists, then why are we trying to find one? If multiple solutions exist, which one are we hoping to find? Usually, if a PDE-problem is ill-posed, someone made a mistake and should go back to the drawing board.

We do not yet have enough background to discuss requirement 3, so we will revisit it later. Requirement 1 (existence) is often difficult to prove for complex PDE, so in general such proofs are outside the scope of this course. Note, however, that the formula (1.2) proves existence for the advection equation. At the same time, it proves that requirement 2 is violated, because the solution is not unique! As we mentioned, there are infinitely many functions that satisfy (1.1). To ensure uniqueness, we will need to add more constraints so that we single out exactly one solution. Note that we need to be careful—if we add too many constraints there might not be a function that satisfies all of them! That is, we might violate requirement 1.

The usual way to construct a well-posed advection problem is to augment the PDE (1.1) with an *initial condition* and a *boundary condition*.

1.1.2 Initial condition

An initial condition specifies $u(x, t)$ at the initial time, usually taken to be $t = 0$ for simplicity. We can write the initial condition as

$$u(x, 0) = u_0(x), \quad x \in [0, W], \tag{1.5}$$

where u_0 is a *known* function. We refer to u_0 as initial data.

1.1.3 Boundary conditions

In addition to the initial condition we need to specify a boundary condition at the *left* boundary, $x = 0$. The typical boundary condition for the advection equation (with $c > 0$) is

$$u(0, t) = g(t), \quad t > 0, \tag{1.6}$$

where g is a known function. We refer to g as *boundary data*. If $g = 0$, the boundary condition is said to be *homogeneous*, otherwise *inhomogeneous*.

Note that no boundary condition is prescribed at the right boundary. One way to understand this is to recall that waves travel *to the right*. That is, the flow of information is from left to right. The left boundary is an inflow boundary, so we need to specify what is “coming in” from the left. The right boundary is an outflow where information simply leaves the domain, so no boundary condition is needed.

If the wave speed c is negative, the situation is reversed. Waves now travel *to the left*, which means that the left boundary is an outflow and the right boundary is an inflow. In this case we need a condition on the right boundary.

Another possible boundary condition is the periodic condition

$$u(0, t) = u(W, t). \tag{1.7}$$

This condition has the effect that information that leaves the domain at the outflow re-enters at the inflow. It can be thought of as using the solution at the outflow as boundary data for the inflow.

1.1.4 The initial-boundary value problem

We summarize the discussion above by stating the advection equation augmented with appropriate initial and boundary conditions:

$$\begin{aligned} u_t + cu_x &= 0, \quad x \in (0, W), \quad t > 0, \\ u &= u_0, \quad x \in [0, W], \quad t = 0, \\ u &= g, \quad x = 0, \qquad\quad t > 0. \end{aligned} \tag{1.8}$$

The problem (1.8) is an example of an *initial-boundary value problem* (IBVP); it is a PDE augmented with initial and boundary conditions. IBVP are the center of attention in this course. So far we have hinted that this IBVP is well-posed, and we will soon have the tools to prove it.

1.2 Energy analysis

The so-called energy method is a powerful and versatile tool for analyzing IBVP. The point is to derive a bound on the solution in some norm. Such a bound shows that the solution cannot become arbitrarily "large". Before we introduce the energy method, we need to introduce some properties of the space L^2.

1.2.1 L^2 space

Let $L^2(\Omega)$ denote the space of complex-valued functions that are square-integrable over a domain $\Omega \subset \mathbb{R}^n$. That is,

$$L^2(\Omega) = \{f : \Omega \to \mathbb{C} \text{ such that } \int_\Omega |f(\vec{x})|^2 \, \mathrm{d}\vec{x} < \infty\}. \tag{1.9}$$

Let $u, v \in L^2(\Omega)$. The L^2 inner product is defined as

$$(u, v) = \int_\Omega u^*(\vec{x}) v(\vec{x}) \, \mathrm{d}\Omega, \tag{1.10}$$

where $*$ denotes conjugate transpose. The corresponding norm is

$$\|u\|^2 = (u, u)\,. \tag{1.11}$$

The inner product is conjugate symmetric:

$$(u, v) = (v, u)^*\,. \tag{1.12}$$

For real-valued functions $a > 0$, we introduce the weighted inner product and norm

$$(u, v)_a = (u, av)\,, \quad \|u\|_a^2 = (u, u)_a\,. \tag{1.13}$$

The norms $\|\cdot\|$ and $\|\cdot\|_a$ are equivalent in the sense that

$$\|u\|_a^2 \leq a_{max}\|u\|^2, \quad \|u\|^2 \leq \frac{1}{a_{min}}\|u\|_a^2. \tag{1.14}$$

where the constants a_{max} and a_{min} denote the maximum and minimum values of a over Ω.

Remark. We will often consider real-valued functions only, in which case the complex conjugate can be omitted. The conjugate symmetry of the inner product then reduces to symmetry.

1.2.2 The energy method

The energy method consists of the following steps:

1. Identify the highest-order time derivative appearing in the PDE. Suppose that the highest order is qth order. That is, the PDE contains a term $\frac{\partial^q u}{\partial t^q}$. Then take the inner product of the PDE and $\frac{\partial^{q-1} u}{\partial t^{q-1}}$ (one derivative order lower).

2. Take the conjugate transpose of the relation that you derived in step 1. (The transpose operation is only necessary for systems of equations, not scalar equations. Similiarly, the complex conjugate can be omitted if all quantities are real.)

3. Add the relations from step 1 and 2 and integrate by parts in space “as many times as necessary”. This is best explained by examples.

1.2.3 Example: the energy method for the advection equation

Consider the 1D advection equation with variable wave speed $c = c(x) > 0$:

$$\begin{aligned} u_t + cu_x &= 0, \quad x \in (0, W), \quad t > 0, \\ u &= u_0, \quad x \in [0, W], \quad t = 0, \\ u &= g, \quad x = 0, \qquad\quad t > 0. \end{aligned} \tag{1.15}$$

Assume for simplicity that the solution is real. Here, our domain is $\Omega = [0, W]$, so the inner product is simply

$$(u, v) = \int_0^W u(x)v(x)\,\mathrm{d}x. \tag{1.16}$$

The PDE contains a first derivative in time (u_t), so we shall multiply it by u. After some trial and error one realizes that it is clever to first divide the PDE through by c to obtain

$$c^{-1}u_t = -u_x. \tag{1.17}$$

Taking the inner product with u yields

$$\left(u, c^{-1}u_t\right) = -\left(u, u_x\right). \tag{1.18}$$

Since the inner product is symmetric and c is assumed to be constant in *time*, we have

$$\left(u, c^{-1}u_t\right) = \left(u_t, c^{-1}u\right) = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\left(u, c^{-1}u\right) = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{c^{-1}}. \tag{1.19}$$

We can now write (1.18) as

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{c^{-1}} = -\left(u, u_x\right). \tag{1.20}$$

Next, we integrate the right-hand side by parts to obtain

$$-\left(u, u_x\right) = -u^2|_{x=0}^{x=W} + \left(u_x, u\right), \tag{1.21}$$

which simplifies to

$$-\left(u, u_x\right) = -\frac{1}{2}u^2|_{x=0}^{x=W}. \tag{1.22}$$

Using (1.22) in (1.20) leads to

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{c^{-1}} = -\frac{1}{2}u^2|_{x=0}^{x=W}. \tag{1.23}$$

We may define the mathematical energy (the connection to a physical energy is not obvious in this case)

$$E = \|u\|^2_{c^{-1}}, \tag{1.24}$$

which highlights that (1.23) describes the rate of change of energy:

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -u^2|_{x=0}^{x=W}. \tag{1.25}$$

We note that energy can only enter and leave the system via the boundaries. There is no dissipation or growth of energy in the domain interior in this model.

Remark. As we will later see, the energy method usually derives the rate of change of the physical energy of he system, but sometimes the "energy" is a purely mathematical construction. Just like physical energies, mathematical energies are always non-negative quantities.

It is possible to use the energy rate (1.25) to determine how many boundary conditions are required for well-posedness. We have

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -u^2|_{x=0}^{x=W} = -u(W,t)^2 + u(0,t)^2 \leq u(0,t)^2. \tag{1.26}$$

The right boundary (the outflow, at $x = W$) does not contribute to energy growth. Therefore, no boundary condition is required there. To control the growth of energy stemming from the left boundary, we need to specify $u(0,t)$. The boundary condition $u(0,t) = g(t)$ yields the energy rate

$$\frac{\mathrm{d}E}{\mathrm{d}t} \leq g(t)^2. \tag{1.27}$$

Integrating in time yields

$$E(t) \leq E(0) + \int_0^t g(s)^2 \, \mathrm{d}s, \tag{1.28}$$

which we can write as

$$\|u(\cdot,t)\|^2_{c^{-1}} \leq \|u_0\|^2_{c^{-1}} + \int_0^t g(s)^2 \, \mathrm{d}s. \tag{1.29}$$

The bound (1.29) is an *energy estimate*, because it bounds (estimates) the energy at time t in terms of initial and boundary data.

1.2.4 Proof of uniqueness

We may use the energy estimate to prove that any solution to the IBVP (1.15) is unique. Assume that u and v are two solutions and define $w = u - v$. Since both u and v satisfy (1.15), w must satisfy the same IBVP but with homogeneous boundary and initial data ($g = 0, u_0 = 0$). The energy estimate (1.29) yields

$$\|w(\cdot,t)\|^2_{c^{-1}} \leq 0 + 0 = 0. \tag{1.30}$$

This implies that $0 = w = u - v$, which means that u must equal v and hence the solution is unique.

Note that specifying appropriate boundary conditions was essential in proving uniqueness. Without the boundary condition at the left boundary we could not have derived the energy estimate (1.29), and hence not completed this proof.

In this case one boundary condition is enough to guarantee uniqueness. We will see examples of other PDEs that require multiple conditions.

1.3 Well-posedness for general IBVP

Consider the general linear IBVP

$$\begin{aligned} u_t &= \mathcal{D}u + F, & x &\in (0,W), & t &> 0, \\ u &= u_0, & x &\in [0,W], & t &= 0, \\ \mathcal{L}_\ell u &= g_\ell, & x &= 0, & t &> 0, \\ \mathcal{L}_r u &= g_r, & x &= W, & t &> 0, \end{aligned} \tag{1.31}$$

where $\mathcal{D}$ is a differential operator in space, and $\mathcal{L}_{\ell,r}$ are boundary operators at the left and right boundary, respectively. The boundary operators determine the boundary conditions and can, for example, be identity or differential operators. One of them could be zero, if a boundary condition is not needed at that boundary. Note that the PDE is first order in time, but may be of higher order in space (we will discuss PDE that are second order in time later). The restriction that the PDE is linear implies that the operator $\mathcal{D}$ does not depend on u. As usual, $g_{\ell,r} = g_{\ell,r}(t)$ are boundary data functions and $u_0 = u_0(x)$ is the initial data. The function $F = F(x,t)$ is called a *forcing* function. The functions F, $g_{\ell,r}$, and u_0 are assumed to be known.

Gustafsson [2] (p. 32) uses the following definition of well-posedness, which we will henceforth adopt:

Definition 2. The IBVP (1.31) is well-posed if, for $F = 0$, $g_\ell = 0$, $g_r = 0$, there is a unique solution satisfying

$$\|u\| \leq K e^{\alpha t} \|u_0\|, \tag{1.32}$$

where K and α are constants independent of u_0.

The definition states that the IBVP is well-posed if the solution does not grow "too quickly" in time. Note that the definition is quite relaxed in the sense that it allows for exponential growth. This bound on the growth rate ensures that the solution depends smoothly on initial and boundary data, so it replaces Requirement 3 in Definition 1.

Definition 2 is very useful in practice, because it tells us that for well-posedness, it is enough to study the IBVP with $F = 0$ and homogeneous boundary conditions ($g_{\ell,r} = 0$):

$$\begin{aligned} u_t &= \mathcal{D}u, & x &\in (0,W), & t &> 0, \\ u &= u_0, & x &\in [0,W], & t &= 0, \\ \mathcal{L}_\ell u &= 0, & x &= 0, & t &> 0, \\ \mathcal{L}_r u &= 0, & x &= W, & t &> 0 \end{aligned} \tag{1.33}$$

We will refer to (1.33) as *the homogeneous version* (THV) of (1.31). If THV (1.33) is well-posed, then so is the original IBVP (1.31). We will use this result frequently. A hand-waving explanation of the result is that growth that violates well-posedness (faster than exponential) must be caused by a feedback mechanism where a growing solution causes increased growth. Known data F and $g_{\ell,r}$ cannot cause this kind of growth—it must stem from the structure of the homogeneous IBVP.

We will sometimes encounter energy growth of the form

$$\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_a^2 \leq \beta \|u\|_a^2. \tag{1.34}$$

Solving the ODE and using the initial condition leads to the estimate

$$\|u\|_a^2 \leq \|u_0\|_a^2 e^{\beta t}. \tag{1.35}$$

Because the norms $\|\cdot\|_a$ and $\|\cdot\|$ are equivalent (recall (1.14)), we have

$$\|u\|^2 \leq \frac{1}{a_{min}}\|u\|_a^2 \leq \frac{1}{a_{min}}\|u_0\|_a^2 e^{\beta t} \leq \frac{a_{max}}{a_{min}}\|u_0\|^2 e^{\beta t} \tag{1.36}$$

Taking the square root yields the desired estimate

$$\|u\| \leq K e^{\alpha t} \|u_0\| \tag{1.37}$$

with

$$K = \sqrt{\frac{a_{max}}{a_{min}}}, \quad \alpha = \frac{\beta}{2}. \tag{1.38}$$

This shows that an estimate of the form (1.34) is sufficient for well-posedness.

1.3.1 Well-posed boundary conditions

For a general, linear, *scalar* PDE of the form

$$u_t = Du + F \tag{1.39}$$

where the spatial differential operator D is of order r, we should always impose exactly r boundary conditions in total. In general, for systems of linear PDE, where u is a vector with m components, we should always impose *at most* rm conditions. There are systems with singular matrices for which fewer conditions should be imposed.

With too few conditions, the solution will not be unique. With too many conditions, no solution will exist. So, to ensure well-posedness, we need to select the *minimial number* of boundary conditions that allows a bound of the form (1.32).

1.4 The heat equation

The heat equation in 1D reads

$$\rho c_p u_t = (\kappa u_x)_x + F, \quad x \in (0, W), \quad t > 0, \tag{1.40}$$

where u is temperature, $\kappa = \kappa(x)$ is the thermal conductivity, c_p is specific heat capacity and ρ is density. Let us for simplicity assume that ρ and c_p are constant, in which case the equation can be written as

$$u_t = (au_x)_x + G, \quad x \in (0, W), \quad t > 0, \tag{1.41}$$

where the coefficient $a > 0$ is called the thermal diffusivity. The heat equation is identical to the *diffusion* equation, where a would instead be the diffusion coefficient.

Exercise: Which of the following sets of boundary conditions are well-posed for the heat equation? The coefficient b in condition e) is a real constant.

a) $u(0,t) = 0, \quad u(W,t) = 0$ (Dirichlet conditions on both sides)

b) $u_x(0,t) = 0, \quad u_x(W,t) = 0$ (Neumann conditions on both sides)

c) $u(0,t) = 0, \quad u_x(W,t) = 0$ (Dirichlet on one side, Neumann on the other)

d) $u(0,t) = 0, \quad u_x(0,t) = 0$ (two conditions on one side)

e) $u(0,t) = 0, \quad a(W)u_x(W,t) + bu(W,t) = 0,$ (Dirichlet + Robin)

f) $u(0,t) = 0$

g) $u(0,t) = 0, \quad u(W,t) = 0, \quad u_x(W,t) = 0$

Solution: We can immediately say that f) and g) are ill-posed. The PDE is linear, scalar, and of second order in space. We need exactly 2 boundary conditions.

We know that when it comes to well-posedness, it is sufficient to study THV, so we set the forcing function G to zero. We will apply the energy method. The PDE contains u_t, so we multiply by u. Here it is reasonable to assume that all quantities are real. We obtain, using integration by parts:

$$(u, u_t) = (u, (au_x)_x) = uau_x|_0^W - (u_x, au_x)\,. \tag{1.42}$$

In this case it is not necessary to add the complex conjugate. Note that

$$(u, u_t) = \frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2, \quad (u_x, au_x) = \|u_x\|_a^2. \tag{1.43}$$

We obtain the energy rate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = uau_x|_0^W - \|u_x\|_a^2 \le uau_x|_0^W\,. \tag{1.44}$$

Note that $\|u\|^2$ does not correspond to the physical energy, so this energy rate is purely mathematical. The term $-\|u_x\|_a^2$ corresponds to dissipation. It is characteristic of the heat/diffusion equations that sharp features of u are dissipated quickly. The solution tends to flatten out.

Writing out the boundary terms explicitly yields

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 \le a(W)u(W,t)u_x(W,t) - a(0)u(0,t)u_x(0,t) \tag{1.45}$$

Any set of exactly two boundary conditions that allows us to bound the growth of u according to Definition 2 yields a well-posed IBVP.

For a)-c), we obtain

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 \le a(W)u(W,t)u_x(W,t) - a(0)u(0,t)u_x(0,t) = 0. \tag{1.46}$$

The boundary conditions do not produce growth or dissipation. They preserve (mathematical) energy. We see that $\|u\|$ does not grow with time, so the IBVP is clearly well posed.

For d), we obtain

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 \le a(W)u(W,t)u_x(W,t). \tag{1.47}$$

Here it is not possible to bound the growth rate, so we cannot prove well-posedness. It is nontrivial to prove that the IBVP is actually ill-posed, but it is, and it would be a good guess to say that it is.

For e), we obtain

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 \le a(W)u(W,t)u_x(W,t) = -bu(W,t)^2. \tag{1.48}$$

The Dirichlet condition at the left boundary conserves energy. The boundary condition $au_x + bu = 0$ at the right boundary yields dissipation (and is therefore well posed) for $b > 0$. This is called a *Robin* boundary condition. For $b = 0$, it reduces to a Neumann condition. For $b < 0$, the condition yields growth and is ill-posed.

1.5 The Schrödinger equation

The Schrödinger equation for a single particle in one dimension reads

$$i\hbar u_t = -\frac{\hbar^2}{2m} u_{xx} + Vu, \quad x \in (0, W), \quad t > 0. \tag{1.49}$$

where u is the wave function, m is the particle's mass, $V = V(x,t) \in \mathbb{R}$ is the potential, and $\hbar$ is the reduced Planck constant. The equation is often posed on the real line ($x \in (-\infty, \infty)$), but here we assume that it is posed on a finite domain.

Before applying the energy method, it is convenient to divide through by $i\hbar$ to obtain

$$u_t = i\frac{\hbar}{2m} u_{xx} + \frac{i}{\hbar} Vu. \tag{1.50}$$

We now multiply by u^* (since there are complex quantities the complex conjugate is important) and integrate to obtain

$$(u, u_t) = i\frac{\hbar}{2m}(u, u_{xx}) + \frac{i}{\hbar}(u, Vu). \tag{1.51}$$

Integrating by parts leads to

$$(u, u_t) = i\frac{\hbar}{2m} u^* u_x|_0^W - i\frac{\hbar}{2m}(u_x, u_x) + \frac{i}{\hbar}(u, Vu). \tag{1.52}$$

Taking the complex conjugate of (1.52) yields

$$(u_t, u) = -i\frac{\hbar}{2m} u_x^* u|_0^W + i\frac{\hbar}{2m}(u_x, u_x) - \frac{i}{\hbar}(u, Vu). \tag{1.53}$$

Adding (1.52) and (1.53) yields the energy rate

$$\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = i\frac{\hbar}{2m}(u^* u_x - u_x^* u)|_0^W. \tag{1.54}$$

Notice that all the volume integrals cancelled and we are left with only boundary terms. The "energy" $\|u\|^2$ is not actually the physical energy, but rather the probability of finding the particle in the domain $(0, W)$, so at least it is an important physical quantity.

Exercise: Determine a set of well-posed (non-periodic) boundary conditions that corresponds to

a) conservation of probability.

b) dissipation of probability at both boundaries.

Solution: The PDE is second order in space, so we need two boundary conditions in total.

For a), imposing $u = 0$ at both boundaries leads to conservation of probability:

$$\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = 0. \tag{1.55}$$

The same holds for $u_x = 0$ at both boundaries, as well as $u = 0$ at one and $u_x = 0$ at the other.

For b), we note that the boundary terms are products of u and u_x. To control the sign of these terms, we would like to be able to write them as $C|u|^2$ or $C|u_x|^2$. We therefore try the boundary conditions

$$u - au_x = 0,\ x = 0, \quad u - bu_x = 0,\ x = W, \tag{1.56}$$

where the coefficients a, b are to be determined. We obtain

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 &= i\frac{\hbar}{2m}(u^* u_x - u_x^* u)|_0^W \\
&= i\frac{\hbar}{2m}\left(u_x^* b^* u_x - u_x^* b u_x\right)\bigg|_{x=W} - i\frac{\hbar}{2m}\left(u_x^* a^* u_x - u_x^* a u_x\right)\bigg|_{x=0} \\
&= i\frac{\hbar}{2m}(b^* - b)|u_x|^2\bigg|_{x=W} - i\frac{\hbar}{2m}(a^* - a)|u_x|^2\bigg|_{x=0}
\end{aligned} \tag{1.57}$$

Note that $i(a^* - a) = 2\mathcal{I}(a)$, where $\mathcal{I}$ denotes the imaginary part. For dissipation of probability, we need $\mathcal{I}(a) > 0$ and $\mathcal{I}(b) < 0$. The real part does not cause dissipation or growth, and therefore does not affect well-posedness.

2 Well-posedness of first-order systems and the second-order wave equation in 1D

2.1 First order hyperbolic systems

Consider the following system of linear PDE:

$$\mathbf{u}_t = A\mathbf{u}_x + \mathbf{F}, \quad x \in (0, W), \quad t > 0, \tag{2.1}$$

where the solution vector $\mathbf{u} = [u_1, \ldots, u_m]^T$ is a vector-valued function with m components and $A = A(x)$ is an $m \times m$ matrix.

Example: A system of $m = 2$ equations is given by

$$\mathbf{u}(x,t) = \begin{bmatrix} u_1(x,t) \\ u_2(x,t) \end{bmatrix}, \quad A(x) = \begin{bmatrix} a_{11}(x) & a_{12}(x) \\ a_{21}(x) & a_{22}(x) \end{bmatrix}, \quad \mathbf{F}(x,t) = \begin{bmatrix} F_1(x,t) \\ F_2(x,t) \end{bmatrix}. \tag{2.2}$$

This system can equivalently be written

$$\begin{cases} u_{1,t} = a_{11}u_{1,x} + a_{12}u_{2,x} + F_1 \\ u_{2,t} = a_{21}u_{1,x} + a_{22}u_{2,x} + F_2 \end{cases}. \tag{2.3}$$

Definition 3. The PDE (2.1) is *hyperbolic* if $A(x)$ is diagonalizable and has only real eigenvalues for all $x \in [0, W]$.

