Université de Ain Témouchent — Belhadj Bouchaib
Faculté des Sciences et de la Technologie
Département de Mathématiques et Informatiques

Polycopié pédagogique

Titre

Numerical Analysis of partial Differential Equations: Methods and Applications

Cours destiné aux étudiants de

M1 Biomathématiques

Rédigé par :

M. Bentout Soufiane

Année universitaire : 2025-2026

Contents

Introduction

The study of *partial differential equations (PDEs)* is central to the mathematical modeling of natural and engineered systems. PDEs arise in diverse contexts such as heat conduction, fluid dynamics, wave propagation, and population dynamics. Since analytical solutions are available only for a limited class of PDEs, the development of reliable *numerical methods* is essential for approximating their solutions (see, e.g., [4, 2, 5, 3, 1]).

This handout is devoted to an introduction to the numerical treatment of PDEs. We begin with a *classification of PDEs* into elliptic, parabolic, and hyperbolic types, highlighting their mathematical properties and physical interpretations.

The first numerical approach we study is the *finite difference method (FDM)*. In particular, we apply it to one-dimensional elliptic problems, parabolic problems, and the *heat equation*. Special emphasis will be placed on the *Crank–Nicolson method*, which offers a balance between stability and accuracy for time-dependent parabolic equations [5].

In the second part, we introduce the *finite volume method (FVM)*, which is especially suited for conservation laws. Applications will cover both *elliptic* and *hyperbolic* problems, illustrating the method's flexibility in different contexts [3].

The last numerical approach is the *finite element method (FEM)*. We present it through its application to the *heat equation*, focusing on how FEM handles geometrical complexity and boundary conditions [1].

Finally, in the *numerical simulations* section, we implement the studied methods in MATLAB. Several computational experiments will be carried out, and graphical results will be provided to illustrate the effectiveness of the methods.

Organization of the Handout

- **Chapter 1.** Classification of PDEs (elliptic, parabolic, and hyperbolic).
- **Chapter 2.** Finite Difference Method: applications to one-dimensional elliptic problems, parabolic problems, and the heat equation.
- **Chapter3.** Implicit method Crank Nicholson
- **Chapter 4.** Finite Volume Method: applications to elliptic and hyperbolic problems.
- **Chapter 5.** Finite Element Method: application to the heat equation.
- **Chapter 6.** Numerical Simulations in MATLAB with graphical illustrations.

Chapter 1

Classification the Partial differential equations

1.1 First-order partial differential equation

In this section, we study a first-order linear partial differential equation of the following form:

$$A\frac{\partial u}{\partial x}+B\frac{\partial u}{\partial y}+Cu=D, \tag{2.1}$$

where A, B, C, and D are continuously differentiable functions on a domain $D\subset\mathbb{R}^2$.

Definition 1.1.1. *We define the parametric curve γ as the image of a function $\gamma: I\to D$, with $\gamma(\tau)=(x(\tau),y(\tau))$, for all $\tau\in I$.*

Remark 1.1.1. *We can view γ as a parameterization of a curve C, and the vector $(x'(\tau),y'(\tau))$ is called the tangent vector to the curve C.*

Example 1.1.1. *Consider a part of a circle as follows: $\gamma:\left[0,\frac{\pi}{2}\right]\to\mathbb{R}^2$. We define the parameterization:*

$$\gamma(\tau)=(\cos(\tau),\sin(\tau)).$$

The tangent vector at $\tau=\frac{\pi}{4}$ is

$$\gamma'\left(\frac{\pi}{4}\right)=\left(-\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}\right),$$

and this is the tangent vector at the point

$$\gamma\left(\frac{\pi}{4}\right)=\left(\frac{\sqrt{2}}{2},\frac{\sqrt{2}}{2}\right).$$

Definition 1.1.2. *The directional derivative of a function $f(x,y)$ on a domain D in the direction of a vector $\mathbf{d}=(d_1,d_2)$ is defined by:*

$$f'\left((x,y);\mathbf{d}\right)=\lim_{t\to 0}\frac{f(x+td_1,y+td_2)-f(x,y)}{t}.$$

We can rewrite this formula as follows:

$$f'((x,y);\mathbf{d}) = \left.\frac{\partial f}{\partial x}\right|_{(x,y)} d_1 + \left.\frac{\partial f}{\partial y}\right|_{(x,y)} d_2 = \nabla f\big|_{(x,y)} \cdot \mathbf{d},$$

where $\nabla f\big|_{(x,y)}$ *is the gradient of* f *at the point* (x,y)*, given by:*

$$\nabla f\big|_{(x,y)} = \left(\left.\frac{\partial f}{\partial x}\right|_{(x,y)}, \left.\frac{\partial f}{\partial y}\right|_{(x,y)} \right).$$

Definition 1.1.3. *The method of characteristics seeks curves (called "characteristic lines") along which the partial differential equation reduces to a simple ordinary differential equation. Solving the ODE along a characteristic allows us to recover the solution to the original problem.*

Let us return to our equation (2.1). Consider the following hypothesis: $(A(x,y),B(x,y)) \neq (0,0)$. We consider the following parametric curves $\sigma : \mathbb{R} \to \mathbb{R}^2$ defined by $s \mapsto (x(s),y(s))$, with the tangent vector $\sigma'(s) = (x'(s),y'(s))$ at the point $\sigma(s) = (x(s),y(s))$.