Recall that hyperbolic PDEs describe wave propagation. The fact that A can be diagonalized with real eigenvalues means that we have

$$A(x) = T(x)\Lambda(x)T(x)^{-1}, \tag{2.4}$$

where Λ is a diagonal matrix that holds the eigenvalues of A. That is, if λ_i denotes the ith eigenvalue of A, then

$$\Lambda_{ii} = \lambda_i. \tag{2.5}$$

If A is Hermitian, then all its eigenvalues are real and (2.4) holds with a unitary matrix T (implying $T^{-1} = T^*$), yielding

$$A(x) = T(x)\Lambda(x)T(x)^*. \tag{2.6}$$

2.1.1 Constant coefficients

If A is constant we can decouple the equations in (2.1) by a change of variables. Let $\mathbf{w} = T^{-1}\mathbf{u}$. Multiplying (2.1) by T^{-1} from the left yields

$$T^{-1}\mathbf{u}_t = T^{-1}ATT^{-1}\mathbf{u}_x + T^{-1}\mathbf{F}, \tag{2.7}$$

where we used that $TT^{-1} = I$. Since A is assumed to be constant, so is T. This implies that $T^{-1}\mathbf{u}_x = (T^{-1}\mathbf{u})_x = \mathbf{w}_x$. Using also that $T^{-1}AT = \Lambda$ and introducing $\tilde{\mathbf{F}} = T^{-1}\mathbf{F}$, we obtain

$$\mathbf{w}_t = \Lambda\mathbf{w}_x + \tilde{\mathbf{F}}. \tag{2.8}$$

Notice that we have decoupled the equations, since the matrix Λ is diagonal! Every equation in the system (2.8) is a scalar advection equation. The new variables (the components of $\mathbf{w}$), which satisfy advection equations, are called the *characteristic variables*. The wave speeds are given by the eigenvalues of A. The i:th equation in (2.8) is (assuming zero forcing, for simplicity)

$$w_{i,t} = \lambda_i w_{i,x}. \tag{2.9}$$

Notice that this corresponds to an advection equation with wave speed $-\lambda_i$.

Exercise: Consider the problem

$$\begin{aligned} \mathbf{u}_t = A\mathbf{u}_x + \mathbf{F}, \quad & x \in (0, W), \quad t > 0, \\ \mathbf{u} = \mathbf{u}_0, \quad & x \in [0, W], \quad t = 0, \end{aligned} \tag{2.10}$$

Suppose that A is constant and diagonalizable with only real eigenvalues, with m_+ positive eigenvalues, m_- negative eigenvalues, and m_0 zero eigenvalues, where $m_+ + m_- + m_0 = m$. How many (scalar) boundary conditions must be imposed at each boundary for (2.10) to be well posed? Any linear boundary condition can be written in the form

$$\sum_{i=1}^{m} \alpha_i u_i = g, \tag{2.11}$$

for some coefficients α_i and boundary data g.

Solution: This question can be answered with very few calculations. We know that the forcing function F does not affect well-posedness, so we will study the homogeneous version. The PDE can be transformed into diagonal form:

$$\mathbf{w}_t = \Lambda\mathbf{w}_x. \tag{2.12}$$

Since the problems are equivalent, if one of them is well posed, then so is the other. It is therefore enough to study the diagonal form. Equation number i reads

$$w_{i,t} = \lambda_i w_{i,x}. \tag{2.13}$$

If $\lambda_i > 0$, this corresponds to an advection equation with left-going waves. We know that a BC is required on the inflow, the right boundary. Conversely, if $\lambda_i < 0$, a BC is required on the left boundary. If $\lambda_i = 0$, the PDE collapses to

$$w_{i,t} = 0 \tag{2.14}$$

which simply states that w_i is constant in time. No BC is required for this equation.

Conclusion: We need to impose m_+ BC on the right boundary and m_- BC on the left boundary, for a total of $m - m_0$ BC.

Remark. Note that there are typically many different sets of well-posed BC. They do not have to be of the simple form $w_i = 0$. The discussion above just answers *how many* BC are required at each boundary.

2.1.2 Energy analysis with constant coefficients

Consider the IBVP

$$\begin{aligned} \mathbf{u}_t &= A\mathbf{u}_x, & x &\in (0,W), & t &> 0, \\ \mathbf{u} &= \mathbf{u}_0, & x &\in [0,W], & t &= 0, \\ \mathcal{L}_\ell \mathbf{u} &= g_\ell, & x &= 0, & t &> 0, \\ \mathcal{L}_r \mathbf{u} &= g_r, & x &= W, & t &> 0, \end{aligned} \tag{2.15}$$

where the boundary conditions (described by $\mathcal{L}_{\ell,r}$) are yet to be prescribed. The diagonal form of the PDE and IC is

$$\begin{aligned} \mathbf{w}_t &= \Lambda \mathbf{w}_x, & x &\in (0,W), & t &> 0, \\ \mathbf{w} &= T^{-1}\mathbf{u}_0, & x &\in [0,W], & t &= 0. \end{aligned} \tag{2.16}$$

where $\mathbf{w} = T^{-1}\mathbf{u}$. Let us apply the energy method to the diagonal form. That is, we multiply by $\mathbf{w}^*$ from the left and integrate in space:

$$(\mathbf{w}, \mathbf{w}_t) = (\mathbf{w}, \Lambda \mathbf{w}_x) = [IBP] = \mathbf{w}^* \Lambda \mathbf{w}|_0^W - (\mathbf{w}_x, \Lambda \mathbf{w}) \,. \tag{2.17}$$

The conjugate transpose is

$$(\mathbf{w}_t, \mathbf{w}) = (\mathbf{w}_x, \Lambda^* \mathbf{w}) \,. \tag{2.18}$$

Adding the two equations above yields the energy rate

$$\frac{\mathrm{d}}{\mathrm{d}t} \|\mathbf{w}\|^2 = \mathbf{w}^* \Lambda \mathbf{w}|_0^W + (\mathbf{w}_x, (\Lambda^* - \Lambda)\mathbf{w}) \,. \tag{2.19}$$

Recall that the PDE is hyperbolic only if all eigenvalues of A are real, which implies $\Lambda^* = \Lambda$. We now obtain only boundary terms in the energy rate, which we can write as follows:

$$\frac{\mathrm{d}}{\mathrm{d}t} \|\mathbf{w}\|^2 = \mathbf{w}^* \Lambda \mathbf{w}|_0^W = \sum_{i=1}^{m} \lambda_i |w_i|^2 \Bigg|_0^W = \left[\sum_{i \in I_+} \lambda_i |w_i|^2 + \sum_{i \in I_-} \lambda_i |w_i|^2 \right]_0^W , \tag{2.20}$$

where I_+ and I_- denote the set of indices corresponding to positive and negative eigenvalues, i.e.

$$I_+ = \{i : \lambda_i > 0\}, \quad I_- = \{i : \lambda_i < 0\}. \tag{2.21}$$

For well-posedness we need to impose the minimal number of BC ($m - m_0$ conditions, as discussed above) that allows us to bound the right-hand side in the homogeneous version (zero boundary data). One option that yields dissipation in THV is so-called *characteristic boundary conditions*, which amounts to specifiying the ingoing characteristic variables:

$$w_i = g_i \; \forall i \in I_- \text{ at } x = 0, \quad w_i = g_i \; \forall i \in I_+ \text{ at } x = W. \tag{2.22}$$

This corresponds to the usual inflow boundary condition that we saw for the scalar advection equation, applied to each of the characteristic variables separately. After imposing characteristic BC, the energy rate (2.20) becomes (with zero boundary data)

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{w}\|^2 = -\sum_{i\in I_+}\lambda_i|w_i|^2\Big|_0 + \sum_{i\in I_-}\lambda_i|w_i|^2\Big|_W \leq 0. \tag{2.23}$$

Integrating in time leads to the energy estimate

$$\|\mathbf{w}(\cdot,t)\|^2 \leq \|\mathbf{w}(\cdot,0)\|^2 = \|T^{-1}\mathbf{u}_0\|^2. \tag{2.24}$$

We can transform this bound back to the original variables, $\mathbf{u}$. We have

$$\|\mathbf{u}\| = \|T\mathbf{w}\| \leq \|T\|\|\mathbf{w}\| \leq \|T\|\|T^{-1}\mathbf{u}_0\| \leq \|T\|\|T^{-1}\|\|\mathbf{u}_0\| = cond(T)\|\mathbf{u}_0\|, \tag{2.25}$$

which shows well-posedness in the original variables too. The norm of $\mathbf{u}$ may grow, but only by a factor of the condition number of T.

Remark. Characteristic boundary conditions are not the only well-posed option. There are usually many options. In fact, one can show that any set of conditions that uses the outgoing characteristic variables as data for one of the ingoing characterstic variables is well-posed. Thas is, conditions of the form

$$w_i = \sum_{j\in I_+}\alpha_{ij}w_j + g_i \;\forall i\in I_- \text{ at } x=0, \quad w_i = \sum_{j\in I_-}\beta_{ij}w_j + g_i \;\forall i\in I_+ \text{ at } x=W, \tag{2.26}$$

are well posed.

2.1.3 The acoustic wave equation with variable material parameters

As a first example of a hyperbolic system with variable coefficients, let us study the acoustic wave equation:

$$\begin{aligned} \rho v_t + p_x &= 0, \quad x\in(0,W), \quad t>0,\\ \tfrac{1}{K}p_t + v_x &= 0, \quad x\in(0,W), \quad t>0,\\ v &= v_0, \quad x\in[0,W], \quad t=0,\\ p &= p_0, \quad x\in[0,W], \quad t=0, \end{aligned} \tag{2.27}$$

where the material parameters are density ($\rho = \rho(x) > 0$) and bulk modulus ($K = K(x) > 0$). The unknowns v and p are particle velocity and pressure, respectively. We will discuss appropriate boundary conditions later.

Let $\mathbf{u} = [v,p]^T$. The acoustic wave equation can be written in matrix-vector form:

$$\begin{aligned} C\mathbf{u}_t &= B\mathbf{u}_x, \quad x\in(0,W), \quad t>0,\\ \mathbf{u} &= \mathbf{u}_0, \quad x\in[0,W], \quad t=0, \end{aligned} \tag{2.28}$$

where $C = C(x)$ is Hermitian positive definite and B is constant and Hermitian. These properties are important. In fact, Maxwell's equations and the elastic wave equation can also be written in the form of (2.28). Most of the following analysis is therefore generally applicable.

In the specific case of the acoustic wave equation, we have

$$C = \begin{bmatrix} \rho & \\ & K^{-1} \end{bmatrix}, \quad B = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}. \tag{2.29}$$

Applying the energy method to (2.28) yields:

$$(\mathbf{u}, \mathbf{u}_t)_C = (\mathbf{u}, B\mathbf{u}_x) = [IBP] = \mathbf{u}^* B \mathbf{u}|_0^W - (\mathbf{u}_x, B\mathbf{u}). \tag{2.30}$$

The conjugate transpose is:

$$(\mathbf{u}_t, \mathbf{u})_C = (\mathbf{u}_x, B^* \mathbf{u}). \tag{2.31}$$

Adding the two equations, using $B = B^*$ (and dividing by 2), yields

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{u}\|_C^2 = \frac{1}{2}\mathbf{u}^* B \mathbf{u}|_0^W. \tag{2.32}$$

Finally, this a *physical* energy rate! For the acoustic wave equation, we have the energy

$$\frac{1}{2}\|\mathbf{u}\|_C^2 = \frac{1}{2}(v, \rho v) + \frac{1}{2}\left(p, K^{-1}p\right) =: E, \tag{2.33}$$

where $\frac{1}{2}(v, \rho v)$ is the kinetic energy and $\frac{1}{2}(p, K^{-1}p)$ is the potential energy. We further have

$$\frac{1}{2}\mathbf{u}^* B \mathbf{u}|_0^L = -pv|_0^L \tag{2.34}$$

which can be interpreted as a *work rate* (units of power), i.e., work being done on the system per unit time. The energy rate reads

$$\frac{\mathrm{d}E}{\mathrm{d}t} = -pv|_0^W. \tag{2.35}$$

Boundary conditions that lead to energy conservation, and thus well-posedness, are:

$$v = 0 \text{ or } p = 0, \; x = 0, \quad v = 0 \text{ or } p = 0, \; x = W. \tag{2.36}$$

There are many other well-posed BC.

2.1.4 Characteristic variables of the acoustic wave equation

The aoustic wave equation can be written in the form $\mathbf{u}_t = A\mathbf{u}_x$, where

$$A = C^{-1}B = \begin{bmatrix} 0 & -\rho^{-1} \\ -K & 0 \end{bmatrix}. \tag{2.37}$$

The eigenvalues of A are $\pm c$, where $c = \sqrt{K/\rho}$ is the sound speed. A can be diagonalized by the matrix T:

$$T^{-1} = \begin{bmatrix} \rho c & 1 \\ -\rho c & 1 \end{bmatrix}, \quad T = \frac{1}{2}\begin{bmatrix} \frac{1}{\rho c} & -\frac{1}{\rho c} \\ 1 & 1 \end{bmatrix} \tag{2.38}$$

Proof:

$$\begin{aligned} T\Lambda T^{-1} &= \frac{1}{2}\begin{bmatrix} \frac{1}{\rho c} & -\frac{1}{\rho c} \\ 1 & 1 \end{bmatrix}\begin{bmatrix} -c & \\ & c \end{bmatrix}\begin{bmatrix} \rho c & 1 \\ -\rho c & 1 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \frac{1}{\rho c} & -\frac{1}{\rho c} \\ 1 & 1 \end{bmatrix}\begin{bmatrix} -\rho c^2 & -c \\ -\rho c^2 & c \end{bmatrix} \\ &= \begin{bmatrix} 0 & -\rho^{-1} \\ -K & 0 \end{bmatrix} = A. \end{aligned} \tag{2.39}$$

Since A has real eigenvalues and can be diagonalized, the acoustic wave equation is a *hyperbolic* PDE. The characteristic variables are given by $\mathbf{w} = T^{-1}\mathbf{u}$ and are $p \pm \rho c v$. Note however, that the PDE can only be written in diagonal form if ρ and K are constant.

2.1.5 General hyperbolic systems with variable coefficients

As mentioned above, the acoustic and elastic wave equations as well as Maxwell's equations with variable material properties can all be written in the form

$$C\mathbf{u}_t = B\mathbf{u}_x \tag{2.40}$$

where $C = C(x)$ is Hermitian positive definite and B is constant and Hermitian. The energy method yields the energy rate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{u}\|_C^2 = \frac{1}{2}\mathbf{u}^* B\mathbf{u}|_0^W. \tag{2.41}$$

To determine how many BC are needed at each boundary, it is sufficient to study the eigenvalues of B. Note that B is assumed to be Hermitian and therefore has only real eigenvalues and can be diagonalized by a unitary matrix T, i.e.,

$$B = T\Lambda T^*. \tag{2.42}$$

We can write the energy rate as

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{u}\|_C^2 = \frac{1}{2}\mathbf{u}^* B\mathbf{u}\Big|_0^W = \frac{1}{2}\mathbf{u}^* T\Lambda T^*\mathbf{u}\Big|_0^L = \frac{1}{2}\mathbf{w}^*\Lambda\mathbf{w}\Big|_0^W = \frac{1}{2}\sum_{i=1}^m \lambda_i |w_i|^2\Big|_0^W, \tag{2.43}$$

where $\mathbf{w} = T^*\mathbf{u}$ and λ_i are the eigenvalues of B. This shows that for every positive eigenvalue we need one BC at $x = W$ and for every negative eigenvalue we need one at $x = 0$. Zero eigenvalues do not require a BC.

2.2 The acoustic wave equation in second-order form

The acoustic wave equation can be written as a scalar, second-order PDE (often called simply “the wave equation”) and is often solved in this form. One way to derive the second-order form is to start from the first-order system:

$$\begin{aligned} \rho v_t &= -p_x, \\ \tfrac{1}{K} p_t &= -v_x, \end{aligned} \tag{2.44}$$

and then apply ∂_x to the first equation and ∂_t to the second, yielding

$$\begin{aligned} v_{tx} &= -(\rho^{-1} p_x)_x, \\ \tfrac{1}{K} p_{tt} &= -v_{xt}. \end{aligned} \tag{2.45}$$

Eliminating v yields a scalar equation for p:

$$\frac{1}{K} p_{tt} = (\rho^{-1} p_x)_x. \tag{2.46}$$

It is common to solve this equation for p. However, this form of the equation suffers from the annoying property that the energy method does not yield the physical energy. It might therefore be preferrable to introduce the *momentum potential*, ϕ, which by definition satisfies

$$\phi_x = \rho v, \quad \phi_t = -p. \tag{2.47}$$

Using (2.47) and (2.44), we can derive

$$\frac{1}{K} \phi_{tt} = -\frac{1}{K} p_t = v_x = (\rho^{-1} \phi_x)_x, \tag{2.48}$$

i.e.,

$$\frac{1}{K} \phi_{tt} = (\rho^{-1} \phi_x)_x. \tag{2.49}$$

Notice that ϕ satisfies the same PDE as p.

Remark. If ρ is constant the acoustic wave equation can be written in the canonical form

$$\phi_{tt} = c^2 \phi_{xx}, \tag{2.50}$$

where $c = \sqrt{K/\rho}$ is the sound speed.

2.2.1 Energy analysis

Since the PDE contains ϕ_{tt}, the energy method starts by mutiplying with ϕ_t. Let us assume that all quantities are real. We have

$$\underbrace{\left(\phi_t, K^{-1} \phi_{tt}\right)}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(\phi_t, K^{-1}\phi_t)} = \left(\phi_t, (\rho^{-1} \phi_x)_x\right) = [IBP] = \phi_t \rho^{-1} \phi_x \Big|_0^W - \underbrace{\left(\phi_{tx}, \rho^{-1} \phi_x\right)}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}(\phi_x, \rho^{-1}\phi_x)}. \tag{2.51}$$

We obtain the energy rate

$$\frac{\mathrm{d}E}{\mathrm{d}t} = \phi_t \rho^{-1} \phi_x \big|_0^W , \tag{2.52}$$

where the energy is

$$E = \frac{1}{2}\left(\phi_t, K^{-1}\phi_t\right) + \frac{1}{2}\left(\phi_x, \rho^{-1}\phi_x\right). \tag{2.53}$$

Using the definition of the momentum potential, we can show that this energy rate is identical to what we obtained for the first-order system, given by (2.35) and (2.33).

We know that we need two BC in total because the PDE is scalar and second order in space. BC that yield energy conservation (with zero boundary data) are

$$\phi = 0 \text{ or } \phi_x = 0, \; x = 0, \quad \phi = 0 \text{ or } \phi_x = 0, \; x = W. \tag{2.54}$$

2.2.2 Well-posed IBVP

For PDE that are second order in time, we need two initial conditions. An IBVP for the wave equation is thus

$$\begin{aligned}
\tfrac{1}{K}\phi_{tt} = (\rho^{-1}\phi_x)_x + F, &\quad x \in (0,W), &\quad t > 0,\\
\phi = \phi_0, &\quad x \in [0,W], &\quad t = 0,\\
\phi_t = \varphi_0, &\quad x \in [0,W], &\quad t = 0,\\
\mathcal{L}_\ell \phi = g_\ell, &\quad x = 0, &\quad t > 0,\\
\mathcal{L}_r \phi = g_r, &\quad x = W, &\quad t > 0.
\end{aligned} \tag{2.55}$$

This IBVP is well-posed if the boundary operators $\mathcal{L}_{\ell,r}$ are such that the homogeneous version of the IBVP (where $F = 0$, $g_{\ell,r} = 0$) does not allow for energy growth, i.e., the energy rate satisfies

$$\frac{\mathrm{d}E}{\mathrm{d}t} \leq 0. \tag{2.56}$$

3 The energy method in 2D and 3D

3.1 Integration by parts in higher dimensions

Let Ω denote a bounded domain in $\mathbb{R}^n$, with boundary $\partial\Omega$ and outward unit normal $\hat{\mathbf{n}}$. For a (sufficiently smooth) scalar function $u \in L^2(\Omega)$ and a (sufficiently smooth) vector field $\mathbf{v} \in L^2(\Omega)$, we have

$$\int\limits_{\Omega} u \nabla \cdot \mathbf{v} \, \mathrm{d}\Omega = \oint\limits_{\partial\Omega} u (\mathbf{v} \cdot \hat{\mathbf{n}}) \, \mathrm{d}S - \int\limits_{\Omega} \nabla u \cdot \mathbf{v} \, \mathrm{d}\Omega. \tag{3.1}$$

This is a consequence of the divergence theorem and can be viewed as a higher-dimensional version of the integration-by-parts formula. Substituting $\mathbf{v} = \alpha \nabla w$ in (3.1), where α and w are scalar functions, yields the following useful relation:

$$\int\limits_{\Omega} u \nabla \cdot \alpha \nabla w \, \mathrm{d}\Omega = \oint\limits_{\partial\Omega} u (\alpha \nabla w \cdot \hat{\mathbf{n}}) \, \mathrm{d}S - \int\limits_{\Omega} \nabla u \cdot \alpha \nabla w \, \mathrm{d}\Omega. \tag{3.2}$$

The normal derivative $\frac{\partial}{\partial \hat{\mathbf{n}}}$ is defined by

$$\frac{\partial w}{\partial \hat{\mathbf{n}}} = \nabla w \cdot \hat{\mathbf{n}}. \tag{3.3}$$

We can therefore write (3.2) as

$$\int\limits_{\Omega} u \nabla \cdot \alpha \nabla w \, \mathrm{d}\Omega = \oint\limits_{\partial\Omega} u \alpha \frac{\partial w}{\partial \hat{\mathbf{n}}} \, \mathrm{d}S - \int\limits_{\Omega} \nabla u \cdot \alpha \nabla w \, \mathrm{d}\Omega. \tag{3.4}$$

Introducing the inner product for (possibly) vector-valued functions $\boldsymbol{\phi}, \boldsymbol{\psi}$ in $L^2(\Omega)$,

$$(\boldsymbol{\phi}, \boldsymbol{\psi}) = \int\limits_{\Omega} \boldsymbol{\phi}^* \boldsymbol{\psi} \, \mathrm{d}\Omega, \tag{3.5}$$

we can write (3.1) and (3.4) as

$$(u, \nabla \cdot \mathbf{v}) = \oint\limits_{\partial\Omega} u^* (\mathbf{v} \cdot \hat{\mathbf{n}}) \, \mathrm{d}S - (\nabla u, \mathbf{v}) \tag{3.6}$$

and

$$(u, \nabla \cdot \alpha \nabla w) = \oint\limits_{\partial\Omega} u^* \alpha \frac{\partial w}{\partial \hat{\mathbf{n}}} \, \mathrm{d}S - (\nabla u, \alpha \nabla w) \, . \tag{3.7}$$

The relation (3.7) is useful when applying the energy method to, for example, the wave equation, the heat equation, and the Schrödinger equation.

3.2 Example: the heat equation

The heat equation can be written

$$\rho c_p u_t = \nabla \cdot \kappa \nabla u + F, \quad \mathbf{x} \in \Omega, \quad t > 0 \tag{3.8}$$

where u is temperature, $\kappa = \kappa(\mathbf{x})$ is the thermal conductivity, $c_p = c_p(\mathbf{x})$ is specific heat capacity, $\rho = \rho(\mathbf{x})$ is density, and $F = F(\mathbf{x}, t)$ is the volumetric heat source. We assume that all quantities are real. The energy method yields (with $F = 0$)

$$\underbrace{(u, u_t)_{\rho c_p}}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{\rho c_p}} = (u, \nabla \cdot \kappa \nabla u) = [\text{IBP, using (3.7)}] = \oint\limits_{\partial\Omega} u\kappa \frac{\partial u}{\partial \hat{\mathbf{n}}} \,\mathrm{d}S - \underbrace{(\nabla u, \kappa \nabla u)}_{=\|\nabla u\|^2_\kappa}. \tag{3.9}$$

We obtain the energy rate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{\rho c_p} = \oint\limits_{\partial\Omega} u\kappa \frac{\partial u}{\partial \hat{\mathbf{n}}} \,\mathrm{d}S - \|\nabla u\|^2_\kappa \leq \oint\limits_{\partial\Omega} u\kappa \frac{\partial u}{\partial \hat{\mathbf{n}}} \,\mathrm{d}S. \tag{3.10}$$

For well-posedness, we need a boundary condition on the entire boundary $\partial\Omega$. For example, the boundary integral vanishes if we impose homogeneous Dirichlet conditions,

$$u = 0, \quad \mathbf{x} \in \partial\Omega, \tag{3.11}$$

or homogeneous Neumann conditions,

$$\kappa \frac{\partial u}{\partial \hat{\mathbf{n}}} = 0, \quad \mathbf{x} \in \partial\Omega. \tag{3.12}$$

We then obtain

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{\rho c_p} \leq 0, \tag{3.13}$$

and thus well-posedness. Since boundary data do not affect well-posedness, corresponding inhomogeneous conditions are also well-posed. There are many other well-posed BC, including Robin conditions. It is also possible to divide $\partial\Omega$ into parts and impose, for example, Dirichlet conditions on one part and Neumann conditions on the other.

3.2.1 The heat equation on a rectangular domain

As an example of a domain, let $\Omega = [0, W_x] \times [0, W_y] \subset \mathbb{R}^2$. The boundary, $\partial\Omega$, can be divided into four vertical or horizontal lines (with subscripts corresponding to compass directions):

$$\begin{aligned} \Gamma_W &= \{(x, y) \in \partial\Omega : x = 0\} \\ \Gamma_E &= \{(x, y) \in \partial\Omega : x = W_x\} \\ \Gamma_S &= \{(x, y) \in \partial\Omega : y = 0\} \\ \Gamma_N &= \{(x, y) \in \partial\Omega : y = W_y\} \end{aligned} \tag{3.14}$$

For this specific domain, the general energy rate (3.10) can be written

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2_{\rho c_p} = \int\limits_{\Gamma_E} u\kappa u_x \,\mathrm{d}S - \int\limits_{\Gamma_W} u\kappa u_x \,\mathrm{d}S + \int\limits_{\Gamma_N} u\kappa u_y \,\mathrm{d}S - \int\limits_{\Gamma_S} u\kappa u_y \,\mathrm{d}S - \|\nabla u\|^2_{\kappa}. \tag{3.15}$$

4 Fourier analysis of well-posedness for periodic problems

So far, we have used the energy method to analyze well-posedness. The energy method is useful because it provides insight into which boundary conditions are well posed and whether they cause dissipation or growth of energy. The energy method also allows for variable coefficients in many cases. However, to determine if a PDE with constant coefficients is well posed in itself (that is, with periodic boundary conditions), it is often easier to use the so-called Fourier method.

4.1 Fourier series

The Fourier-series representation of a W-periodic function $u = u(x)$ is

$$u(x) = \sum_{k \in K_\infty} \widehat{u}_k e^{ikx}, \tag{4.1}$$

where $\widehat{u}_k$ is the Fourier coefficient corresponding to wavenumber k and K_∞ is the set of wavenumbers corresponding to W-periodic Fourier modes, i.e.,

$$K_\infty = \left\{0, \pm\frac{2\pi}{W}, \pm\frac{4\pi}{W}, \dots\right\}. \tag{4.2}$$

The Fourier coefficients are given by

$$\widehat{u}_k = \frac{1}{W}\int_0^W e^{-ikx} u(x)\, \mathrm{d}x. \tag{4.3}$$

Parselval's relation connects the norm of u and the magnitude of the Fourier coefficients:

$$\|u\|^2 = \sum_{k \in K_\infty} |\widehat{u}_k|^2. \tag{4.4}$$

4.2 Example: the advection equation

Consider the 1D advection equation,

$$u_t = -cu_x, \quad x \in (0, W), \quad t > 0, \tag{4.5}$$

with constant c and *periodic* boundary conditions. For any fixed time, we can decompose the solution into Fourier modes (e^{ikx}) and represent it as a Fourier series according to (4.1). This implies that the solution can be represented as a Fourier series with time-dependent coefficients:

$$u(x,t) = \sum_{k \in K_\infty} \widehat{u}_k(t) e^{ikx}. \tag{4.6}$$

Inserting this representation in the PDE yields

$$\sum_{k\in K_\infty} \frac{\mathrm{d}\widehat{u}_k}{\mathrm{d}t} e^{ikx} = -c \sum_{k\in K_\infty} ik\widehat{u}_k e^{ikx}. \tag{4.7}$$

Since the Fourier modes are *orthogonal*, the equality must hold mode by mode, i.e.,

$$\frac{\mathrm{d}\widehat{u}_k}{\mathrm{d}t} = -cik\widehat{u}_k, \quad \forall k \in K_\infty. \tag{4.8}$$

This is a set of scalar ODE for the Fourier coefficients. The solutions are

$$\widehat{u}_k(t) = \widehat{u}_k(0)e^{-ikct}, \tag{4.9}$$

where $\widehat{u}_k(0)$ are the Fourier coefficients of the initial data. In this case, we can see that the Fourier coefficients do not grow or decay in time, since $|e^{-ikct}| = 1$. The same holds for $\|u\|$ according to Parseval's relation:

$$\|u(\cdot,t)\|^2 = \sum_{k\in K_\infty} |\widehat{u}_k(t)|^2 = \sum_{k\in K_\infty} |\widehat{u}_k(0)|^2 = \|u(\cdot,0)\|^2, \tag{4.10}$$

which matches the energy estimates that we have derived for the advection equation.

This example illustrates that we can analyze well-posedness of periodic problems by studying the growth of the Fourier coefficients.

4.3 General 1D PDEs

Consider a general linear and periodic problem

$$u_t = \mathcal{D}u + F, \quad x \in (0,W), \quad t > 0, \tag{4.11}$$

where $\mathcal{D}$ is a differential operator. For well-posedness it is sufficient to study the homogeneous version, so we set $F = 0$. The Fourier coeffiecients satisfy the ODE

$$\frac{\mathrm{d}\widehat{u}_k}{\mathrm{d}t} = \widehat{\mathcal{D}}(k)\widehat{u}_k, \tag{4.12}$$

where $\widehat{\mathcal{D}}(k)$ is the Fourier transform of $\mathcal{D}$, which can be obtained by replacing every $\frac{\partial}{\partial x}$ by ik, just like in the example with the advection equation. $\widehat{\mathcal{D}}(k)$ is sometimes referred to as the *symbol* of $\mathcal{D}$.

Let c be a real constant and $a > 0$ a real, positive constant. Examples of $\widehat{\mathcal{D}}(k)$ are

$$\begin{aligned}
\mathcal{D} &= -c\frac{\partial}{\partial x} \longrightarrow \quad \widehat{\mathcal{D}}(k) = -cik && \text{(advection equation)}\\
\mathcal{D} &= a\frac{\partial^2}{\partial x^2} \longrightarrow \quad \widehat{\mathcal{D}}(k) = a(ik)^2 = -ak^2 && \text{(heat equation)}\\
\mathcal{D} &= ic\frac{\partial^2}{\partial x^2} \longrightarrow \quad \widehat{\mathcal{D}}(k) = ic(ik)^2 = -ick^2 && \text{(Schrödinger equation).}
\end{aligned} \tag{4.13}$$

The Fourier coefficients satisfy

$$\widehat{u}_k(t) = \widehat{u}_k(0) e^{\widehat{\mathcal{D}}(k)t}, \tag{4.14}$$

and hence the periodic problem is well posed if there are constants K and α such that

$$|e^{\widehat{\mathcal{D}}(k)t}| \leq K e^{\alpha t} \tag{4.15}$$

for all k. See [2], page 19.

For our three example PDEs, we have

$$\begin{array}{llll}
\text{advection equation:} & \mathcal{D} = -c\dfrac{\partial}{\partial x}, & \widehat{\mathcal{D}}(k) = -cik, & |e^{\widehat{\mathcal{D}}(k)t}| = 1 \qquad\qquad \text{(conservation)} \\
\text{heat equation:} & \mathcal{D} = a\dfrac{\partial^2}{\partial x^2}, & \widehat{\mathcal{D}}(k) = -ak^2, & |e^{\widehat{\mathcal{D}}(k)t}| = e^{-ak^2 t} \leq 1 \quad \text{(dissipation)} \\
\text{Schrödinger equation:} & \mathcal{D} = ic\dfrac{\partial^2}{\partial x^2}, & \widehat{\mathcal{D}}(k) = -ick^2, & |e^{\widehat{\mathcal{D}}(k)t}| = 1 \qquad\qquad \text{(conservation)}
\end{array}$$

All three examples are clearly well posed (they satisfy (4.15) with $K = 1$, $\alpha = 0$). We can see that a purely imaginary $\widehat{\mathcal{D}}$ conserves the magnitude of all Fourier coefficients and is therefore characteristic of wave propagation (advection and Schrödinger equations). A negative real part of $\widehat{\mathcal{D}}$ implies decay (the heat equation) while a positive real part indicates growth. In the heat equation, the rate of dissipation depends on the wavenumber, k. Large wavenumbers dissipate the fastest.

An example of an ill-posed problem is the *backwards* heat equation:

$$\begin{array}{lll}
u_t = au_{xx}, & x \in (0, W), & 0 \leq t < T, \\
u = u_T, & x \in [0, W], & t = T,
\end{array} \tag{4.16}$$

where the temperature at some time T is known and we are trying to determine the temperature at previous times. After the change of variables $\tau = T - t$, the problem reads

$$\begin{array}{lll}
u_\tau = -au_{xx}, & x \in (0, W), & 0 \leq \tau < T, \\
u = u_T, & x \in [0, W], & \tau = 0.
\end{array} \tag{4.17}$$

The symbol of the spatial operator is

$$\widehat{\mathcal{D}}(k) = ak^2 \tag{4.18}$$

and therefore

$$|e^{\widehat{\mathcal{D}}(k)\tau}| = e^{ak^2\tau}, \tag{4.19}$$

which cannot be bounded by $Ke^{\alpha\tau}$, since k may be arbitrarily large. This shows that the backwards heat equation is ill posed.

5 Finite difference methods for periodic problems

5.1 Semi-discrete approximation

Consider a general linear PDE in one dimension,

$$u_t = \mathcal{D}u + F, \quad x \in (0, W), \quad t > 0, \tag{5.1}$$

with periodic boundary conditions and initial data $u(x, 0) = u_0(x)$. The operator $\mathcal{D}$ is a linear differential operator in space. To solve (5.1) numerically, we will first discretize in space while leaving time continuous. This approach is known as the *method of lines.* We introduce an equidistant grid of m points, uniformly distributed between $x = 0$ and $x = W$:

$$x_j = jh, \quad j = 0, 1, \ldots m-1, \tag{5.2}$$

where x_j is the jth grid point and h is the grid spacing:

$$h = \frac{W}{m} \tag{5.3}$$

Notice that the last grid point x_{m-1} is positioned at $x = W - h$. This is due to the fact that the solution is assumed to be periodic so that $u(W, t) = u(0, t)$ and the grid point at $x = W$ would be redundant. For non-periodic problems, which we will study later, we will make sure to place a grid point at $x = W$.

Let $\mathbf{v}$ denote the semi-discrete solution vector, where

$$\mathbf{v} = \left[v_0, v_1, \ldots, v_{m-1}\right]^T \tag{5.4}$$

and v_j approximates the solution at grid point j, i.e.,

$$v_j(t) \approx u(x_j, t). \tag{5.5}$$

We will need to be able to compute approximations of spatial derivative operators such as $\frac{\partial}{\partial x}$ and $\frac{\partial^2}{\partial x^2}$.

5.2 Introduction to finite difference approximations

The *forward difference* operator D_+ approximates $\frac{\partial}{\partial x}$ as follows:

$$u_x(x_j) \approx \frac{v_{j+1} - v_j}{h} =: (D_+\mathbf{v})_j \tag{5.6}$$

This approximation is first-order accurate, because the error when applied to a smooth function is $\mathcal{O}(h)$:

$$\frac{u(x_{j+1}) - u(x_j)}{h} = u_x(x_j) + \mathcal{O}(h). \tag{5.7}$$

Similarly, the *backward difference* operator D_-,

$$(D_-\mathbf{v})_j = \frac{v_j - v_{j-1}}{h}, \tag{5.8}$$

is first-order accurate, while the *central difference* operator D_0 is second-order accurate:

$$(D_0\mathbf{v})_j = \frac{v_{j+1} - v_{j-1}}{2h}. \tag{5.9}$$

In matrix form, D_0 is an $m \times m$ matrix:

$$D_0 = \frac{1}{2h}\begin{bmatrix} 0 & 1 & & & & & -1 \\ -1 & 0 & 1 & & & & \\ & -1 & 0 & 1 & & & \\ & & & \ddots & & & \\ & & & & -1 & 0 & 1 \\ 1 & & & & & -1 & 0 \end{bmatrix} \tag{5.10}$$

so that

$$D_0\mathbf{v} = \frac{1}{2h}\begin{bmatrix} v_1 - v_{m-1} \\ v_2 - v_0 \\ v_3 - v_1 \\ \vdots \\ v_{m-1} - v_{m-3} \\ v_0 - v_{m-2} \end{bmatrix} \tag{5.11}$$

Let us solve the advection equation with periodic boundary conditions:

$$\begin{aligned} u_t &= -cu_x + F, \quad && x \in (0, W), \quad t > 0, \\ u &= u_0, && x \in [0, W], \quad t = 0. \end{aligned} \tag{5.12}$$

A semi-discrete approximation using the second order central finite difference operator D_0 reads

$$\begin{aligned} \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} &= -cD_0\mathbf{v} + \mathbf{F}, \quad && t > 0, \\ \mathbf{v} &= \mathbf{v}_0, && t = 0, \end{aligned} \tag{5.13}$$

where $\mathbf{v}_0$ is given by evaluating the initial data u_0 on the grid,

$$\mathbf{v}_0 = \left[u_0(x_0), u_0(x_1), \ldots, u_0(x_{m-1})\right]^T \tag{5.14}$$

and $\mathbf{F}(t)$ is given by evaluating the forcing function $F(x,t)$ on the grid,

$$\mathbf{F}(t) = \left[F(x_0,t), F(x_1,t), \ldots, F(x_{m-1},t)\right]^T. \tag{5.15}$$

Notice that the semi-discrete approximation (5.13) is a system of ODEs, which can be solved by any suitable method, for example an explicit Runge–Kutta method, which will be a frequent choice in this course.

5.3 Proof that D_0 is second order accurate

To determine the order of accuracy of a finite difference operator, we apply it to a smooth function $f(x)$ and study the truncation error. We have

$$(D_0 f)_j = \frac{f(x_{j+1}) - f(x_{j-1})}{2h} = \frac{f(x_j + h) - f(x_j - h)}{2h}. \tag{5.16}$$

Taylor expanding f around $x = x_j$ yields

$$f(x_j + \alpha h) = f(x_j) + \alpha h f'(x_j) + \frac{\alpha^2 h^2}{2!} f''(x_j) + \frac{\alpha^3 h^3}{3!} f'''(x_j) + ... \tag{5.17}$$

Let us introduce the shorthand notation

$$f_j = f(x_j), \quad f'_j = f'(x_j), \ldots \tag{5.18}$$

so that the Taylor expansion reads

$$f(x_j + \alpha h) = f_j + \alpha h f'_j + \frac{\alpha^2 h^2}{2!} f''_j + \frac{\alpha^3 h^3}{3!} f'''_j + \frac{\alpha^4 h^4}{4!} f''''_j + \mathcal{O}(h^5). \tag{5.19}$$

Returning to D_0, with $\alpha = 1$, we obtain

$$f(x_j + h) = f_j + h f'_j + \frac{h^2}{2!} f''_j + \frac{h^3}{3!} f'''_j + \mathcal{O}(h^4) \tag{5.20}$$

and, with $\alpha = -1$, we obtain

$$f(x_j - h) = f_j - h f'_j + \frac{h^2}{2!} f''_j - \frac{h^3}{3!} f'''_j + \mathcal{O}(h^4). \tag{5.21}$$

Hence,

$$\begin{aligned} (D_0 f)_j &= \frac{f(x_j + h) - f(x_j - h)}{2h} \\ &= \frac{1}{2h}\left(2f'_j + 2\frac{h^3}{3!} f'''_j + \mathcal{O}(h^4)\right) = f'_j + \mathcal{O}(h^2), \end{aligned} \tag{5.22}$$

which shows that the truncation error is $\mathcal{O}(h^2)$, and hence D_0 is second order accurate. In general, a finite difference operator is pth order accurate if the truncation error is $\mathcal{O}(h^p)$.

If the error from the time discretization is negligible, we expect the numerical error compared to the true solution to be proportional to h^p.

5.4 High-order finite difference approximations of $\frac{\partial}{\partial x}$

Consider a five-point central finite difference operator $D_{0,4}$. The subscript 0 indicates a central stencil and the subscript 4 indicates 4th order accuracy, which we hope to obtain. We make the ansatz:

$$(D_{0,4} f)_j = \frac{1}{h} \sum_{k=-2}^{2} \gamma_k f(x_j + kh). \tag{5.23}$$

Let us try to find the coefficients γ_k that maximize the order of accuracy of $D_{0,4}$.

$$\begin{aligned}
f(x_j-2h) =& f_j - 2hf_j' + \frac{2^2h^2}{2!}f_j'' - \frac{2^3h^3}{3!}f_j''' + \frac{2^4h^4}{4!}f_j'''' - \frac{2^5h^5}{5!}f_j''''' + \mathcal{O}(h^6)\\
f(x_j-h) =& f_j - hf_j' + \frac{h^2}{2!}f_j'' - \frac{h^3}{3!}f_j''' + \frac{h^4}{4!}f_j'''' - \frac{h^5}{5!}f_j''''' + \mathcal{O}(h^6)\\
f(x_j) =& f_j \\
f(x_j+h) =& f_j + hf_j' + \frac{h^2}{2!}f_j'' + \frac{h^3}{3!}f_j''' + \frac{h^4}{4!}f_j'''' + \frac{h^5}{5!}f_j''''' + \mathcal{O}(h^6)\\
f(x_j+2h) =& f_j + 2hf_j' + \frac{2^2h^2}{2!}f_j'' + \frac{2^3h^3}{3!}f_j''' + \frac{2^4h^4}{4!}f_j'''' + \frac{2^5h^5}{5!}f_j''''' + \mathcal{O}(h^6)
\end{aligned} \tag{5.24}$$

We obtain

$$\begin{aligned}
(D_{0,4}f)_j = & \frac{1}{h}f_j\left(\gamma_{-2}+\gamma_{-1}+\gamma_0+\gamma_1+\gamma_2\right)\\
&+f_j'\left(-2\gamma_{-2}-\gamma_{-1}+\gamma_1+2\gamma_2\right)\\
&+\frac{h}{2!}f_j''\left(2^2\gamma_{-2}+\gamma_{-1}+\gamma_1+2^2\gamma_2\right)\\
&+\frac{h^2}{3!}f_j'''\left(-2^3\gamma_{-2}-\gamma_{-1}+\gamma_1+2^3\gamma_2\right)\\
&+\frac{h^3}{4!}f_j''''\left(2^4\gamma_{-2}+\gamma_{-1}+\gamma_1+2^4\gamma_2\right)\\
&+\mathcal{O}(h^4)
\end{aligned} \tag{5.25}$$

Requiring 4th order of accuracy leads to a 5×5 system of equations,

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1\\ -2 & -1 & 0 & 1 & 2\\ 2^2 & 1 & 0 & 1 & 2^2\\ -2^3 & -1 & 0 & 1 & 2^3\\ 2^4 & 1 & 0 & 1 & 2^4 \end{bmatrix} \begin{bmatrix}\gamma_{-2}\\ \gamma_{-1}\\ \gamma_0\\ \gamma_1\\ \gamma_2\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\\0\end{bmatrix} \tag{5.26}$$

which has the unique solution

$$\begin{bmatrix}\gamma_{-2}\\ \gamma_{-1}\\ \gamma_0\\ \gamma_1\\ \gamma_2\end{bmatrix} = \frac{1}{12}\begin{bmatrix}1\\-8\\0\\8\\-1\end{bmatrix}. \tag{5.27}$$

Central finite difference stencils of arbitrarily high orders exist. In general, there is a unique $(2n+1)$-point stencil of order $2n$. These stencils all have the properties

$$\gamma_0 = 0, \quad \gamma_{-k} = -\gamma_k. \tag{5.28}$$

Note that the above is only true for stencils approximating $\frac{\partial}{\partial x}$.

5.5 Fourier representations

The Fourier series representation of a W-periodic function $u = u(x)$ is

$$u(x) = \sum_{k \in K_\infty} \widehat{u}_k e^{ikx}, \tag{5.29}$$

where $\widehat{u}_k$ is the Fourier coefficient corresponding to wavenumber k and K_∞ is the set of wavenumbers corresponding to W-periodic Fourier modes, i.e.,

$$K_\infty = \left\{0, \pm\frac{2\pi}{W}, \pm\frac{4\pi}{W}, \ldots\right\}. \tag{5.30}$$

Now consider a grid vector

$$\mathbf{v} = \left[v_0, v_1, \ldots, v_{m-1}\right]^T. \tag{5.31}$$

For simplicity, we assume that m is odd so that $m = 2N + 1$, with N an integer. Then $\mathbf{v}$ can be represented by its Fourier expansion

$$v_j = \sum_{k \in K_N} \widehat{v}_k e^{ikx_j}, \tag{5.32}$$

where K_N is the set of wavenumbers corresponding to the first $2N + 1$ Fourier modes,

$$K_N = \left\{0, \pm\frac{2\pi}{W}, \ldots \pm N\frac{2\pi}{W}\right\}. \tag{5.33}$$

5.6 Dispersion errors

Let us differentiate a pure Fourier mode:

$$\frac{\mathrm{d}}{\mathrm{d}x} e^{ikx} = ike^{ikx}. \tag{5.34}$$

Now consider the action of D_0 on a pure Fourier mode:

$$\begin{aligned}(D_0 e^{ikx})_j &= \frac{1}{2h}\left(e^{ikx_{j+1}} - e^{ikx_{j-1}}\right) = \frac{1}{2h}\left(e^{ikh} - e^{-ikh}\right)e^{ikx_j} = \frac{1}{2h} 2i\sin(kh) e^{ikx_j} \\ &= ik\frac{\sin(kh)}{kh} e^{ikx_j} = \widehat{D_0}(k) e^{ikx_j},\end{aligned} \tag{5.35}$$

where we have defined

$$\widehat{D_0}(k) = ik\frac{\sin(kh)}{kh}. \tag{5.36}$$

$\widehat{D_0}(k)$ is called the Fourier transform (or symbol) of D_0. For higher orders of accuracy, we have similar expressions:

$$\text{4th order: } \widehat{D_{0,4}}(k) = ik\frac{\sin(kh)}{kh}\left(1 + \frac{2}{3}\sin^2\frac{kh}{2}\right). \tag{5.37}$$

$$\text{6th order: } \widehat{D_{0,6}}(k) = ik\frac{\sin(kh)}{kh}\left(1 + \frac{2}{3}\sin^2\frac{kh}{2} + \frac{8}{15}\sin^4\frac{kh}{2}\right). \tag{5.38}$$

5.6.1 The advection equation

Consider the advection equation with constant wave speed,

$$u_t = -cu_x. \tag{5.39}$$

To derive the dispersion relation for this equation, we look for plane wave solutions. That is, we make the ansatz

$$u(x,t) = e^{i(kx-\omega t)}, \tag{5.40}$$

and then seek a relation $\omega = \omega(k)$ such that u satisfies the PDE. Inserting the ansatz yields

$$-i\omega e^{i(kx-\omega t)} = -cike^{i(kx-\omega t)}, \tag{5.41}$$

which can only be satisfied for all x and t if

$$-i\omega = -cik, \tag{5.42}$$

which implies

$$\omega = ck. \tag{5.43}$$

Equation (5.43) is the exact dispersion relation for the advection equation. The phase velocity v_p is defined as ω/k, which in this case yields

$$v_p = \frac{ck}{k} = c. \tag{5.44}$$

This matches what we know about the advection equation: all wavenumbers travel with velocity c.

Now let us study the semi-discrete dispersion relation. After using some finite-difference approximation D to discretize $\frac{\partial}{\partial x}$, the semi-discrete equation reads

$$\frac{\mathrm{d}v_j}{\mathrm{d}t} = -c(D\mathbf{v})_j. \tag{5.45}$$

We make the ansatz

$$v_j(t) = e^{i(kx_j-\omega t)}. \tag{5.46}$$

Inserting this ansatz, using (5.35), yields

$$-i\omega e^{i(kx_j-\omega t)} = -c\widehat{D}(k)e^{i(kx_j-\omega t)}. \tag{5.47}$$

The semi-discrete dispersion relation is

$$\omega = c\frac{\widehat{D}(k)}{i} \tag{5.48}$$

and the semi-discrete phase velocity is

$$v_p^{num} = \frac{\omega}{k} = c\frac{\widehat{D}(k)}{ik}. \tag{5.49}$$

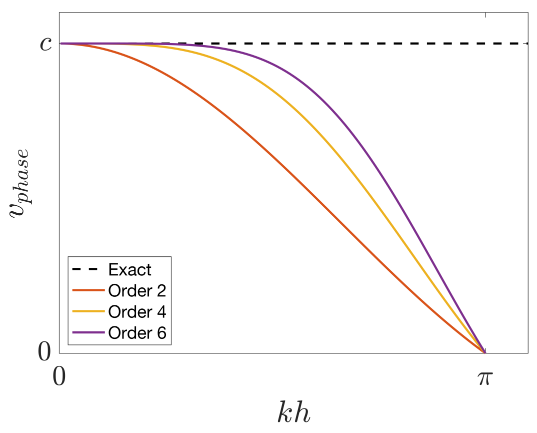


Figure 5.1: Exact and approximate phase velocities as functions of grid resolution kh, for the advection equation and central difference operators of orders 2, 4, and 6.

For the 2nd order accurate D_0, we get

$$v_p^{num} = c\frac{\widehat{D_0}(k)}{ik} = c\frac{\sin(kh)}{kh}. \tag{5.50}$$

The quantity kh is a measure of how well resolved the wavenumber k is on the grid. The smaller kh is, the better resolved the mode is. $kh = \pi$ corresponds to the highest mode that can be represented on the grid, the Nyquist mode, with two grid points per wavelength. To see this, note that the relation between wavenumber (k) and wavelength (λ) is

$$\lambda = \frac{2\pi}{k}. \tag{5.51}$$

The number of grid points per wavelength, N_{ppwl}, can be computed as

$$N_{ppwl} = \frac{\lambda}{h} = \frac{2\pi}{kh} \tag{5.52}$$

which yields

$$kh = \frac{2\pi}{N_{ppwl}}. \tag{5.53}$$

Numerical phase velocities for orders 2, 4, and 6 are plotted in Fig. 5.1.

6 Summation-by-parts finite difference methods

For non-periodic problems, centered finite difference stencils only work in the interior of the domain, far from any boundaries. Near boundaries, one has to transition to biased stencils, which use more points on one side than the other. It is very difficult to do this in a way that is both *accurate* and *stable*. The solution is given by the *summation-by-parts* framework, which is introduced in Section 6.2. We first introduce the concept of *stability*, which is a discrete version of well-posedness.

6.1 Stability

Consider a well posed, linear IBVP,

$$\begin{aligned} u_t &= \mathcal{D}u + F, & x &\in (0,W), & t &> 0,\\ \mathcal{L}_\ell u &= g_\ell, & x &= 0, & t &> 0,\\ \mathcal{L}_r u &= g_r, & x &= W, & t &> 0,\\ u &= u_0, & x &\in [0,W], & t &= 0, \end{aligned} \tag{6.1}$$

We introduce an equidistant grid of $m+1$ points, uniformly distributed between $x=0$ and $x=W$:

$$x_j = jh, \quad j = 0,1,\dots m, \tag{6.2}$$

where x_j is the jth grid point and h is the grid spacing:

$$h = \frac{W}{m}. \tag{6.3}$$

Let $\mathbf{v}$ denote the semi-discrete solution vector, where

$$\mathbf{v} = \left[v_0, v_1, \dots, v_m\right]^T \tag{6.4}$$

and v_j approximates the solution at grid point j, i.e.,

$$v_j(t) \approx u(x_j,t). \tag{6.5}$$

A semi-discrete approximation of (6.1) can be written

$$\begin{aligned} \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} &= D\mathbf{v} + \mathbf{F}, & t &> 0,\\ L_\ell \mathbf{v} &= g_\ell, & t &> 0,\\ L_r \mathbf{v} &= g_r, & t &> 0,\\ \mathbf{v} &= \mathbf{v}_0, & t &= 0, \end{aligned} \tag{6.6}$$

where $L_{\ell,r}$ and D denote discrete operators. Henceforth, we will use $\mathbf{v}_t$ to mean $\frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t}$.

In (6.6), we have discretized the boundary conditions, but not specified *how* we are going to impose that the discrete solution $\mathbf{v}$ satisfies them. There are essentially two different ways to impose boundary conditions: i) *weakly*, and ii) *strongly*. With strong enforcement, the discrete solution satisfies the boundary condition *exactly*. With weak enforcement, the discrete solution only satisfies the boundary condition *approximately*.

A semi-discrete approximation of (6.1), with *weakly enforced boundary conditions* can for example be written

$$\begin{aligned} \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} = D\mathbf{v} + \mathbf{F} + A_\ell(L_\ell\mathbf{v} - g_\ell) + A_r(L_r\mathbf{v} - g_r), \quad t > 0, \\ \mathbf{v} = \mathbf{v}_0, \quad t = 0, \end{aligned} \tag{6.7}$$

with some suitable (yet unspecified) matrices $A_{\ell,r}$. (A corresponding semi-discrete approximation of (6.1) with *strongly enforced boundary conditions* is for example given by (6.31).)

We are now ready to define *stability* for semi-discrete approximations:

Definition 4 (Stability)**.** The semi-discrete approximations (6.7), or (6.31) are *stable* if, for $F = 0$, $g_\ell = 0$, $g_r = 0$, the semi-discrete solution satisfies

$$\|\mathbf{v}\|_H \leq Ke^{\alpha t}\|\mathbf{v}_0\|_H \tag{6.8}$$

in some discrete norm $\|\cdot\|_H$, for some constants K and α.

Notice the similarity between the definitions of stability and well-posedness. Stability is essential—if you accidentally create an *unstable* discretization, you will see the numerical solution "blow up". More formally, the importance of stability is given by the following classical theorem:

Theorem 1 (Lax equivalence theorem)**.** *Assume that the PDE problem is well posed and that the difference approximation is consistent. Then the approximation is convergent if and only if it is stable.*

Convergence means that $\mathbf{v}$ approaches the true solution u as the grid is refined (i.e., as $h \to 0$). This is what we are really after! To get convergence, the Lax equivalence theorem (also known as the Lax-Richtmyer theorem) states the we need *consistency* and stability. Consistency means that all local errors tend to zero as $h \to 0$. This is relatively easy to ensure, by only using finite difference approximations that are at least first-order accurate. In practice, the difficult part is therefore to ensure stability. In the semi-discrete approximation (6.7), we would need to select the matrices $A_{\ell,r}$ in a way that yields stability. Before we can do that, we need the summation-by-parts framework.

6.2 Summation-by-parts finite difference operators

Summation-by-parts (SBP) finite difference operators were developed in Uppsala in the 1970's by Kreiss and Scherer [3, 8] as a way to construct stable finite difference methods for non-periodic problems. SBP operators are typically based on central difference operators. Near boundaries, they transition to carefully chosen non-central stencils, which are designed so that the finite difference operator satisfies a property that mimics *integration* *by parts.* Recall that integration by parts is a key ingredient in the energy method, which allows us to prove well-posedness in the continuous setting. The SBP property allows us to use a discrete energy method to prove semi-discrete stability.

Let $\mathbf{e}_\ell$ and $\mathbf{e}_r$ denote the following vectors in $\mathbb{R}^{m+1}$:

$$\mathbf{e}_\ell = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \quad \mathbf{e}_r = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Definition 5. A difference operator D_1 approximating $\partial/\partial x$ is a first-derivative SBP operator with quadrature matrix H if $H = H^T > 0$ and

$$HD_1 = \mathbf{e}_r \mathbf{e}_r^T - \mathbf{e}_\ell \mathbf{e}_\ell^T - D_1^T H.$$

Definition 6. A difference operator D_2 approximating $\partial^2/\partial x^2$ is a second-derivative SBP operator with quadrature matrix H if $H = H^T > 0$ and

$$HD_2 = \mathbf{e}_r \mathbf{d}_r^T - \mathbf{e}_\ell \mathbf{d}_\ell^T - M,$$

where $M = M^T \geq 0$, and $\mathbf{d}_\ell^T \mathbf{v} \simeq u_x|_{x=0}$, $\mathbf{d}_r^T \mathbf{v} \simeq u_x|_{x=W}$ are finite difference approximations of the first derivatives at the left and right boundary points.

Definition 7. Let $(\cdot,\cdot)_H$ denote the discrete inner product, defined by

$$(\mathbf{u}, \mathbf{v})_H = \mathbf{u}^* H \mathbf{v}.$$

Note that $(\cdot,\cdot)_H$ approximates the L^2 inner product $(\cdot,\cdot)$, since H is a quadrature operator. In the discrete inner product, the SBP operators satisfy

$$(\mathbf{u}, D_1\mathbf{v})_H = (\mathbf{e}_r^T \mathbf{u})^*(\mathbf{e}_r^T \mathbf{v}) - (\mathbf{e}_\ell^T \mathbf{u})^*(\mathbf{e}_\ell^T \mathbf{v}) - (D_1 \mathbf{u}, \mathbf{v})_H$$

and

$$(\mathbf{u}, D_2\mathbf{v})_H = (\mathbf{e}_r^T \mathbf{u})^*(\mathbf{d}_r^T \mathbf{v}) - (\mathbf{e}_\ell^T \mathbf{u})^*(\mathbf{d}_\ell^T \mathbf{v}) - \mathbf{u}^* M \mathbf{v}.$$

Note the similarities with the corresponding integration-by-parts formulas.

The simplest example of an SBP operator is the second-order accurate first-derivative case:

$$D_1 = \frac{1}{h} \begin{bmatrix} -1 & 1 & & & & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & & & & \\ & -\frac{1}{2} & 0 & \frac{1}{2} & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & & -\frac{1}{2} & 0 & \frac{1}{2} \\ & & & & & -1 & 1 \end{bmatrix} \tag{6.9}$$

$$D_1 = \frac{1}{h}\begin{bmatrix} -\frac{24}{17} & \frac{59}{34} & -\frac{4}{17} & -\frac{3}{34} & 0 & 0 & 0 & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 & & \\ \frac{4}{43} & -\frac{59}{86} & 0 & \frac{59}{86} & -\frac{4}{43} & 0 & 0 & & \\ \frac{3}{98} & 0 & -\frac{59}{98} & 0 & \frac{32}{49} & -\frac{4}{49} & 0 & & \\ 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & & \\ 0 & 0 & 0 & \frac{1}{12} & -\frac{2}{3} & 0 & \frac{2}{3} & -\frac{1}{12} & \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix}$$

2nd order stencils

4th order stencil

Figure 6.1: The upper left corner of a 4th order SBP operator

with quadrature operator

$$H = h\begin{bmatrix} \frac{1}{2} & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & \frac{1}{2} \end{bmatrix}. \tag{6.10}$$

In this case, the quadrature operator is actually the composite trapezoidal rule. In general, the order of accuracy of the quadrature operator typically matches the order of the SBP operator.

Figure 6.1 shows an example of a 4th order first-derivative SBP operator. The corresponding quadrature matrix is

$$H = h\begin{bmatrix} \frac{17}{48} & & & & & & \\ & \frac{59}{48} & & & & & \\ & & \frac{43}{48} & & & & \\ & & & \frac{49}{48} & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & \ddots \end{bmatrix}, \tag{6.11}$$

which is a fourth-order accurate quadrature formula. The bottom-right corners of D_1 and H are rotated versions of the upper-left corners.

For SBP operators with diagonal H (which are the only kind we will study in this course) the boundary accuracy is always half that of the interior. For example, a 6th order

SBP operator, which uses standard central 6th order differences in the interior, is only 3rd order accurate near the boundaries, and so on. There are proofs that it is impossible to have higher boundary accuracy while maintaining the SBP property.

As we will see, SBP operators facilitate semi-discrete stability, but only when combined with an appropriate method for enforcing boundary conditions. There are essentially two alternatives: i) the *Simultaneous Approximation Term* (SAT) method first presented in [1], and ii) the projection method developed in Uppsala in 1995 by Olsson [6, 7]. A comparison between these two methods can be found in [5].

6.3 The SBP-SAT method

The SAT method imposes BC weakly, using so-called *penalty terms*. We will demonstrate it by examples.

6.3.1 SBP-SAT for the advection equation

As our first example of SBP-SAT, we take the advection equation with constant wave speed $c > 0$ and an inhomogeneous boundary condition:

$$\begin{aligned} u_t + cu_x &= 0, \quad && x \in (0, W), \quad && t > 0, \\ u &= u_0, && x \in [0, W], && t = 0, \\ u &= g, && x = 0, && t > 0. \end{aligned} \tag{6.12}$$

For simplicity we assume that the solution is real-valued. Recall for comparison that the energy rate for the continuous problem is

$$\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = -cu^2\Big|_0^W = cg(t)^2 - cu(W,t)^2. \tag{6.13}$$

The semi-discrete approximation using the SBP-SAT method reads

$$\mathbf{v}_t = -cD_1\mathbf{v} + \underbrace{\tau H^{-1}\mathbf{e}_\ell(v_0 - g)}_{SAT}, \tag{6.14}$$

where the last term in the right-hand side is the SAT that enforces the boundary condition. The scalar parameter τ is at this point unknown—we will choose it such that the semi-discrete problem is stable. The SAT method enforces the boundary condition *weakly* (as opposed to strongly), which means that the boundary condition will in general not be satisfied exactly, only approximately, to some order of accuracy. The SAT is a so-called penalty term, which "penalizes" the system by the deviation from the boundary condition. If the difference $v_0 - g$ becomes large, the SAT will become large and force v_0 towards g. At least, that is the idea.

To prove stability, we proceed by applying the discrete energy method, which starts by setting $g = 0$ and multiplying the semi-discrete problem by $\mathbf{v}^*H$ (or here $\mathbf{v}^TH$, since $\mathbf{v}$ is real) from the left, which yields

$$(\mathbf{v}, \mathbf{v}_t)_H = -c\,(\mathbf{v}, D_1\mathbf{v})_H + \mathbf{v}^T H\tau H^{-1}\mathbf{e}_\ell(v_0 - 0). \tag{6.15}$$

By the SBP property,

$$- c\,(\mathbf{v}, D_1\mathbf{v})_H = -c(v_m^2 - v_0^2) + c\,(D_1\mathbf{v}, \mathbf{v})_H\,. \tag{6.16}$$

We obtain

$$(\mathbf{v}, \mathbf{v}_t)_H = -c(v_m^2 - v_0^2) + c\,(D_1\mathbf{v}, \mathbf{v})_H + \tau v_0^2. \tag{6.17}$$

Adding (6.17) and (the conjugate of) (6.15) leads to the discrete energy rate

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{v}\|_H^2 = -c(v_m^2 - v_0^2) + 2\tau v_0^2 = -cv_m^2 + (c + 2\tau)v_0^2 \le (c+2\tau)v_0^2. \tag{6.18}$$

We obtain a stable semi-discretization by setting $\tau \le -\frac{c}{2}$, which leads to the energy estimate

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{v}\|_H^2 \le 0. \tag{6.19}$$

We clearly satisfy the requirement for stability in Definition 4.

Remark. It is not obvious what the optimal value for τ is in this case, only that $\tau \le -\frac{c}{2}$ is necessary. Some research on the topic indicates that $\tau = -c$ is optimal in terms of accuracy. Setting $|\tau| >> c$ would increase the spectral radius of the spatial discretization matrix and thus affect the time step restriction for explicit time-stepping methods negatively.

Remark. To understand why the SBP-SAT method works, recall the Lax equivalence theorem (Theorem 1). To guarantee convergence, we need *stability* and *consistency*. We proved stability (with suitable choice of τ), so it remains to verify consistency. Assume that $u(x,t)$ is the true solution to (6.12), and let $\mathbf{u}(t)$ be $u(x,t)$ evaluated on the grid. Inserting $\mathbf{u}$ in the semidiscrete approximation (6.14) and moving all terms to the same side yields the residual

$$\mathbf{r} = \mathbf{u}_t + \underbrace{cD_1\mathbf{u}}_{=-\mathbf{u}_t + ce_{D_1}} - \tau H^{-1}\mathbf{e}_\ell \underbrace{(u_0 - g)}_{=0,\text{ due to BC.}} = ce_{D_1} \approx 0, \tag{6.20}$$

where e_{D_1} denotes the error coming from the D_1 operator. As long as every row in D_1 is at least first-order accurate, D_1 is consistent and $e_{D_1} \to 0$ as $h \to 0$. The SAT is consistent by construction! In this case, the residual from the SAT is exactly zero (this is not always the case, however).

6.3.2 SBP-SAT for the heat equation with Neumann BC

Consider the heat equation with constant coefficients ($a > 0 =$ constant) and Neumann BC:

$$\begin{aligned} u_t &= au_{xx}, & x &\in (0,W), & t &> 0,\\ au_x &= g_\ell, & x &= 0, & t &> 0,\\ au_x &= g_r, & x &= W, & t &> 0,\\ u &= u_0, & x &\in [0,W], & t &= 0, \end{aligned} \tag{6.21}$$

Recall that the solution satisfies the energy rate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = uau_x|_0^W - \|u_x\|_a^2. \tag{6.22}$$

With homogeneous Neumann (or Dirichlet) BC, the boundary terms vanish and we obtain the energy rate

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = 0 - \|u_x\|_a^2 \leq 0. \tag{6.23}$$

An SBP-SAT discretization of the IBVP reads:

$$\mathbf{v}_t = aD_2\mathbf{v} + \tau_\ell H^{-1}\mathbf{e}_\ell(a\mathbf{d}_\ell^T\mathbf{v} - g_\ell) + \tau_r H^{-1}\mathbf{e}_r(a\mathbf{d}_r^T\mathbf{v} - g_r), \tag{6.24}$$

where $\tau_{\ell,r}$ are scalar parameters that we need to determine for stability. Notice that we used the boundary derivative operators $\mathbf{d}_{\ell,r}$ in the SATs.

To analyze stability, we apply the discrete energy method: set $g_\ell = g_r = 0$ and multiply by $\mathbf{v}^T H$ from the left (we assume that the solution is real),

$$\begin{aligned}
(\mathbf{v}, \mathbf{v}_t)_H &= a\,(\mathbf{v}, D_2\mathbf{v})_H + \tau_\ell(\mathbf{e}_\ell^T\mathbf{v})a(\mathbf{d}_\ell^T\mathbf{v}) + \tau_r(\mathbf{e}_r^T\mathbf{v})a(\mathbf{d}_r^T\mathbf{v}) \\
&= a(\mathbf{e}_r^T\mathbf{v})(\mathbf{d}_r^T\mathbf{v}) - a(\mathbf{e}_\ell^T\mathbf{v})(\mathbf{d}_\ell^T\mathbf{v}) - \mathbf{v}^T aM\mathbf{v} + \tau_\ell(\mathbf{e}_\ell^T\mathbf{v})a(\mathbf{d}_\ell^T\mathbf{v}) + \tau_r(\mathbf{e}_r^T\mathbf{v})a(\mathbf{d}_r^T\mathbf{v}) \\
&= -\mathbf{v}^T aM\mathbf{v} + a(\mathbf{e}_r^T\mathbf{v})(\mathbf{d}_r^T\mathbf{v})(1+\tau_r) + a(\mathbf{e}_\ell^T\mathbf{v})(\mathbf{d}_\ell^T\mathbf{v})(-1+\tau_\ell),
\end{aligned}$$

where we used the SBP property of D_2. With the penalty parameters

$$\tau_\ell = 1, \quad \tau_r = -1, \tag{6.25}$$

we obtain the energy rate

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{1}{2}\|\mathbf{v}\|_H^2 = -\mathbf{v}^T aM\mathbf{v} \leq 0, \tag{6.26}$$

which shows that the semi-discrete problem is stable.

Remark. Notice that in this case, there is no freedom in the choice of $\tau_{\ell,r}$.

6.4 The SBP-Projection method

The projection method imposes BC *strongly*, which means that the discrete solution satisfies the (discretized) BC *exactly*. Consider the general semidiscrete problem (6.6). Let

$$L = \begin{bmatrix} L_\ell \\ L_r \end{bmatrix},$$

where L is the complete boundary operator (that defines all boundary conditions). For simplicity we assume homogeneous BC (i.e. $g_{\ell,r} = 0$). Hence, the boundary conditions can be written $L\mathbf{v} = 0$. Next, we define the *projection operator* P as

$$P = I - H^{-1}L^T\left(LH^{-1}L^T\right)^{-1}L, \tag{6.27}$$

where I is the identity matrix and H the SBP quadrature matrix. By construction, P has the following properties (assuming that L has full rank):

$$P^2 = P \quad (P \text{ is idempotent; projecting twice is the same as projecting once}), \tag{6.28}$$

$$HP = P^T H \quad (HP \text{ is symmetric}), \tag{6.29}$$

$$LP\mathbf{v} = 0 \quad (\text{the projected solution vector satisfies the BC}). \tag{6.30}$$

The projection operator P is designed to be an orthogonal projection onto the subset of vectors that satisfy the BC exactly.

A semi-discrete approximation of (6.1) with the projection method reads

$$\begin{aligned} \frac{\mathrm{d}\mathbf{v}}{\mathrm{d}t} &= PDP\mathbf{v} + P\mathbf{F}, \quad t > 0, \\ \mathbf{v} &= \mathbf{v}_0, \quad t = 0, \end{aligned} \tag{6.31}$$

Notice that the forcing function $\mathbf{F}$ is projected and that we project both before and after applying the operator D. This double projection ensures that D acts on a vector that satisfies the BC and that the result satisfies the BC.

Remark. It can be shown that (6.31) is stable (see Definition 4) by construction, assuming that (6.1) is a well-posed IBVP. We will use examples to demonstrate this.

Remark. The projection method is perfectly capable of handling inhomogeneous BC, but the notation becomes more involved. For simplicity we therefore restrict the analysis to homogeneous BC in this compendium.

6.4.1 SBP-Projection for the advection equation

Once P is constructed, the semi-discrete approximation of the advection equation (6.12) with homogeneous BC using the SBP-Projection method reads

$$\mathbf{v}_t = -cPD_1P\mathbf{v}. \tag{6.32}$$

To construct P, notice that the BC can be discretized as

$$\mathbf{e}_\ell^T \mathbf{v} = 0, \tag{6.33}$$

which means that the boundary operator is $L = \mathbf{e}_\ell^T$. We can now compute P according to the definition (6.27), which here yields

$$P = \begin{bmatrix} 0 & & & & \\ & 1 & & & \\ & & 1 & & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}, \tag{6.34}$$

The projected solution vector here becomes $P\mathbf{v} = [0, v_1, v_2, \ldots, v_m]^T$. Notice that the projection operator enforces the BC exactly, by simply setting the first element to zero.

In a code, one would typically not form the operator P explicitly, but rather implement the action of P. This technique, where the boundary element of the discrete solution is simply replaced by the boundary data (in this case 0) is also known as *injection*.

To prove stability, we proceed by applying the discrete energy method, which starts by multiplying the semi-discrete problem by $\mathbf{v}^T H$ (assuming that $\mathbf{v}$ is real) from the left, which yields

$$(\mathbf{v}, \mathbf{v}_t)_H = -c\,(\mathbf{v}, PD_1P\mathbf{v})_H = -c\,(P\mathbf{v}, D_1P\mathbf{v})_H \ . \tag{6.35}$$

In the last step we used (6.29) i.e. that $HP = P^T H$. By the SBP property,

$$-\,c\,(P\mathbf{v}, D_1P\mathbf{v})_H = -c(P\mathbf{v})_m^2 + c(P\mathbf{v})_0^2 + c\,(D_1P\mathbf{v}, P\mathbf{v})_H \,. \tag{6.36}$$

We obtain

$$(\mathbf{v}, \mathbf{v}_t)_H = -c(P\mathbf{v})_m^2 + (P\mathbf{v})_0^2 + c\,(D_1P\mathbf{v}, P\mathbf{v})_H \,. \tag{6.37}$$

Adding (6.37) and the conjugate of (6.35) leads to the discrete energy rate

$$\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{v}\|_H^2 = -c(P\mathbf{v})_m^2 + (P\mathbf{v})_0^2 = -cP\mathbf{v}_m^2 \leq 0 \ . \tag{6.38}$$

In the last step we used that $P\mathbf{v}$ satisfies the BC exactly, which means that $(P\mathbf{v})_0 = 0$. The discrete energy rate mimics the continuous energy rate (6.13) (with $g = 0$) perfectly. We clearly satisfy the requirement for stability in Definition 4.

Remark. Since the projection method satisfies BC *exactly*, the SBP-Projection method generally mimics the continuous energy rate exactly. This is not always the case with the SBP-SAT method.

6.4.2 SBP-Projection for the heat equation with Dirichlet BC

Consider the heat equation with constant coefficients ($a > 0 =$ constant) and Dirichlet BC:

$$\begin{aligned} u_t = au_{xx}, &\quad x \in (0, W), &\quad t > 0, \\ u = 0, &\quad x = 0, &\quad t > 0, \\ u = 0, &\quad x = W, &\quad t > 0, \\ u = u_0, &\quad x \in [0, W], &\quad t = 0, \end{aligned} \tag{6.39}$$

The semi-discrete approximation using the SBP-Projection method reads:

$$\mathbf{v}_t = aPD_2P\mathbf{v}. \tag{6.40}$$

Here, the boundary operator L is given by

$$L = \begin{bmatrix} \mathbf{e}_\ell^T \\ \mathbf{e}_r^T \end{bmatrix}, \tag{6.41}$$

and, as always, P is built from L according to the definiton (6.27). To prove stability, we apply the discrete energy method: multiply by $\mathbf{v}^T H$ from the left,

$$\begin{aligned}
\underbrace{(\mathbf{v}, \mathbf{v}_t)_H}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{v}\|_H^2} &= a\,(\mathbf{v}, PD_2P\mathbf{v})_H = a\,(P\mathbf{v}, D_2P\mathbf{v})_H \\
&= a\,\underbrace{(\mathbf{e}_r^T P\mathbf{v})}_{=0}(\mathbf{d}_r^T P\mathbf{v}) - a\,\underbrace{(\mathbf{e}_\ell^T P\mathbf{v})}_{=0}(\mathbf{d}_\ell^T P\mathbf{v}) - a(P\mathbf{v})^T MP\mathbf{v} \\
&= -a(P\mathbf{v})^T MP\mathbf{v} \leq 0,
\end{aligned} \tag{6.42}$$

where we used (6.29) in the first step, the SBP property of D_2 in the second step, and thirdly that $LP\mathbf{v} = 0$, i.e., $\mathbf{e}_\ell^T P\mathbf{v} = 0$ and $\mathbf{e}_r^T P\mathbf{v} = 0$. Finally, we used that M is positive semidefinite. This shows that the semi-discrete problem is stable.

6.4.3 SBP-Projection for the heat equation with Robin BC

Consider again the heat equation, but this time with Robin BC:

$$\begin{aligned}
au_x &= bu, \quad x = 0, \quad t > 0, \\
au_x &= -bu, \quad x = W, \quad t > 0,
\end{aligned} \tag{6.43}$$

where $b > 0$ is a constant. Notice that with Robin BC, the energy rate (6.22) becomes:

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|^2 = uau_x|_0^W - \|u_x\|_a^2 = -bu(W,t)^2 - bu(0,t)^2 - \|u_x\|_a^2 \leq 0. \tag{6.44}$$

As before, the semi-discrete approximation using the SBP-Projection method reads:

$$\mathbf{v}_t = aPD_2P\mathbf{v}. \tag{6.45}$$

The only change is in the boundary operator L, which here is

$$L = \begin{bmatrix} a\mathbf{d}_\ell^T - b\mathbf{e}_\ell^T \\ a\mathbf{d}_r^T + b\mathbf{e}_r^T \end{bmatrix}, \tag{6.46}$$

and, as always, P is built from L according to the definiton (6.27). To prove stability, we apply the discrete energy method: multiply by $\mathbf{v}^T H$ from the left,

$$\begin{aligned}
\underbrace{(\mathbf{v}, \mathbf{v}_t)_H}_{=\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|\mathbf{v}\|_H^2} &= a\,(\mathbf{v}, PD_2P\mathbf{v})_H = a\,(P\mathbf{v}, D_2P\mathbf{v})_H \\
&= (\mathbf{e}_r^T P\mathbf{v})\,\underbrace{a(\mathbf{d}_r^T P\mathbf{v})}_{=-b\mathbf{e}_r^T P\mathbf{v}} - (\mathbf{e}_\ell^T P\mathbf{v})\,\underbrace{a(\mathbf{d}_\ell^T P\mathbf{v})}_{=b\mathbf{e}_\ell^T P\mathbf{v}} - a(P\mathbf{v})^T MP\mathbf{v} \\
&= -b(\mathbf{e}_\ell^T P\mathbf{v})^2 - b(\mathbf{e}_r^T P\mathbf{v})^2 - a(P\mathbf{v})^T MP\mathbf{v} \leq 0,
\end{aligned} \tag{6.47}$$

where we used (6.29) in the first step, the SBP property of D_2 in the second step, and thirdly that $LP\mathbf{v} = 0$, i.e., $a\mathbf{d}_\ell^T P\mathbf{v} = b\mathbf{e}_\ell^T P\mathbf{v}$ and $a\mathbf{d}_r^T P\mathbf{v} = -b\mathbf{e}_r^T P\mathbf{v}$. Finally, we used that M is positive semidefinite. This shows that the semi-discrete problem is stable.

7 SBP methods for 2D IBVP

We will now study finite difference methods in two spatial dimensions. For simplicity, we restrict ourselves to rectangular domains Ω:

$$\Omega = [0, W_x] \times [0, W_y] \subset \mathbb{R}^2. \tag{7.1}$$

The domain Ω is discretized with an $(m_x+1)\times(m_y+1)$-point grid. The four boundaries (West, East, South and North) are denoted Γ_W, Γ_E, Γ_S and Γ_N. The grid points are defined as

$$\begin{aligned} x_i &= ih_x, \quad i = 0,1,\ldots,m_x, \quad h_x = \tfrac{W_x}{m_x}, \\ y_j &= jh_y, \quad j = 0,1,\ldots,m_y, \quad h_y = \tfrac{W_y}{m_y}. \end{aligned}$$

The numerical approximation at grid point $(x_i,\, y_j)$ is denoted $v_{i,j}$. The discrete solution vector is given by $\mathbf{v}^T = [v_{0,0}, v_{0,1}, \ldots v_{0,m_y}, v_{1,0} \ldots, v_{1,m_y}, \ldots, v_{m_x,0} \ldots, v_{m_x,m_y}]$. The location of $v_{i,j}$ in terms of the computational domain can be visualized by the matrix representation (equivalent to a 2D plot) of the solution vector:

$$\mathbf{V} = \begin{bmatrix} v_{0,m_y} & v_{1,m_y} & \cdots & v_{m_x,m_y} \\ \vdots & \vdots & \vdots & \vdots \\ v_{0,1} & v_{1,1} & \cdots & v_{m_x,1} \\ v_{0,0} & v_{1,0} & \cdots & v_{m_x,0} \end{bmatrix}$$

7.1 The heat equation in 2D

As an example of a well-posed IBVP we consider the 2D heat equation on a rectangular domain:

$$\begin{aligned} cu_t = u_{xx} + u_{yy} + F, \quad & \mathbf{x} \in \Omega, \quad && t > 0, \\ \mathcal{L}u = g, \quad & \mathbf{x} \in \partial\Omega, \quad && t > 0, \\ u = u_0, \quad & \mathbf{x} \in \Omega, \quad && t > 0, \end{aligned} \tag{7.2}$$

where $\mathcal{L}$ is the boundary operator (that includes all four boundaries). The parameter $c > 0$ might vary in space. We will consider both Neumann and Dirichlet boundary conditions. Neumann boundary conditions (on all four boundaries) can be written,

$$\begin{aligned} u_x = g_W, \quad & \mathbf{x} \in \Gamma_W, \quad && t > 0, \\ u_x = g_E, \quad & \mathbf{x} \in \Gamma_E, \quad && t > 0, \\ u_y = g_S, \quad & \mathbf{x} \in \Gamma_S, \quad && t > 0, \\ u_y = g_N, \quad & \mathbf{x} \in \Gamma_N, \quad && t > 0, \end{aligned} \tag{7.3}$$

Dirichlet boundary conditions (on all four boundaries) would instead be

$$\begin{aligned} u = g_W, \quad & \mathbf{x} \in \Gamma_W, \quad && t > 0, \\ u = g_E, \quad & \mathbf{x} \in \Gamma_E, \quad && t > 0, \\ u = g_S, \quad & \mathbf{x} \in \Gamma_S, \quad && t > 0, \\ u = g_N, \quad & \mathbf{x} \in \Gamma_N, \quad && t > 0,. \end{aligned} \tag{7.4}$$

The energy method (assuming zero boundary data) for either Neumann or Dirichlet boundary conditions leads to

$$\frac{1}{2}\frac{\mathrm{d}}{\mathrm{d}t}\|u\|_c^2 = \int\limits_{\Gamma_E} u u_x \,\mathrm{d}S - \int\limits_{\Gamma_W} u u_x \,\mathrm{d}S + \int\limits_{\Gamma_N} u u_y \,\mathrm{d}S - \int\limits_{\Gamma_S} u u_y \,\mathrm{d}S - \|\nabla u\|^2 = -\|u_x\|^2 - \|u_y\|^2 \,. \tag{7.5}$$

7.2 Extending 1D operators to 2D

To extend 1D finite difference operators to 2D we can use the Kronecker product, denoted by $\otimes$, which for an $m \times n$ matrix A and any matrix B is defined as

$$A \otimes B = \begin{bmatrix} A_{11}B & A_{12}B & \dots & A_{1n}B \\ A_{21}B & A_{22}B & \dots & A_{2n}B \\ \vdots & & \ddots & \vdots \\ A_{m1}B & A_{m2}B & \dots & A_{mn}B \end{bmatrix}. \tag{7.6}$$

To distinguish whether a 2D difference operator S is operating in the x- or the y-direction, the notations S_x and S_y are used, respectively. The following 2D operators are frequently used:

$$\begin{array}{llll} D_x & = D_1 \otimes I_{m_y}, & D_y & = I_{m_x} \otimes D_1 \\ D_{2x} & = D_2 \otimes I_{m_y}, & D_{2y} & = I_{m_x} \otimes D_2 \\ H_x & = H \otimes I_{m_y}, & H_y & = I_{m_x} \otimes H \\ \mathbf{e}_W & = \mathbf{e}_\ell \otimes I_{m_y}, & \mathbf{e}_S & = I_{m_x} \otimes \mathbf{e}_\ell \\ \mathbf{e}_E & = \mathbf{e}_r \otimes I_{m_y}, & \mathbf{e}_N & = I_{m_x} \otimes \mathbf{e}_r \\ \mathbf{d}_W & = d_\ell \otimes I_{m_y}, & \mathbf{d}_S & = I_{m_x} \otimes d_\ell \\ \mathbf{d}_E & = d_r \otimes I_{m_y}, & \mathbf{d}_N & = I_{m_x} \otimes d_r \end{array} , \tag{7.7}$$

where $\mathbf{d}_\ell$, $\mathbf{e}_\ell$, $\mathbf{d}_r$, $\mathbf{e}_r$, D_1, D_2, and H are 1D operators, and I_{m_x}, I_{m_y} are identity matrices of appropriate sizes. The 2D boundary operators $\mathbf{e}_{S,E,S,N}$ pick out the boundary points at the West, East, South and North boundaries.

The interpretation of the Kronecker products above is as follows: The operator D_x corresponds to applying the 1D operator D_1 on every grid line in the x direction (every horizontal grid line). Similarly, D_y corresponds to applying D_1 on every vertical grid line.

Remark. In the Kronecker products defined in (7.7), D_1, D_2 and H are either of size $(m_x + 1) \times (m_x + 1)$ (if to the left of $\otimes$) or size $(m_y + 1) \times (m_y + 1)$ (if to the right of $\otimes$).

The 2D quadrature operator H_Ω is here given by the matrix-product:

$$H_\Omega = H_x H_y. \tag{7.8}$$

For a non-constant parameter c, we introduce the diagonal matrix C with $c = c(x, y)$

projected onto the diagonal:

$$C = \begin{bmatrix} c(x_0,y_0) & & & & & & & & \\ & \ddots & & & & & & & \\ & & c(x_0,y_{m_y}) & & & & & & \\ & & & c(x_1,y_0) & & & & & \\ & & & & \ddots & & & & \\ & & & & & c(x_1,y_{m_y}) & & & \\ & & & & & & \ddots & & \\ & & & & & & & c(x_{m_x},y_0) & \\ & & & & & & & & \ddots & \\ & & & & & & & & & c(x_{m_x},y_{m_y}) \end{bmatrix}.$$

7.3 SBP approximations of 2D heat equation

We will now discretize (7.2) combined with Neumann boundary conditions using the SBP-SAT method. The SBP-SAT discretization reads

$$\begin{aligned} C\mathbf{v}_t = D_{2x}\mathbf{v} + D_{2y}\mathbf{v} + \mathbf{F} + SAT, \quad t > 0, \\ \mathbf{v} = \mathbf{v}_0, \quad t = 0, \end{aligned} \tag{7.9}$$

where SAT contains the penalty terms that impose the BC. It is possible to construct SAT such that (7.9) is stable. However, it is simpler to utilize the fact that the finite difference method is essentially a 1D method. The extension to 2D and 3D amounts to applying the 1D method on every grid line.

Recall that the SBP-SAT approximation of the **1D** heat equation with homogeneous Neumann boundary conditions, and zero forcing F is

$$C\mathbf{v}_t = D_2\mathbf{v} + H^{-1}\mathbf{e}_\ell(\mathbf{d}_\ell^T\mathbf{v} - 0) - H^{-1}\mathbf{e}_r(\mathbf{d}_r^T\mathbf{v} - 0) = A_{1D}\mathbf{v}, \tag{7.10}$$

where we have defined the matrix

$$A_{1D} = D_2 + H^{-1}\mathbf{e}_\ell\mathbf{d}_\ell^T - H^{-1}\mathbf{e}_r\mathbf{d}_r^T. \tag{7.11}$$

The SBP-SAT approximation of (7.2) combined with homogeneous Neumann boundary conditions can be written

$$\begin{aligned} C\mathbf{v}_t = (A_{1D}^{m_x} \otimes I_{m_y} + I_{m_x} \otimes A_{1D}^{m_y})\mathbf{v} + \mathbf{F}, \quad t > 0. \\ \mathbf{v} = \mathbf{v}_0, \quad t = 0. \end{aligned} \tag{7.12}$$

Stability follows since HA_{1D} is negative semi-definite. It is possible to show that the discrete energy rate mimics the continuous energy rate, just like for 1D problems, but for the sake of brevity we omit that here.

8 Finite element methods

The Finite Element Method (FEM) is a general technique for computing numerical solutions to differential equations. Compared to the finite difference method, its main advantage is that is better at handling complex geometries. Its main disadvantage is that it is often more computationally expensive in the sense that it requires more floating point operations to satisfy a given error tolerance.

This chapter is heavily influenced by a book by Larson and Bengzon [4], which we recommend for further reading.

8.1 Piecewise linear functions

Before we dive into FEM, we briefly discuss piecewise linear functions. Consider an interval $\mathcal{I} = [a, b]$. Let us introduce a mesh of N subintervals,

$$a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b. \tag{8.1}$$

where $\{x_i\}_{i=0}^{N}$ are the $N + 1$ nodes of the mesh. Let $\mathcal{I}_i = [x_{i-1}, x_i]$ denote the ith subinterval, of length h_i. Let V_h be the space of all *continuous* functions that are *linear on each subinterval* of the mesh. Formally,

$$V_h = \left\{v : v \in C^0(\mathcal{I}),\ v|_{\mathcal{I}_i} \in P_1(\mathcal{I}_i)\,\forall i\right\}, \tag{8.2}$$

where $C^0(\mathcal{I})$ is the space of continuous functions on $\mathcal{I}$ and $P_1(\mathcal{I}_i)$ is the space of all linear functions on $\mathcal{I}_i$. The so-called *hat functions* φ_i, illustrated in Figure 8.1, form a basis in V_h. Mathematically, the hat functions are defined as

$$\varphi_i(x) = \begin{cases} \dfrac{x - x_{i-1}}{h}, & x \in [x_{i-1}, x_i) \\ \dfrac{x_{i+1} - x}{h}, & x \in [x_i, x_{i+1}) \\ 0, & \text{otherwise} \end{cases}. \tag{8.3}$$

Note that φ_0 and φ_N are only "half hats". Since $\{\varphi_i\}_{i=0}^{N}$ spans V_h, any function $v \in V_h$ can be expressed as a linear combination of the φ_i,

$$v(x) = \sum_{i=0}^{N} \alpha_i \varphi_i(x), \tag{8.4}$$

where the coefficients α_i are the nodal values of v:

$$v(x_i) = \alpha_i. \tag{8.5}$$

This follows from the fact that the basis functions are zero at all nodes except "their own":

$$\varphi_i(x_j) = \begin{cases} 1, & i = j, \\ 0, & i \neq j. \end{cases} \tag{8.6}$$

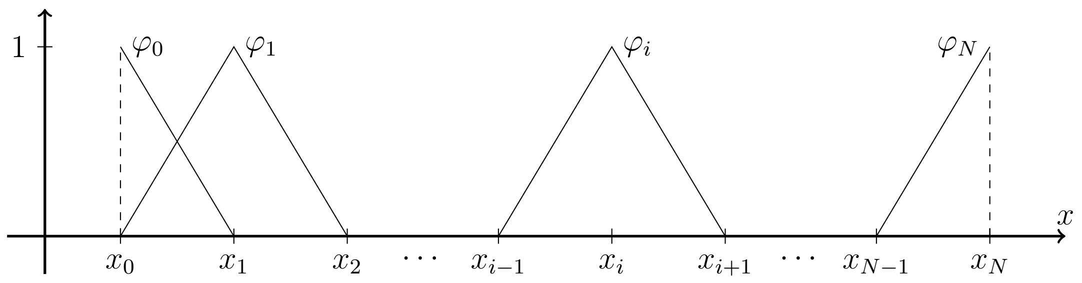


Figure 8.1: The "hat functions" $\{\varphi_i\}$ form a basis in the space of all piecewise linear functions on the partition $a = x_0 < x_1 < \cdots < x_N = b$.

8.2 FEM for the 1D Poisson equation

To introduce the finite element method, let us first study the Poisson equation in 1D:

$$\begin{aligned} -u''(x) &= f(x), \quad a < x < b, \\ u(a) = u(b) &= 0, \end{aligned} \tag{8.7}$$

where u is the (unknown) solution and f is a given function.

8.2.1 The weak form

The formulation (8.7) is known as the *strong form* of the problem. A finite element method, however, is always derived from the *weak* (also known as *variational*) formulation of the problem. To derive the weak form, let us first define the vector space

$$H_0^1 = \{v : \|v\|^2 + \|v'\|^2 < \infty,\ v(a) = v(b) = 0\}. \tag{8.8}$$

Next, we multiply the differential equation $-u'' = f$ by a so-called *test function* $v \in H_0^1$ and integrate, using the integration-by-parts formula:

$$\begin{aligned} \int_a^b fv \,\mathrm{d}x = \int_a^b -u''v &= -[u'v]_a^b + \int_a^b u'v' \,\mathrm{d}x \\ &= -u'(b)v(b) + u'(a)v(a) + \int_a^b u'v' \,\mathrm{d}x. \end{aligned} \tag{8.9}$$

Since $v(a) = v(b) = 0$, the boundary terms vanish and we are left with

$$\int_a^b fv \,\mathrm{d}x = \int_a^b u'v' \,\mathrm{d}x. \tag{8.10}$$

Hence, the *weak* (or *variational*) form of (8.7) reads:

Weak form
Find $u \in H_0^1$, such that

$$\int_a^b u'v' \,\mathrm{d}x = \int_a^b fv \,\mathrm{d}x, \tag{8.11}$$

for all $v \in H_0^1$.

The following terminology is often used regarding the weak form:

- u is the solution that we seek, the so-called *trial function*. It belongs to the *trial space* (here H_0^1), which has to satisfy the boundary conditions.
- v is called a *test function*. It belongs to a *test space* (here H_0^1).

Note that in this example the test and trial spaces are the same, but in general they may be different.

8.2.2 Finite element approximation

The space H_0^1 contains many functions and it is therefore just as hard to find a function $u \in H_0^1$ that satisfies the weak form (8.11) as it is to solve the original problem (8.7). The approximation of the finite element method is, therefore, to look for an approximate solution u_h within a small (finite-dimensional) subspace $V_{h,0} \subset H_0^1$, typically a space of piecewise polynomials.

Introduce a partition of the interval $[a, b]$ into N subintervals:

$$a = x_0 < x_1 < x_2 < \ldots < x_N = b. \tag{8.12}$$

For simplicity, we here assume that the mesh is uniform, which means that all subintervals are of equal length h and

$$x_j = a + jh, \quad h = \frac{b-a}{N}. \tag{8.13}$$

Let $V_{h,0}$ denote the space of all continuous piecewise linear functions that vanish at the endpoints a and b. The hat functions $\{\varphi_j\}_{j=1}^{N-1}$ (see Section 8.1) form a basis in $V_{h,0}$. Note that we here omit the "half-hats" φ_0 and φ_N, because the function is required to be zero at the endpoints. Otherwise $V_{h,0}$ would not be a subspace of H_0^1.

The finite element approximation of the weak form (8.11) thus reads:

Finite element approximation:
Find $u_h \in V_{h,0}$ such that

$$\int_a^b u_h'v' \,\mathrm{d}x = \int_a^b fv \,\mathrm{d}x, \tag{8.14}$$

for all $v \in V_{h,0}$.

8.2.3 Deriving the system of equations

It can be shown that (8.14) is equivalent to the $N-1$ equations

$$\int_a^b u_h' \varphi_i' \,\mathrm{d}x = \int_a^b f\varphi_i \,\mathrm{d}x, \quad i = 1, 2, \ldots, N-1, \tag{8.15}$$

because if (8.14) is satisfied for each basis function φ_i separately, then it is also satisfied for any linear combination of them, and hence any function $v \in V_{h,0}$.

Since we require $u_h \in V_{h,0}$, we know that u_h can be expressed as a linear combination of the φ_j. We therefore make the ansatz

$$u_h(x) = \sum_{j=1}^{N-1} \xi_j \varphi_j(x), \tag{8.16}$$

where ξ_j, $j = 1, 2, \ldots, N-1$ are $N-1$ unknown coefficients that we need to determine. Once they are known, the approximate solution is given by (8.16). Note that the ξ_j correspond to the values of u_h at the nodes of the mesh, since $u_h(x_j) = \xi_j$.

Inserting the ansatz (8.16) in (8.15) yields

$$\sum_{j=1}^{N-1} \xi_j \int_a^b \varphi_i' \varphi_j' \,\mathrm{d}x = \int_a^b f\varphi_i \,\mathrm{d}x \quad i = 1, 2, \ldots, N-1, \tag{8.17}$$

which is an $(N-1) \times (N-1)$ system of equations for ξ_j. In matrix form we write

$$A\boldsymbol{\xi} = \mathbf{b}, \tag{8.18}$$

where A is an $(N-1) \times (N-1)$ matrix, the so-called **stiffness** matrix, with entries

$$A_{i,j} = \int_a^b \varphi_i'(x)\, \varphi_j'(x) \,\mathrm{d}x, \quad i, j = 1, 2, \ldots, N-1, \tag{8.19}$$

$\boldsymbol{\xi} = (\xi_1, \xi_2, \ldots, \xi_{N-1})^T$ is an $(N-1)$ vector containing the unknown coefficients ξ_j and $\mathbf{b}$ is an $(N-1)$ vector, the so-called **load vector**, with entries

$$b_i = \int_a^b f(x)\varphi_i(x) \,\mathrm{d}x, \quad i = 1, 2, \ldots, N-1. \tag{8.20}$$

8.2.4 Evaluating the stiffness matrix and the load vector

The entries of A can here be computed analytically. The derivative of a basis function φ_i is

$$\varphi_i'(x) = \begin{cases} \dfrac{1}{h}, & x \in (x_{i-1}, x_i) \\ -\dfrac{1}{h}, & x \in (x_i, x_{i+1}) \\ 0, & \text{otherwise} \end{cases} \tag{8.21}$$

The entries of A_{ij} are thus

$$A_{i,i} = \int_a^b \varphi_i' \varphi_i' \,\mathrm{d}x = \int_{x_{i-1}}^{x_i} \frac{1}{h}\frac{1}{h} \,\mathrm{d}x + \int_{x_i}^{x_{i+1}} \frac{-1}{h}\frac{-1}{h} \,\mathrm{d}x = h\frac{1}{h^2} + h\frac{1}{h^2} = \frac{2}{h}, \tag{8.22}$$

$$A_{i,i+1} = \int_a^b \varphi_i' \varphi_{i+1}' \,\mathrm{d}x = \int_{x_i}^{x_{i+1}} \frac{-1}{h}\frac{1}{h} \,\mathrm{d}x = h\frac{-1}{h^2} = -\frac{1}{h}, \tag{8.23}$$

$$A_{i+1,i} = \int_a^b \varphi_{i+1}' \varphi_i' \,\mathrm{d}x = \int_{x_i}^{x_{i+1}} \frac{1}{h}\frac{-1}{h} \,\mathrm{d}x = h\frac{-1}{h^2} = -\frac{1}{h}. \tag{8.24}$$

Remaining entries A_{ij} with $|i-j| > 1$ equal 0 because ϕ_i' and ϕ_j' do not have common support unless $|i-j| \le 1$. A is therefore a tridiagonal matrix:

$$A = \frac{1}{h} \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}_{(N-1)\times(N-1)} \tag{8.25}$$

Since the load vector involves the right-hand side, f, which may be a complicated function, it is usually computed approximately by numerical integration. Using the trapezoidal rule yields

$$b_i = \int_a^b \varphi_i f \,\mathrm{d}x \approx h\left[\varphi_i(x_{i-1})f(x_{i-1}) + \varphi_i(x_i)f(x_i) + \varphi_i(x_{i+1})f(x_{i+1})\right] = hf(x_i). \tag{8.26}$$

The linear system that we need to solve for $\boldsymbol{\xi}$ is thus

$$\frac{1}{h} \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix} \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \\ \vdots \\ \xi_{N-2} \\ \xi_{N-1} \end{bmatrix} = h \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) \end{bmatrix}. \tag{8.27}$$

Remark. Note that the linear system in (8.27) looks like it could have originated from a finite difference method! After dividing through by h, the right-hand side is just f evaluated at the nodes and we recognize the system matrix as the second-order accurate operator D_+D_-. It is common that piecewise linear FEM and second-order FD yield very similar methods. Higher-order FEM uses piecewise polynomials of degree higher than 1, which typically is not quite equivalent to higher-order finite differences. FEM can also use unstructured meshes, which is not possible with traditional FD methods.

8.3 FEM for the stationary heat equation with variable coefficients and general boundary conditions

Let us know consider the stationary heat equation

$$\begin{aligned} -(a(x)u'(x))' &= f(x), \quad x \in (0, L) \\ u(0) &= g_0, \\ -a(L)u'(L) &= \gamma u(L) - g_L, \end{aligned} \tag{8.28}$$

Note that the left boundary condition (at $x = 0$) is a Dirichlet condition and the right boundary condition (at $x = L$) is a *Robin* condition. If we set $\gamma = 0$, the Robin condition reduces to a Neumann condition. Conversely, a large value of γ yields a condition of (almost) Dirichlet type.

8.3.1 The weak form

To derive the weak form, we multiply the differential equation by a test function v and integrate, using the integration by parts formula. With inner product notation for the integrals, we have

$$\begin{aligned} (v, f) = -\left(v, (au')'\right) = [\text{IBP}] &= -vau'|_0^L + (v', au') \\ &= -v(L)a(L)u'(L) + v(0)a(0)u'(0) + (v', au') \end{aligned} \tag{8.29}$$

At the right boundary, we may use the boundary condition to replace $-a(L)u'(L)$ by $\gamma u(L) - g_L$. At the left boundary, we can make the boundary term vanish by requiring that the test function satisfies a *homogeneous* Dirichlet condition, i.e. $v(0) = 0$. It is useful to remember that an inhomogeneous Dirichlet condition translates to a homogeneous Dirichlet condition on the test function. Using this, (8.29) simplifies to

$$(v, f) = v(L)(\gamma u(L) - g_L) + (v', au')\,. \tag{8.30}$$

It remains to define appropriate test and trial spaces. Let us first define the space of square-integrable functions with square-integrable first derivative:

$$H^1 = \{v : \|v\| + \|v'\| \leq \infty\}\,. \tag{8.31}$$

The conditions on $\|v\|$ and $\|v'\|$ ensure that the integrals in (8.30) are well-defined. Next, we need to take Dirichlet boundary conditions into account. The solution must satisfy the inhomogeneous condition $u(0) = g_0$. Thus, we define the trial space

$$V_{g_0} = \left\{v \in H^1 : v(0) = g_0\right\} \quad \text{(Trial space)}. \tag{8.32}$$

When deriving (8.30), we assumed that the test function satisfies a homogeneous Dirichlet condition. We therefore define the test space

$$V_0 = \left\{v \in H^1 : v(0) = 0\right\} \quad \text{(Test space)}. \tag{8.33}$$

Weak form
Find $u \in V_{g_0}$, such that

$$(v', au') + v(L)(\gamma u(L) - g_L) = (v, f)\,, \tag{8.34}$$

for all $v \in V_0$.

Note that only Dirichlet conditions need to appear in the definitions of test and trial spaces. Robin conditions are imposed by substituting in (8.29)

8.3.2 Finite element approximation

To demonstrate that FEM does not require a structured grid, let us use a non-uniform mesh:

$$0 = x_0 < x_1 \cdots < x_{N-1} < x_N = L. \tag{8.35}$$

Let $\mathcal{I}_i$ denote the ith subinterval, of length h_i:

$$\mathcal{I}_i = [x_{i-1},\, x_i), \quad h_i = x_i - x_{i-1}. \tag{8.36}$$

Next, we define the space of all continuous functions that are piecewise linear on the mesh:

$$V_h = \left\{ v \in C^0([0,\, L]),\; v|_{\mathcal{I}_i} \in P_1(\mathcal{I}_i)\ \forall i \right\}. \tag{8.37}$$

The trial space needs to satisfy the inhomogeneous Dirichlet condition while the test space satisfies a homogeneous Dirichlet condition

$$V_{h,g_0} = \{v \in V_h : v(0) = g_0\} \quad \text{(Trial space)}, \tag{8.38}$$

$$V_{h,0} = \{v \in V_h : v(0) = 0\} \quad \text{(Test space)}. \tag{8.39}$$

We obtain the finite element approximation by replacing test and trial spaces in the weak form (8.34) by the corresponding spaces of piecewise linear functions.

Finite element approximation:
Find $u_h \in V_{h,g_0}$, such that

$$(v', au_h') + v(L)(\gamma u_h(L) - g_L) = (v, f)\,, \tag{8.40}$$

for all $v \in V_{h,0}$.

8.3.3 Deriving the system of equations

Since $v \in V_{h,0}$, v can be expressed as a linear combination of the φ_i:

$$v = \sum_{i=1}^{N} \alpha_i \varphi_i. \tag{8.41}$$

Note that φ_0 is omitted, because $v \in V_{h,0}$ is required to be zero at the left boundary. By linearity, if the finite element approximation of the weak form (8.40) holds for all φ_i separately, then it holds for any $v \in V_{h,0}$. We obtain the N equations

$$(\varphi_i', a u_h') + \varphi_i(L)(\gamma u_h(L) - g_L) = (\varphi_i, f)\,, \quad i = 1, \dots, N. \tag{8.42}$$

Since $u_h \in V_{h,g_0}$, we make the ansatz

$$u_h = g_0 \varphi_0 + \sum_{j=1}^{N} \xi_j \varphi_j, \tag{8.43}$$

where the ξ_j, $j = 1, \dots, N$, are unknown coefficients. Notice that $\xi_0 = g_0$ is already determined by the boundary condition. Inserting the ansatz yields

$$\underbrace{(\varphi_i', a g_0 \varphi_0')}_{d_i} + \sum_{j=1}^{N} \underbrace{(\varphi_i', a\varphi_j')}_{A_{ij}} \xi_j + \underbrace{\varphi_i(L)\gamma}_{a_i} \xi_N - \underbrace{\varphi_i(L) g_L}_{r_i} = \underbrace{(\varphi_i, f)}_{b_i}, \quad i = 1, \dots, N, \tag{8.44}$$

where we used that $u_h(L) = \xi_N$. With the notation introduced in (8.44), the system of equations reads

$$\sum_{j=1}^{N} A_{ij}\xi_j + a_i \xi_N = b_i - d_i + r_i, \quad i = 1, \dots, N. \tag{8.45}$$

With homogeneous Dirichlet conditions on both sides, a_i, d_i and r_i would all drop out. The term d_i arises due to the inhomogeneous Dirichlet condition (notice that d_i is proportional to g_0) whereas a_i and r_i stem from the Robin condition. Since a_i multiplies one of the unknowns, ξ_N, it can be viewed as a boundary correction to the stiffness matrix. We have

$$a_i = \varphi_i(L)\gamma = \left\{ \begin{array}{ll} \gamma, & i = N \\ 0, & i < N \end{array} \right. , \tag{8.46}$$

$$r_i = \varphi_i(L) g_L = \left\{ \begin{array}{ll} g_L, & i = N \\ 0, & i < N \end{array} \right. , \tag{8.47}$$

$$d_i = (\varphi_i', a g_0 \varphi_0') = \left\{ \begin{array}{ll} (\varphi_1', a\varphi_0')\, g_0, & i = 1 \\ 0, & i > 1 \end{array} \right. , \tag{8.48}$$

where we used that ϕ_0 does not overlap with ϕ_i for $i > 1$ to simplify d_i. The system of equations (8.45) can be written in matrix-vector form as

$$\widetilde{A}\boldsymbol{\xi} = \widetilde{\mathbf{b}}, \tag{8.49}$$

where

$$\widetilde{A}_{ij} = \begin{cases} A_{ij} + \gamma, & i = j = N \\ A_{ij}, & \text{otherwise} \end{cases}, \quad \widetilde{b}_i = \begin{cases} b_i - (\varphi_1', a\varphi_0')\, g_0, & i = 1 \\ b_i + g_L, & i = N \\ b_i, & \text{otherwise} \end{cases}. \tag{8.50}$$

8.3.4 Evaluating the stiffness matrix and the load vector

With a non-uniform mesh, the derivatives of the basis functions are

$$\varphi_i' = \begin{cases} \dfrac{1}{h_i}, & x \in \mathcal{I}_i \\ \dfrac{-1}{h_{i+1}}, & x \in \mathcal{I}_{i+1} \end{cases}. \tag{8.51}$$

Since we do not expect to be able to integrate the variable coefficient $a(x)$ analytically, in general, it is convenient to use numerical integration. Let a_i denote the value of $a(x)$ at the midpoint of subinterval $\mathcal{I}_i$, i.e.,

$$a_i = a\left(\frac{x_{i-1} + x_i}{2}\right). \tag{8.52}$$

With the mid-point quadrature formula, we obtain the approximation

$$\begin{aligned} A_{ii} = (\varphi_i', a\varphi_i') &= \int_{x_{i-1}}^{x_i} a\varphi_i'\varphi_i' \,\mathrm{d}x + \int_{x_i}^{x_{i+1}} a\varphi_i'\varphi_i' \,\mathrm{d}x \\ &\approx h_i a_i \frac{1}{h_i^2} + h_{i+1}a_{i+1}\left(\frac{-1}{h_{i+1}}\right)^2 \\ &= \frac{a_i}{h_i} + \frac{a_{i+1}}{h_{i+1}}, \quad i = 1, \dots, N-1. \end{aligned} \tag{8.53}$$

The last diagonal element only gets a contribution from subinterval $\mathcal{I}_N$ since the last basis function φ_N is only a half-hat:

$$A_{NN} = (\varphi_N', a\varphi_N') = \int_{x_{N-1}}^{x_N} a\varphi_i'\varphi_i' \,\mathrm{d}x \approx h_N a_N \frac{1}{h_N^2} = \frac{a_N}{h_N}. \tag{8.54}$$

As before, the off-diagonal elements are zero for $|i-j| > 1$, since φ_i and φ_j do not overlap in this case. For $|i-j| = 1$, we have

$$A_{i,i+1} = \left(\varphi_i', a\varphi_{i+1}'\right) = \int_{x_i}^{x_{i+1}} a\varphi_i'\varphi_{i+1}' \,\mathrm{d}x \approx h_{i+1}a_{i+1}\frac{-1}{h_{i+1}}\frac{1}{h_{i+1}} = -\frac{a_{i+1}}{h_{i+1}}, \tag{8.55}$$

and

$$A_{i+1,i} = A_{i,i+1} \approx -\frac{a_{i+1}}{h_{i+1}}. \tag{8.56}$$

To summarize, the corrected stiffness matrix is

$$\widetilde{A} = \begin{bmatrix} \frac{a_1}{h_1} + \frac{a_2}{h_2} & -\frac{a_2}{h_2} & & & & \\ -\frac{a_2}{h_2} & \frac{a_2}{h_2} + \frac{a_3}{h_3} & -\frac{a_3}{h_3} & & & \\ & -\frac{a_3}{h_3} & \frac{a_3}{h_3} + \frac{a_4}{h_4} & -\frac{a_4}{h_4} & & \\ & & \ddots & \ddots & \ddots & \\ & & & -\frac{a_{N-1}}{h_{N-1}} & \frac{a_{N-1}}{h_{N-1}} + \frac{a_N}{h_N} & -\frac{a_N}{h_N} \\ & & & & -\frac{a_N}{h_N} & \frac{a_N}{h_N} + \gamma \end{bmatrix}. \tag{8.57}$$

Notice the boundary correction γ, which stems from the Robin condition, in the bottom right corner.

It remains to compute the load vector $\widetilde{b}$. The elements b_i are almost the same as in Section 8.2.4. The only difference is that we are here using a non-uniform mesh. The trapezoidal rule yields

$$b_i = (\varphi_i, f) = \int_{x_{i-1}}^{x_i} \varphi_i f \, \mathrm{d}x + \int_{x_i}^{x_{i+1}} \varphi_i f \, \mathrm{d}x \approx \frac{h_i + h_{i+1}}{2} f(x_i), \quad i = 1, \dots, N-1, \tag{8.58}$$

$$b_N = (\varphi_N, f) = \int_{x_{N-1}}^{x_N} \varphi_N f \, \mathrm{d}x \approx \frac{h_N}{2} f(x_N). \tag{8.59}$$

The boundary correction d_1 can be computed in the same way as the entries of the stiffness matrix:

$$d_1 = (\varphi_1', a g_0 \varphi_0') \approx -g_0 \frac{a_1}{h_1}. \tag{8.60}$$

Thus, the corrected load vector is

$$\widetilde{b} = \frac{1}{2} \begin{bmatrix} f(x_1)(h_1 + h_2) \\ f(x_2)(h_2 + h_3) \\ \vdots \\ f(x_{N-1})(h_{N-1} + h_N) \\ f(x_N) h_N \end{bmatrix} + \begin{bmatrix} g_0 \frac{a_1}{h_1} \\ \\ \vdots \\ \\ g_L \end{bmatrix}. \tag{8.61}$$

8.4 FEM for the heat equation

We will now study FEM for a time-dependent problem: the heat equation. For simplicity, we assume constant coefficients and homogeneous Dirichlet conditions:

$$\begin{aligned} u_t - a u_{xx} = f, &\quad x \in (0, L), \quad t > 0, \\ u = u_0, &\quad x \in [0, L], \quad t = 0, \\ u = 0, &\quad x = 0, \qquad\quad t > 0, \\ u = 0, &\quad x = L, \qquad\quad t > 0, \end{aligned} \tag{8.62}$$

where a is a positive constant. We will use FEM to discretize in space while leaving time continuous. Both trial and test functions now depend on space and time:

$$u = u(x, t), \quad v = v(x, t). \tag{8.63}$$

We will use the inner product and norm notation for integration over space, i.e.,

$$(u, v) = \int_0^L u(x, t) v(x, t)\, \mathrm{d}x, \tag{8.64}$$

$$\|u\|^2 = (u, u)\,. \tag{8.65}$$

Notice that e.g. $\|u\|$ is a function of time, but not space. For brevity, we avoid writing out the dependence on time explicitly.

8.4.1 The weak form

Just like for time-independent problems, the first step is to derive the weak form. Let us introduce the space V:

$$V = \{v : \|v\| + \|v_x\| \leq \infty\}\,. \tag{8.66}$$

We further need a space of functions that satisfy the Dirichlet boundary conditions:

$$V_0 = \{v \in V : v(0, t) = v(L, t) = 0\}\,. \tag{8.67}$$

Just like for time-independent problems, we multiply the PDE by a test function and integrate over space, which leads to

$$\begin{aligned} (v, f) &= (v, u_t) - a\,(v, u_{xx}) = [\text{IBP}] \\ &= (v, u_t) - a v u_x|_0^L + a\,(v_x, u_x) \\ &= (v, u_t) + a\,(v_x, u_x) \end{aligned} \tag{8.68}$$

where we assumed $v \in V_0$ so that the boundary terms vanish. We have derived the weak form.

Weak form
Find $u \in V_0$, such that

$$(v, u_t) + a\,(v_x, u_x) = (v, f)\,, \tag{8.69}$$

for all $v \in V_0$.

8.4.2 Finite element approximation

Introduce a mesh of N intervals in space. Let $V_{h,0}$ denote the space of all functions $v(x,t)$ such that

i) v is a continuous function of x on $[0, L]$,

ii) v is a linear function of x on every subinterval of the mesh, and

iii) $v(0,t) = v(L,t) = 0$.

The finite element approximation of the weak form is:

Finite element approximation:
Find $u_h \in V_{h,0}$, such that

$$(v, u_{h,t}) + a\,(v_x, u_{h,x}) = (v, f) \tag{8.70}$$

for all $v \in V_{h,0}$.

For every fixed time, u_h and v are piecewise linear functions of x, just like in previous sections on time-independent problems. This implies that u_h and v can be expressed as linear combinations of the hat basis function with *time-dependent* coefficients:

$$u_h(x,t) = \sum_{j=1}^{N-1} \xi_j(t)\varphi_j(x), \tag{8.71}$$

$$v(x,t) = \sum_{i=1}^{N-1} \alpha_i(t)\varphi_i(x). \tag{8.72}$$

8.4.3 Deriving the system of ODEs

By linearity, the finite element approximation (8.70) must hold for every basis function that appears in (8.72) separately:

$$(\varphi_i, u_{h,t}) + a\,(\varphi_i', u_{h,x}) = (\varphi_i, f)\,, \quad i = 1, \ldots, N-1. \tag{8.73}$$

Next, we want to insert the ansatz (8.71). First, note that the derivatives of u_h are

$$u_{h,t}(x,t) = \sum_{j=1}^{N-1} \xi_j'(t)\varphi_j(x), \tag{8.74}$$

$$u_{h,x}(x,t) = \sum_{j=1}^{N-1} \xi_j(t)\varphi_j'(x). \tag{8.75}$$

Inserting the ansatz leads to

$$\sum_{j=1}^{N-1} \underbrace{(\varphi_i,\varphi_j)}_{M_{ij}} \xi_j'(t) + \sum_{j=1}^{N-1} \underbrace{a\left(\varphi_i',\varphi_j'\right)}_{A_{ij}} \xi_j(t) = \underbrace{(\varphi_i, f(t))}_{b_i}, \quad i = 1,\dots,N-1, \tag{8.76}$$

which is a system of ODEs for the unknown coefficients ξ_j. In matrix-vector form, the system reads

$$M\boldsymbol{\xi}'(t) + A\boldsymbol{\xi}(t) = \mathbf{b}(t). \tag{8.77}$$

The initial condition for ξ can obtained by projecting the initial data, u_0, onto $V_{h,0}$. Notice that A and $\mathbf{b}$ are the same stiffness matrix and load vector that we encountered for time-independent problems. M is known as the **mass matrix**. Just like A, M is tridiagonal because (φ_i,φ_j) is zero for $|i-j|>1$. The presence of M in (8.77) means that evaluating $\boldsymbol{\xi}'(t)$ requires solving a linear system with M. This linear solve appears in every time step and therefore increases the computational cost significantly for certain kinds of problems.

Remark. Finite difference methods also come with a "mass matrix", although it is typically not referred to as such. In SBP-FD methods, the quadrature operator H is the equivalent of a mass matrix. Since H is diagonal, inverting it is trivial. This avoids the linear solve.

8.5 FEM for the wave equation in 2D

Consider a bounded domain $\Omega \subset \mathbb{R}^2$, with boundary $\partial\Omega$ and outward unit normal $\hat{\mathbf{n}}$, as illustrated in Figure 8.2. Consider the acoustic wave equation on Ω:

$$\begin{array}{rll} K^{-1}\phi_{tt} = \nabla\cdot\rho^{-1}\nabla\phi, & (x,y)\in\Omega, & t>0 \\ \phi = 0, & (x,y)\in\partial\Omega, & t>0 \\ \phi = \phi_0, & (x,y)\in\Omega, & t=0 \\ \phi_t = \eta_0, & (x,y)\in\Omega, & t=0 \end{array}. \tag{8.78}$$

8.5.1 The weak form

As always, we derive the weak form by multiplying the PDE by a test function and integrating over space, which leads to

$$\left(v, K^{-1}\phi_{tt}\right) = \left(v, \nabla\cdot\rho^{-1}\nabla\phi\right). \tag{8.79}$$

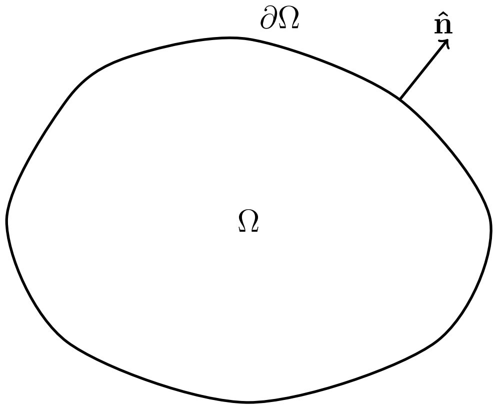


Figure 8.2: A bounded domain $\Omega \in \mathbb{R}^2$

The next step is to integrate by parts. Recall the following integration-by-parts formula for scalar functions u, w, and α:

$$\int\limits_{\Omega} u \nabla \cdot \alpha \nabla w \, \mathrm{d}\Omega = \oint\limits_{\partial\Omega} u \alpha \frac{\partial w}{\partial \hat{\mathbf{n}}} \, \mathrm{d}S - \int\limits_{\Omega} \nabla u \cdot \alpha \nabla w \, \mathrm{d}\Omega. \tag{8.80}$$

With inner product notation (assuming that u is real-valued), the formula can be written

$$(u, \nabla \cdot \alpha \nabla w)\, \Omega = \oint\limits_{\partial\Omega} u \alpha \frac{\partial w}{\partial \hat{\mathbf{n}}} \, \mathrm{d}S - (\nabla u, \alpha \nabla w)\,. \tag{8.81}$$

Using (8.81) in (8.79) yields

$$\left(v, K^{-1}\phi_{tt}\right) = \underbrace{\oint\limits_{\partial\Omega} v \rho^{-1} \frac{\partial \phi}{\partial \hat{\mathbf{n}}} \, \mathrm{d}S}_{=0 \text{ if } v=0 \text{ on } \partial\Omega} - \left(\nabla v, \rho^{-1} \nabla \phi\right). \tag{8.82}$$

The boundary condition is a homogeneous Dirichlet condition and we can see that the boundary integral vanishes if the test function also satisfies a homogeneous Dirichlet condition. We therefore introduce the space

$$H_0^1 = \{v : \|v\| + \|\nabla v\| < \infty, v = 0 \text{ on } \partial\Omega\}\,, \tag{8.83}$$

which here will serve as both test and trial space.

Weak form
Find $\phi \in H_0^1$, such that

$$\left(v, K^{-1}\phi_{tt}\right) = -\left(\nabla v, \rho^{-1} \nabla \phi\right), \tag{8.84}$$

for all $v \in H_0^1$.

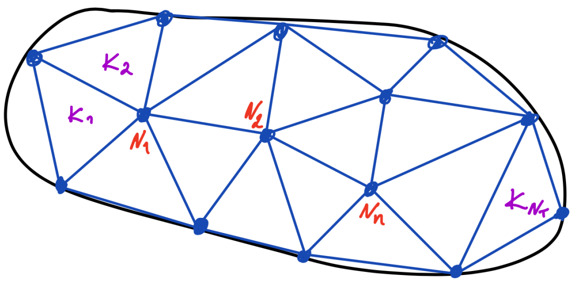


Figure 8.3: A triangular mesh in 2D

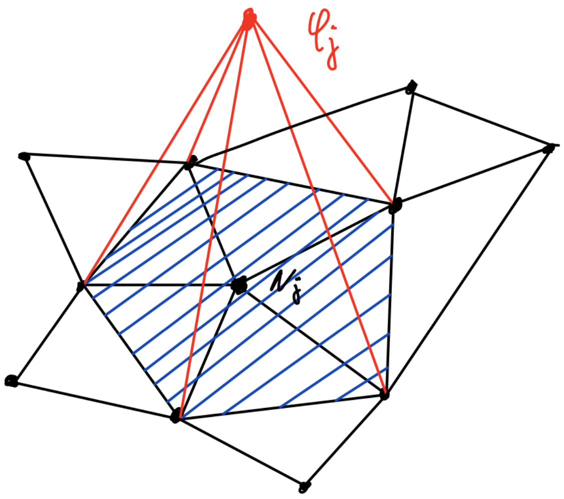


Figure 8.4: A "tent" basis function in 2D

8.5.2 Finite element approximation

We need to construct a finite-dimensional subspace of H_0^1. We start by dividing Ω into a set of triangles, a so-called **triangulation** or **mesh**, as illustrated in Figure 8.3. Let the mesh consist of N_T triangles called K_1, K_2, …, K_{N_T}. Let us define the space $V_{0,h}$, consisting of all functions that are continuous on Ω, linear on every triangle, and zero on $\partial\Omega$:

$$V_{0,h} = \left\{ v \in C^0(\Omega) : v|_{K_i} \in P_1(K_i)\, \forall i,\, v = 0 \text{ on } \partial\Omega \right\}. \tag{8.85}$$

Assume that the mesh has n interior nodes N_i, $i = 1, \ldots n$. The "tent functions" (the 2D version of the hat functions) $\varphi_j, j = 1, \ldots, n$, form a basis in $V_{h,0}$. One of the tent functions is illustrated in Figure 8.4. The φ_j are piecewise linear (linear on every triangle) and satisfy

$$\varphi_j(N_i) = \left\{ \begin{array}{ll} 1, & i = j \\ 0, & i \neq j \end{array} \right. . \tag{8.86}$$

As always, we obtain the finite element approximation by replacing trial and test spaces in the weak form by appropriate finite-element spaces. We obtain:

Finite element approximation:
Find $\phi_h \in V_{0,h}$, such that

$$\left(v, K^{-1}\phi_{h,tt}\right) = -\left(\nabla v, \rho^{-1}\nabla\phi_h\right), \tag{8.87}$$

for all $v \in V_{0,h}$.

8.5.3 Deriving the system of ODEs

Since the φ_j form a basis in $V_{h,0}$, any $v \in V_{h,0}$ can be expresed as a linear combination:

$$v = \sum_{i=1}^{n} \alpha_i \varphi_i. \tag{8.88}$$

Similarly, we make the following ansatz for the solution:

$$\phi_h(x,y,t) = \sum_{j=1}^{n} \xi_j(t)\varphi_j(x,y). \tag{8.89}$$

Inserting the ansatz in (8.87) yields the system of equations

$$\sum_{j=1}^{n} \underbrace{\left(\varphi_i, K^{-1}\varphi_j\right)}_{M_{ij}} \xi_j''(t) = -\sum_{j=1}^{n} \underbrace{\left(\nabla\varphi_i, \rho^{-1}\nabla\varphi_j\right)}_{A_{ij}} \xi_j(t) \quad i = 1,\dots,n, \tag{8.90}$$

which in matrix-vector form reads

$$M\boldsymbol{\xi}''(t) = -A\boldsymbol{\xi}(t). \tag{8.91}$$

This is a second-order ODE. The mass matrix is non-diagonal (in 2D the structure is more complicated than simply tri-diagonal) and causes a linear solve in every time step.

8.6 Finite elements or finite differences?

In this section we briefly compare finite element and finite difference methods. We saw in Section 8.2.4 that for the 1D Poisson equation with a uniform mesh, 2nd order FD and 2nd order FEM yield the same system of equations (at least in the interior of the domain). This shows that in certain idealized cases, FD and FEM are actually quite similiar. However, in 2D and 3D, the properties of FEM depend a lot on the mesh. Unless the mesh is structured, almost like an FD grid, then FD and FEM become quite different.

Pros and cons of FEM

+ **Can handle complex geometries**. Triangular meshes offer great geometric flexibility. It is much harder to create a good finite difference grid for complex geometries.

- **Mass matrix in front of time derivative term in time-dependent PDE**. Examples:

$$M\frac{\mathrm{d}\boldsymbol{\xi}}{\mathrm{d}t} = -A\boldsymbol{\xi} \quad \text{(heat equation)}$$

$$M\frac{\mathrm{d}^2\boldsymbol{\xi}}{\mathrm{d}t^2} = -c^2A\boldsymbol{\xi} \quad \text{(wave equation)}$$

The mass matrix M causes a linear solve in every time step, which makes the method **expensive**. (There are ways to avoid the solve, e.g. "mass lumping", with other drawbacks...)

Inverting the mass matrix is particularly costly for PDE that are well suited for **explicit** time-stepping. This includes hyperbolic PDE (wave propagation problems), where the CFL condition dictates

$$\Delta t \sim \frac{h}{c}.$$

For the heat equation, for example, explicit time-stepping requires much smaller time steps:

$$\Delta t \sim h^2.$$

Implicit time-stepping methods, which allow for larger time-steps, are often a good choice for the heat equation. Since implicit methods require a solve anyway, the mass matrix is not a big problem.

Based on the discussion above, we can identify which problems are good fits for FD and FEM, respectively:

- **Perfect problem for FD**: Wave-dominated (hyperbolic) PDE in simple or moderately complex geometry.
- **Perfect problem for FEM**: Time-independent or diffusion-dominated PDE in complex geometry.

Many PDEs include both wave-like and diffusion-like behavior. One example is the Navier–Stokes equations, which are used extensively in computational fluid dynamics (CFD). Many versions of FEM and FDM exist, and it is not obvious which one is for choice.

We end this section by commenting on low-order methods vs high-order methods.

- Generally speaking, the lower the error tolerance, the higher the ideal order of your method. If a 1% error is tolerable, a method of relatively low order will probably be good.
- 2nd-order methods are often quite inaccurate. For FD methods, going to at least 4th order often yields a big improvement.

9 Time integration

Thus far, we have talked about spatial discretization. The resulting semi-discrete problem is a system of ODEs, which must be solved (or time-integrated) by some appropriate ODE solver. There are many alternatives. First and foremost, we must decide whether to use an *explicit* or an *implicit method*. Implicit methods generally allow for larger time steps but require solving systems of equations and are therefore more computationally costly per step.

Assuming that the semidiscrete problem is stable, most implicit methods are stable for any time step. The optimal time step is therefore determined by accuracy. We want to select a time step size such that the error from the time discretization matches that from the spatial discretization.

Explicit methods, on the other hand, suffer from a stability restriction on the time step of the form $\Delta t \leq \Delta t_{max}$, where Δt_{max} denotes the largest stable time step. Of course, we must also take accuracy into account, but the stability restriction is typically so severe that it is ideal to use a time step close to the stability limit (for example $\Delta t \approx 0.8\Delta t_{max}$).

9.1 The CFL condition for hyperbolic PDEs

For hyperbolic PDEs, the stability condition on the time step for explicit methods is of the form

$$\Delta t \leq K\frac{h}{c}, \tag{9.1}$$

where h is the grid spacing and c is the wave speed. The dimensionless constant K is called the Courant number and is typically of order 1. The exact value of K depends both on the spatial discretization and the time-integration method.

The stability condition (9.1) was first stated by Courant, Friedrichs, and Lewy in 1928 and is commonly known as the *CFL* condition. They used a geometric argument, which, loosely speaking, states that the time step cannot be so large that information travels between "several" grid points in a single time step (exactly what is meant by "several" depends on the problem). The time required for a wave to travel between two adjacent grid points is h/c, and the requirement (9.1) with $K \approx 1$ follows.

Examples of hyperbolic PDEs include the advection equation:

$$u_t + cu_x = 0, \tag{9.2}$$

and the wave equation:

$$u_{tt} + c^2 u_{xx} = 0. \tag{9.3}$$

If h and/or c are variable, the CFL condition holds with the smallest ratio of h/c on the grid, i.e.,

$$\Delta t \leq K \min\left(\frac{h}{c}\right). \tag{9.4}$$

9.2 General time step restrictions

PDEs that are not hyperbolic come with different stability restrictions on the time step. To study other PDEs, let us first use the advection equation as an example and try to re-derive the CFL condition. With zero forcing, the PDE reads

$$u_t = -cu_x, \tag{9.5}$$

and we here consider periodic boundary conditions for simplicity. The semidiscrete problem can be written

$$\mathbf{v}_t = -cD\mathbf{v}, \tag{9.6}$$

where D could be a central difference operator such as the 2nd-order accurate D_0:

$$D = D_0 = \frac{1}{2h}\begin{bmatrix} 0 & 1 & & & & & -1 \\ -1 & 0 & 1 & & & & \\ & -1 & 0 & 1 & & & \\ & & & \ddots & & & \\ & & & & -1 & 0 & 1 \\ 1 & & & & & -1 & 0 \end{bmatrix} \tag{9.7}$$

Since the elements of D are proportional to h^{-1} (this is true for any order of accuracy), we expect the largest-magnitude eigenvalue of D to be proportional to h^{-1}.

Formally, let D be $m \times m$ and let

$$\lambda_i, \quad i = 1, 2 \ldots, m, \tag{9.8}$$

denote the eigenvalues D. Let $\rho(D)$ denote the *spectral radius* of D, which is defined as the absolute value of the largest-magnitude eigenvalue, i.e.,

$$\rho(D) = \max_i |\lambda_i|. \tag{9.9}$$

Since all entries of D are proportional to h^{-1}, we expect

$$\rho(D) \approx ah^{-1}. \tag{9.10}$$

for some constant a.

For a system of ODEs of the form

$$\mathbf{v}_t = A\mathbf{v}, \tag{9.11}$$

the stability restriction on the time step Δt is that all eigenvalues of $\Delta t A$ must be contained inside the stability region of the ODE method. Figure 9.1 shows the stability regions for three different Runge–Kutta methods. In the figure, k denotes the time step and λ an eigenvalue of A. One can determine an approximate "radius" R of the stability

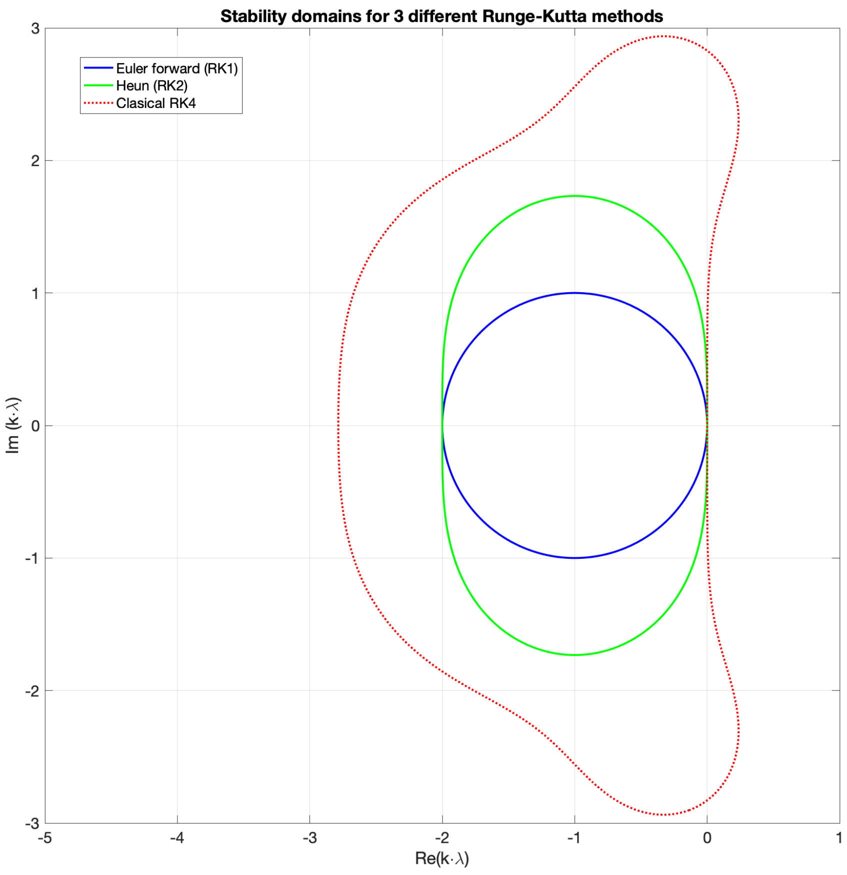


Figure 9.1: Stability regions for RK1 (Euler forward), RK2 (Heun's method), and RK4

region. For RK4, we have $R \approx 2.78$. For stability, we therefore need

$$\Delta t \rho(A) \leq R, \tag{9.12}$$

or, equivalently,

$$\Delta t \leq \frac{R}{\rho(A)}. \tag{9.13}$$

For our ODE system, the matrix is $A = -cD$ and the spectral radius is approximately

$$\rho(A) = \rho(-cD) = c\rho(D) \approx cah^{-1}. \tag{9.14}$$

We therefore obtain the (approximate) stability condition

$$\Delta t \leq \frac{Rh}{ac}. \tag{9.15}$$

That is, we have derived the CFL condition with Courant number $K = R/a$.

Notice that the condition (9.13) is for a general system of ODEs of the form $\mathbf{v}_t = A\mathbf{v}$. Consider, for example, a discretization of the heat equation:

$$u_t = bu_{xx} \longrightarrow \mathbf{v}_t = bD_2\mathbf{v}. \tag{9.16}$$

Since the entries of D_2 are proportional to h^{-2}, we expect $\rho(D_2) \approx \alpha h^{-2}$, for some constant α. This leads to the time step restriction

$$\Delta t \leq \frac{Rh^2}{b\alpha}. \tag{9.17}$$

Notice that Δt here needs to be proportional to h^2. This restriction is much more severe than the CFL condition that we saw for hyperbolic PDEs (the heat equation is a parabolic equation). Consider refining the grid by a factor of 10. For a hyperbolic problem, we would need to decrease Δt by a factor of 10, but for the heat equation we would need to decrease Δt by a factor of $10^2 = 100$. Due to this severe restriction, implicit ODE solvers are much more attractive for the heat equation than for hyperbolic problems.

10 Iterative methods and factorizations for sparse linear systems of equations

We consider linear systems of equations

$$Ax = b, \tag{10.1}$$

where

- A is a *sparse* $m \times m$ matrix,
- b is a known $m \times 1$ vector, and
- x is the unknown $m \times 1$ vector.

The computational cost of a direct solve for x (using Gaussian elimination) is $\mathcal{O}(m^3)$ arithmetic operations.

The fact that A is sparse means that most of its entries are 0. Finite difference and finite element discretizations of PDEs frequently yield *large* ($m > 10^6$) sparse matrices. In a computer implementation, these matrices must be stored in sparse format, since storing a full $10^6 \times 10^6$ matrix in double precision would require $10^{12} \cdot 8\text{B} = 8\text{TB}$ of memory!

Computing A^{-1} explicitly is usually a bad idea. First of all, computing A^{-1} requires $\mathcal{O}(m^3)$ operations. Moreover, since A^{-1} is typically dense (even if A is sparse), storing A^{-1} would require storing $\mathcal{O}(m^2)$ floating point numbers, which may be impossible if m is large.

10.1 LU-factorization

A common situation is that we need to solve a sequence of s systems,

$$Ax_i = b_i, \quad i = 1, 2, \ldots, s, \tag{10.2}$$

where the matrix A remains the same but the right-hand side b changes. This happens when using FEM with explicit time-stepping (in which case the system matrix is the mass matrix, M) or when solving any linear PDE with *implicit* time-stepping. In these scenarios, one approach is to *factorize* the matrix A.

With LU-factorization, the basic idea is to write

$$A = LU \tag{10.3}$$

where

- L is a lower-triangular matrix (all elements above the main diagonal are zero)
- U is an upper-triangular matrix (all elements above the main diagonal are zero).

In practice, it is necessary to include partial pivoting, which amounts to a permutation of the rows of A. It turns out that any square matrix can be factorized as

$$PA = LU, \tag{10.4}$$

where P is a permutation matrix. Note that the factorization process is expensive; it requires $\mathcal{O}(m^3)$ operations. (The factorization is essentially obtained via Gaussian elimination. L and U contain the factors that are used in the elimination process.) However, once A has been factorized, the system $Ax = b$ can be solved with only $\mathcal{O}(m^2)$ operations. To see this, first note that $Ax = b$ is equivalent to $LUx = Pb$. After introducing the intermediate variable $y = Ux$, we can solve this system efficiently in a two-step procedure:

1. Solve $Ly = Pb$ for y.
2. Solve $Ux = y$ for x.

The cost of solving a *triangular* system is only $\mathcal{O}(m^2)$ operations. To see this, note that L has the structure.

$$L = \begin{bmatrix} \times & 0 & 0 & \dots & 0 \\ \times & \times & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & & \vdots \\ \times & \times & \times & \dots & \times \end{bmatrix}. \tag{10.5}$$

We can use the first row to solve for the first unknown, x_1. Once x_1 is given, we can use the second row to solve for x_2. With both x_1 and x_2 given, we can solve the third row for x_3, and so on. This algorithm is known as *forward substitution* and requires only $\mathcal{O}(m^2)$ operations. This shows the benefit of solving systems with triangular matrices. A system with U can similarly be solved by so-called *backward substitution*, where one first computes x_m (using the last row), then x_{m-1}, and so on.

To solve a sequence of s systems as in (10.2), the computational cost is

- $\mathcal{O}(s \cdot m^3)$ with Gaussian elimination.
- $\mathcal{O}(m^3) + \mathcal{O}(s \cdot m^2)$ with LU-factorization.

Note that with LU-factorization, it is important to perform the factorization (which is expensive) only *once*, in the beginning, and then use the factors L, U, P to solve the sequence of systems.

10.2 Direct and iterative methods

Both LU-factorization and Gaussian elimination are examples of direct methods.

Definition 8 (Direct method)**.** A *direct* method solves the system (10.1) *exactly* after a *fixed* number of operations.

Iterative methods are fundamentally different from direct methods and utilize the fact that we typically are satisfied with a sufficiently accurate approximation $x_{approx} \approx x$.

Definition 9 (Iterative method)**.** An *iterative* method starts with a guess, $x^{(0)}$, and computes a sequence of approximations $x^{(1)}$, $x^{(2)}$, $\ldots$. The iteration terminates when some stopping criterion is satisfied.

The idea behind iterative methods is to improve the current guess with every iteration so that $x^{(k)} \to x$ as $k \to \infty$. Ideally, we want to stop when the error is less than some tolerance, i.e., when $\|x^{(k)} - x\| \leq \text{tol}$. However, since we do not have access to the true solution x, the most common *stopping criterion* is instead

$$\|x^{(k)} - x^{(k-1)}\| \leq \text{tol}. \tag{10.6}$$

In general, direct methods are fast for small systems, but for *large* ($m > 10^6$), *sparse* systems, iterative methods are often faster and use less memory. As a rule of thumb, matrices arising from PDE discretizations in 1D and 2D can often be solved efficiently with direct methods. For 3D problems, however, iterative methods are usually needed.

10.3 Basic iterative methods

Several iterative methods are based on the following idea: if $A = A_1 + A_2$, with A_1 invertible, then

$$Ax = b \Leftrightarrow (A_1 + A_2)x = b \Leftrightarrow x = -A_1^{-1}(A_2 x - b). \tag{10.7}$$

The final equation in (10.7) suggests the iteration:

$$x^{(k+1)} = -A_1^{-1}(A_2 x^{(k)} - b). \tag{10.8}$$

It is not obvious whether (10.8) converges (that is, $x^{(k)} \to x$), but at least it is *consistent*, which means that

$$x^{(k)} = x \Rightarrow x^{(k+1)} = x^{(k)}, \tag{10.9}$$

i.e., if we find the exact solution, the approximation does not change if we keep iterating.

At this point, we can see that:

- For (10.8) to be an efficient iterative method, A_1 should be cheap to invert (or rather cheap to solve a system with, since we might not compute the inverse explicitly).
- If $A_2 = 0$, then (10.8) reduces to $x^{(k+1)} = A_1^{-1}b$, which would yield the exact solution. This is no longer an iterative method, but it suggests that if A_2 is "small", in some sense, then the iterative method is likely to converge.

Several iterative methods utilize that any matrix A can be divided as follows:

$$A = L + D + U, \tag{10.10}$$

where

- L is the strictly lower-triangular part of A,
- D is the diagonal part of A, and
- U is the strictly upper-triangular part of A.

Notice that the diagonal elements are *not* included in L or U.

10.3.1 The Jacobi method

The Jacobi method follows from the general iterative method (10.8) with

$$A_1 = D, \quad A_2 = L + U, \tag{10.11}$$

which yields

$$x^{(k+1)} = -D^{-1}((L+U)x^{(k)} - b). \tag{10.12}$$

Each iteration in the Jacobi method is cheap since D is diagonal and therefore trivial to invert.

10.3.2 The Gauss-Seidel method

The Gauss-Seidel method follows from the general iterative method (10.8) with

$$A_1 = L + D, \quad A_2 = U, \tag{10.13}$$

which yields

$$x^{(k+1)} = -(L+D)^{-1}(Ux^{(k)} - b). \tag{10.14}$$

Notice that $L + D$ is lower triangular, so the action of $(L+D)^{-1}$ is computed efficiently using forward substitution. Still, each iteration with the Gauss-Seidel method is typically significantly more expensive than with the Jacobi method. As we will see, the extra computational cost can sometimes pay off in terms of faster convergence.

10.3.3 Successive over-relaxation (SOR)

The SOR method is an extended version of the Gauss-Seidel method. It follows from the general iterative method (10.8) with

$$A_1 = L + \omega^{-1}D, \quad A_2 = U + \frac{\omega - 1}{\omega}D, \tag{10.15}$$

where the scalar parameter ω is called the *relaxation factor*. The SOR method can be written as (after multiplying both sides of (10.8) by ω)

$$x^{(k+1)} = -(D + \omega L)^{-1}\left[(\omega U + (\omega - 1)D)x^{(k)} - \omega b\right]. \tag{10.16}$$

Notice that:

- with $\omega = 0$, the SOR method (10.16) reduces to the (quite useless) method $x^{(k+1)} = x^{(k)}$.
- with $\omega = 1$, the SOR method (10.16) reduces to the Gauss-Seidel method.
- $D + \omega L$ is lower triangular, so the action of $(D + \omega L)^{-1}$ is computed efficiently using forward substitution.

- The convergence rate of SOR depends on the value of the relaxation factor ω. With a good choice of ω, SOR can converge significantly faster than Gauss-Seidel. However, the optimal value of ω depends on the matrix A and may be difficult to determine. Usually, one sets $1 \leq \omega < 2$. However, if A is symmetric and positive definite ($A = A^T > 0$), one can prove that SOR converges for $0 < \omega < 2$.

10.4 Norms and matrix properties

This section introduces some definitions that we will need to analyze the convergence of iterative methods.

10.4.1 Vector norms

Let x be a vector in $\mathbb{R}^m$. For $p > 0$, the p-norm of x is

$$\|x\|_p := \left(\sum_{i=1}^{m} |x_i|^p \right)^{\frac{1}{p}}. \tag{10.17}$$

Commonly used values for p include:

- ("Manhattan norm"), $p = 1$: $\|x\|_1 = |x_1| + |x_2| + \cdots + |x_m|$
- (Euclidean norm), $p = 2$: $\|x\|_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_m^2}$
- (Maximum norm), $p = \infty$: $\|x\|_\infty = \max_i |x|_i$

10.4.2 Matrix norms

The p-norm for vectors $x \in \mathbb{R}^m$ induces a p-norm for matrices $A \in \mathbb{R}^{n \times m}$. By definition,

$$\|A\|_p := \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}. \tag{10.18}$$

It follows immediately from the definition that

$$\|Ax\|_p \leq \|A\|_p \|x\|_p. \tag{10.19}$$

For $p = 1$ and $p = \infty$, computing the matrix norm is straightforward. One can prove that

$$\|A\|_1 = \max_j \sum_{i=1}^{n} |A_{ij}| \quad \text{(largest column sum)}, \tag{10.20}$$

$$\|A\|_\infty = \max_i \sum_{j=1}^{m} |A_{ij}| \quad \text{(largest row sum)}, \tag{10.21}$$

10.4.3 Spectral radius of a matrix

If A is an $m \times m$ matrix with eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$, then the *spectral radius* of A, denoted $\rho(A)$, is defined as the largest-magnitude eigenvalue of A:

$$\rho(A) = \max_i |\lambda_i|. \tag{10.22}$$

10.4.4 Diagonally dominant matrix

Definition 10. The matrix A is *strictly diagonally dominant* (SDD) if

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}|, \text{ for all } i.$$

10.4.5 Positive definite matrix

Definition 11. The matrix $A \in \mathbb{R}^{m \times m}$ is *symmetric positive definite* (SPD) if $A = A^T$ and

$$x^T A x > 0 \quad \forall x \in \mathbb{R}^m \setminus \{\vec{0}\}.$$

It is useful to remember that a symmetric matrix is SPD if and only if all its eigenvalues are strictly positive.

10.5 Convergence of iterative methods

The general iterative method (10.8) can be written as

$$x^{(k+1)} = Rx^{(k)} + c, \tag{10.23}$$

where

$$R = -A_1^{-1} A_2, \quad c = A_1^{-1} b. \tag{10.24}$$

The matrix R is called the *iteration matrix*. We will now determine under which circumstances (10.23) converges. Let x denote the true solution to the linear system, i.e., x satisfies $Ax = b$. We assume that the iterative method is *consistent* such that

$$x = Rx + c. \tag{10.25}$$

We then have

$$x^{(k)} - x = \underbrace{Rx^{(k-1)} + c}_{x^{(k)}} - \underbrace{(Rx + c)}_{x} = R(x^{(k-1)} - x). \tag{10.26}$$

Repeating the trick above leads to

$$x^{(k)} - x = R(x^{(k-1)} - x) = R^2(x^{(k-2)} - x) = \cdots = R^k(x^{(0)} - x). \tag{10.27}$$

Taking the norm of both sides shows that

$$\|x^{(k)} - x\|_p \leq \|R\|_p^k \|(x^{(0)} - x)\|_p. \tag{10.28}$$

We say that the method converges in the p-norm if

$$\lim_{k\to\infty} \|x^{(k)} - x\|_p = 0. \tag{10.29}$$

It follows from (10.28) that the method converges in the p-norm if $\|R\|_p < 1$. Since the p-norms are equivalent, convergence in one norm implies convergence in all others. To prove convergence, it is therefore sufficient to find one p such that $\|R\|_p < 1$. In other words, the condition

$$\exists p \text{ such that } \|R\|_p < 1, \tag{10.30}$$

is a *sufficient* condition for convergence.

We can use the following well-known result to state a necessary and sufficient condition for convergence:

Theorem 2. *Let R be square matrix. Then,*

$$\lim_{k\to\infty} R^k = 0 \Leftrightarrow \rho(R) < 1.$$

It follows from Theorem 2 that (10.23) converges for any possible starting guess $x^{(0)}$ if and only if $\rho(R) < 1$.

Given a matrix A, we ideally want to select an iterative method such that:

- $\rho(R)$ is as small as possible (and definitely < 1). This gives faster convergence (fewer iterations required to reach the error tolerance).
- the action of $R = -A_1^{-1}A_2$ is cheap to compute. This makes each iteration cheap.

One can prove the following useful *sufficient* (but not necessary) convergence criteria:

Jacobi: Guaranteed to converge if A is SDD.

Gauss-Seidel: Guaranteed to converge if A is SDD *or* SPD.

SOR: Guaranteed to converge if A is SPD (assuming $0 < \omega < 2$).

10.6 Steepest descent and conjugate gradient

Consider a linear system

$$Ax = b, \tag{10.31}$$

where A is SPD. The solution, x_*, is also the unique minimizer of the following function:

$$f(x) = \frac{1}{2}x^T Ax - x^T b. \tag{10.32}$$

To prove this statement, let us use the convention to sum over repeated indices (Einstein notation) so that we may write

$$f(x) = \frac{1}{2}x_i A_{ij} x_j - x_i b_i. \tag{10.33}$$

Let δ_{ij} denote the Kronecker delta:

$$\delta_{ij} = \left\{ \begin{array}{ll} 1, & i = j \\ 0, & i \neq j \end{array} \right. . \tag{10.34}$$

The derivative of f with respect to x_k is

$$\begin{aligned} \frac{\partial f}{\partial x_k} &= \frac{1}{2}\delta_{ik}A_{ij}x_j + \frac{1}{2}x_iA_{ij}\delta_{jk} - \delta_{ik}b_i \\ &= \frac{1}{2}A_{kj}x_j + \frac{1}{2}\underbrace{x_iA_{ik}}_{=x_iA_{ki}=A_{kj}x_j} - b_k \\ &= \underbrace{A_{kj}x_j - b_k}_{=k\text{:th row of Ax-b}} . \end{aligned} \tag{10.35}$$

It follows that the gradient of f is

$$\nabla_x f = \begin{bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_m} \end{bmatrix} = Ax - b. \tag{10.36}$$

Thus, the gradient of f is zero at (and only at) $x = x_* = A^{-1}b$, which means that x_* is the only stationary point of f. By definition, the Hessian matrix H of f is

$$H_{ij} := \frac{\partial^2 f}{\partial x_i \partial x_j} = A_{ij}, \tag{10.37}$$

which shows that $H = A$. Since H is SPD, the stationary point $x = x_*$ is a global minimum.

The point of the above discussion is that solving $Ax = b$ is the same as minimizing $f(x)$. We can therefore develop iterative methods that minimize $f(x)$, and use those methods to solve $Ax = b$.

10.6.1 The steepest descent method

Let $x^{(k)}$ denote the kth approximation of x_*. The function f decreases fastest in the direction of $-\nabla_x f(x^{(k)})$, see Figure 10.1. We therefore want to step in that direction, so we select the *search direction*

$$p^{(k)} = -\nabla_x f(x^{(k)}), \tag{10.38}$$

and set

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \tag{10.39}$$

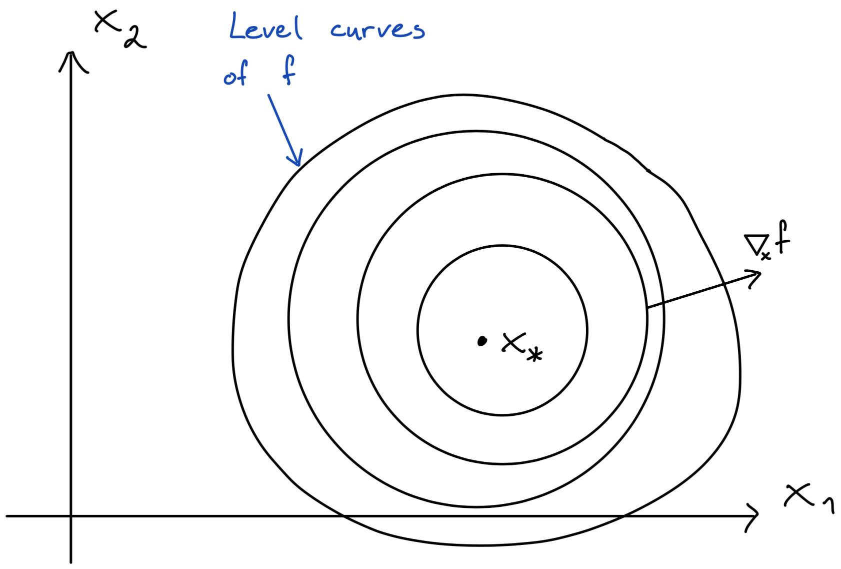


Figure 10.1: Level curves of a function $f(x)$, where $x = [x_1, x_2]^T \in \mathbb{R}^2$.

where α_k is a scalar, called the *step length*. We seek α_k that minimizes f along the search direction. We have

$$f\left(x^{(k)} + \alpha_k p^{(k)}\right) = \frac{1}{2}\left(x^{(k)} + \alpha_k p^{(k)}\right)^T A\left(x^{(k)} + \alpha_k p^{(k)}\right) - \left(x^{(k)} + \alpha_k p^{(k)}\right)^T b. \tag{10.40}$$

To determine the optimal value of α_k, we compute the derivative with respect to α_k:

$$\begin{aligned}\frac{\partial}{\partial \alpha_k} f\left(x^{(k)} + \alpha_k p^{(k)}\right) &= \frac{1}{2}\left(p^{(k)}\right)^T A\left(x^{(k)} + \alpha_k p^{(k)}\right) + \frac{1}{2}\left(x^{(k)} + \alpha_k p^{(k)}\right)^T A p^{(k)} - (p^{(k)})^T b \\ &= \left(p^{(k)}\right)^T A x^{(k)} + \alpha_k \left(p^{(k)}\right)^T A p^{(k)} - (p^{(k)})^T b \\ &= \left(p^{(k)}\right)^T \underbrace{\left(A x^{(k)} - b\right)}_{=-r^{(k)}} + \alpha_k \left(p^{(k)}\right)^T A p^{(k)} \\ &= -\left(p^{(k)}\right)^T r^k + \alpha_k \left(p^{(k)}\right)^T A p^{(k)},\end{aligned}$$

where the *residual* $r^{(k)}$ is defined as

$$r^{(k)} = b - A x^{(k)}. \tag{10.41}$$

We find the optimal value of α_k by requiring the derivative with respect to α_k to be zero:

$$\frac{\partial}{\partial \alpha_k} f\left(x^{(k)} + \alpha_k p^{(k)}\right) = 0 \Longleftrightarrow \alpha_k = \frac{\left(p^{(k)}\right)^T r^{(k)}}{\left(p^{(k)}\right)^T A p^{(k)}}. \tag{10.42}$$

Let us use the standard inner product and norm (the 2-norm) for vectors in $\mathbb{R}^m$:

$$(x, y) := x^T y, \quad \|x\|^2 = (x, x). \tag{10.43}$$

Since A is SPD, we can define the following inner product and corresponding norm:

$$(x, y)_A := x^T A y, \quad \|x\|_A^2 = (x, x)_A\,. \tag{10.44}$$

We have derived the following expressions:

- **Search direction**: $p^{(k)} = -\nabla_x f(x^{(k)}) = -(Ax^{(k)} - b) = r^{(k)}$
- **Step length:** $\alpha_k = \dfrac{\left(p^{(k)}, r^{(k)}\right)}{\|p^{(k)}\|_A^2} = \dfrac{\|r^{(k)}\|^2}{\|r^{(k)}\|_A^2}$
- **New approximation:** $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)} = x^{(k)} + \alpha_k r^{(k)}$

The complete steepest descent algorithm is provided below.

The steepest descent algorithm

Given an initial guess $x^{(0)}$ and an error tolerance *tol*, set $k = 0$ and repeat the following steps until $\|x^{(k+1)} - x^{(k)}\| \leq tol$:

1. Compute the residual (= the search direction): $r^{(k)} = b - Ax^{(k)}$.
2. Compute the step length: $\alpha_k = \dfrac{\|r^{(k)}\|^2}{\|r^{(k)}\|_A^2}$.
3. Compute the next approximation: $x^{(k+1)} = x^{(k)} + \alpha_k r^{(k)}$.
4. Update the counter: $k = k + 1$.

The steepest descent method often "zigzags" its way towards x_* and therefore converges slowly. The *conjugate gradient* method often converges much faster.

10.6.2 The conjugate gradient method

The Conjugate Gradient (CG) method is extremely popular; it is the goto iterative method for linear systems with SPD system matrix A. CG is similar to the steepest descent method. The main difference is that the search directions $p^{(k)}$ are required to be orthogonal (conjugate) in $(\cdot, \cdot)_A$. That is, we require

$$\left(p^{(j)}, p^{(k)}\right)_A = 0, \quad j \neq k. \tag{10.45}$$

This is ensured by setting

$$p^{(k)} = \underbrace{r^{(k)}}_{\text{steepest descent}} \underbrace{- \sum_{i<k} \frac{\left(p^{(i)}, r^{(k)}\right)_A}{\|p^{(i)}\|_A^2} p^{(i)}}_{\text{Gram-Schmidt orthogonalization}}\,. \tag{10.46}$$

Let us verify that $p^{(k)}$ becomes orthogonal to $p^{(0)}, p^{(1)}, \dots, p^{(k-1)}$, assuming that $p^{(0)}, p^{(1)}, \dots, p^{(k-1)}$ are mutually orthogonal. We have, for $j < k$,

$$\begin{aligned}\big(p^{(j)}, p^{(k)}\big)_A &= \big(p^{(j)}, r^{(k)}\big)_A - \sum_{i<k} \frac{\big(p^{(i)}, r^{(k)}\big)_A}{\|p^{(i)}\|_A^2} \underbrace{\big(p^{(j)}, p^{(i)}\big)_A}_{=0, i\neq j} \\ &= \big(p^{(j)}, r^{(k)}\big)_A - \frac{\big(p^{(j)}, r^{(k)}\big)_A}{\|p^{(j)}\|_A^2}\|p^{(j)}\|_A^2 = 0,\end{aligned} \tag{10.47}$$

which proves that $p^{(k)}$ is orthogonal to all previous search directions.

Just as for steepest descent, we update $x^{(k)}$ by stepping in the search direction:

$$x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}, \tag{10.48}$$

and the optimal step length is again given by

$$\alpha_k = \frac{\big(p^{(k)}, r^{(k)}\big)}{\|p^{(k)}\|_A^2}. \tag{10.49}$$

We are now ready to state a basic (not very efficient) version of the CG algorithm.

(A naive) CG algorithm

Given an initial guess $x^{(0)}$ and an error tolerance *tol*, set $k = 0$ and repeat the following steps until $\|x^{(k+1)} - x^{(k)}\| \leq tol$:

1. Compute the residual: $r^{(k)} = b - Ax^{(k)}$.
2. Compute the search direction: $p^{(k)} = r^{(k)} - \sum_{i<k} \frac{\big(p^{(i)}, r^{(k)}\big)_A}{\|p^{(i)}\|_A^2} p^{(i)}$.
3. Compute the step length: $\alpha_k = \frac{\big(p^{(k)}, r^{(k)}\big)}{\|p^{(k)}\|_A^2}$.
4. Compute the next approximation: $x^{(k+1)} = x^{(k)} + \alpha_k p^{(k)}$.
5. Update the counter: $k = k + 1$.

We note that the algorithm can be made much more efficient by utilizing that $\big(p^{(i)}, r^{(k)}\big)_A = 0$ for $i < k - 1$, but proving this is out of the scope of this course.

10.6.3 Convergence properties of the conjugate gradient method

The error after k iterations, denoted by $e^{(k)}$, satisfies

$$\begin{aligned}e^{(k)} &= x^{(k)} - x_* = x^{(k-1)} + \alpha_{k-1}p^{(k-1)} - x_* = e^{(k-1)} + \alpha_{k-1}p^{(k-1)} = \dots \\ &= e^{(0)} + \sum_{i<k} \alpha_i p^{(i)}.\end{aligned} \tag{10.50}$$

Since the search directions $p^{(0)}, p^{(1)}, \ldots, p^{(k)}$ are A-orthogonal, they are linearly independent and form a basis in the $(k+1)$-dimensional space

$$V_k = \text{span}\left(p^{(0)}, p^{(1)}, \ldots, p^{(k)}\right), \quad \text{with } \dim(V_k) = k+1. \tag{10.51}$$

Note that $V_{m-1} = \mathbb{R}^m$, so any vector $v \in \mathbb{R}^m$ is a linear combination of $p^{(0)}, p^{(1)}, \ldots, p^{(m-1)}$. Hence, the initial error $e^{(0)}$ can be expressed as

$$e^{(0)} = \sum_{j=0}^{m-1} \varepsilon_j p^{(j)}, \tag{10.52}$$

for some scalar coefficients ε_j. Since the j:th iteration steps in the direction of $p^{(j)}$, the optimal value of α_j should be $\alpha_j = -\varepsilon_j$, since this would zero out the j:th component of the error! To show that α_j is indeed selected as $-\varepsilon_j$, we first note that(10.50) and (10.52) imply that

$$\left(p^{(k)}, e^{(k)}\right)_A = \left(p^{(k)}, e^{(0)}\right)_A + \sum_{i<k} \alpha_i \underbrace{\left(p^{(k)}, p^{(i)}\right)_A}_{=0} = \sum_{j=0}^{m-1} \varepsilon_j \left(p^{(k)}, p^{(j)}\right)_A = \varepsilon_k \|p^{(k)}\|_A^2, \tag{10.53}$$

which means that

$$\varepsilon_k = \frac{\left(p^{(k)}, e^{(k)}\right)_A}{\|p^{(k)}\|_A^2}. \tag{10.54}$$

To derive a similar expression for α_k, note that

$$r^{(k)} = b - Ax^{(k)} = Ax_* - Ax^{(k)} = -Ae^{(k)}, \tag{10.55}$$

and hence

$$\alpha_k = \frac{\left(p^{(k)}, r^{(k)}\right)}{\|p^{(k)}\|_A^2} = -\frac{(p^{(k)})^T Ae^{(k)}}{\|p^{(k)}\|_A^2} = -\varepsilon_k. \tag{10.56}$$

This means that in iteration k, the kth component of the error is zeroed out. We have

$$e^{(k)} = e^{(0)} + \sum_{i<k} \alpha_i p^{(i)} = \sum_{j=0}^{m-1} \varepsilon_j p^{(j)} + \sum_{i<k} (-\varepsilon_i) p^{(i)} = \sum_{j=k}^{m-1} \varepsilon_j p^{(j)}. \tag{10.57}$$

After m iterations, the error vanishes, which means that (in perfect arithmetic) **CG is guaranteed to converge to the exact solution in at most m iterations.** The convergence of CG can also be described by the following Theorem.

Theorem 3. *Let $\kappa(A) = \|A\|\|A^{-1}\|$ denote the condition number of A. The CG error after k iterations satisfies*

$$\|e^{(k)}\|_A \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^k \|e^{(0)}\|_A. \tag{10.58}$$

We end this section by summarizing some important properties of CG.

- CG is **only** applicable if A is SPD.
- In perfect arithmetic, CG finds the *exact* solution after at most m iterations. In practice, roundoff errors compromise the orthogonality of the search directions $p^{(k)}$.
- CG converges faster if the condition number $\kappa(A)$ is small, see Theorem 3. Note, however, that Theorem 3 only provides an *upper* bound on the error. In practice, CG often converges faster than the bound.

References

[1] M. H. Carpenter, D. Gottlieb, and S. Abarbanel. Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: Methodology and application to high-order compact schemes. *J. Comput. Phys.*, 111(2):220–236, 1994.

[2] B. Gustafsson. *High Order Difference Methods for Time Dependent PDE.* Springer, 2007. doi:10.1007/978-3-540-74993-6.

[3] H.-O. Kreiss and G. Scherer. Finite element and finite difference methods for hyperbolic partial differential equations. *Mathematical Aspects of Finite Elements in Partial Differential Equations., Academic Press, Inc.*, pages 195–212, 1974. doi:10.1016/B978-0-12-208350-1.50012-1.

[4] M. G. Larson and F. Bengzon. *The Finite Element Method: Theory, Implementation, and Applications.* Springer Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-33287-6.

[5] K. Mattsson and P. Olsson. An improved projection method. *Journal of Computational Physics*, 372:349 – 372, 2018. doi:https://doi.org/10.1016/j.jcp.2018.06.030.

[6] P. Olsson. Summation by parts, projections, and stability I. *Math. Comp.*, 64:1035, 1995.

[7] P. Olsson. Summation by parts, projections, and stability II. *Math. Comp.*, 64:1473, 1995.

[8] G. Scherer. *On the existence of energy estimates for difference approximations for hyperbolic systems.* Ph.D. thesis, Dept. of Scientific Computing, Uppsala University, 1977.