Thus, we set:

$$\frac{dx}{ds} = A(x,y) \quad \text{and} \quad \frac{dy}{ds} = B(x,y).$$

Let $u(s) = u(x(s),y(s))$ be the solution along these parametric curves characterized by σ. By using the chain rule, we have:

$$\frac{du}{ds} = \frac{\partial u}{\partial x}\frac{dx}{ds} + \frac{\partial u}{\partial y}\frac{dy}{ds} = A(x(s),y(s)) \left.\frac{\partial u}{\partial x}\right|_{(x(s),y(s))} + B(x(s),y(s)) \left.\frac{\partial u}{\partial y}\right|_{(x(s),y(s))}.$$

This simplifies to:

$$\frac{du}{ds} = -C(x(s),y(s))u(x(s),y(s)) + D(x(s),y(s)),$$

and thus, the solution is given by:

$$u(s) = u(0) + \int_0^s (-C(x(\tau),y(\tau))u(x(\tau),y(\tau)) + D(x(\tau),y(\tau)))\, d\tau.$$

The curve of initial values is:

$$\tau \mapsto (X(\tau),Y(\tau),u(X(\tau),Y(\tau))).$$

For each $\tau \in I$, we obtain a characteristic curve:

$$s \mapsto (x(s,\tau),y(s,\tau),u(s,\tau)),$$

with $(x(0,\tau),y(0,\tau)) = (X(\tau),Y(\tau),F(\tau))$ for all $\tau \in I$.

Assume the following condition holds:

$$J(x,y) = \begin{vmatrix} \frac{\partial x}{\partial s} & \frac{\partial x}{\partial \tau} \\ \frac{\partial y}{\partial s} & \frac{\partial y}{\partial \tau} \end{vmatrix} = \frac{\partial x}{\partial s}\frac{\partial y}{\partial \tau} - \frac{\partial x}{\partial \tau}\frac{\partial y}{\partial s} \neq 0.$$

From this condition, we can express s and τ as functions of x and y. This result follows from the Inverse Function Theorem. By substituting the expressions for s and τ into $u(s,\tau)$, we obtain the solution u as a function of x and y.

1.2 Applications

Example 1.2.1. *Consider the following equation:*

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad u(x,0) = f(x), \quad \forall x \in \mathbb{R}, t \geq 0,$$

where $f(x)$ is an arbitrary function depending on x, and c is a positive constant.

Considering the parametric curves with the following parameterization:

$$\sigma(s) = (x(s), t(s)),$$

and with the tangent vector $\sigma'(s) = (x'(s), t'(s)) = (c, 1)$, we obtain:

$$\frac{dt}{ds} = 1 \quad \textit{and} \quad \frac{dx}{ds} = c.$$

We start by assuming $x(0) = x_0$ and $t(0) = t_0$. By integrating the two differential equations, we get:

$$t(s) = s + t_0 \quad \textit{and} \quad x(s) = cs + x_0, \quad \forall s \in \mathbb{R}.$$

We also consider the solution along these curves $u(s) = u(x(s), t(s))$, and we apply the chain rule:

$$\frac{du}{ds} = \frac{\partial u}{\partial t}\frac{dt}{ds} + \frac{\partial u}{\partial x}\frac{dx}{ds} = \frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0,$$

which can be interpreted as the directional derivative of u in the direction of the tangent vector $(x_0'(s), t_0'(s)) = (c, 1)$, which equals zero. Thus, u is constant along the curves $s \mapsto (x(s), t(s)) = (cs + x_0, s + t_0)$ with $s \in \mathbb{R}$.

We use the initial conditions (x_0, t_0, u_0) to find this constant. Note that these initial values can be parameterized by a curve: $x_0 = \xi, t_0 = 0, u_0 = f(\xi)$ with $\xi \in \mathbb{R}$. By simple calculation, we obtain:

$$x(s, \xi) = cs + \xi, \quad t(s, \xi) = s, \quad u(s, \xi) = f(\xi).$$

Thus, $s = t$ and $\xi = x - cs = x - ct$. Substituting this into the equation for u, and noting that $u(x,t) = u(x_0, 0)$, we deduce:

$$u(x,t) = f(x - ct), \quad \forall x \in \mathbb{R}.$$

Example 1.2.2. *Consider the following problem:*

$$\begin{cases} \frac{\partial u}{\partial y} + x\frac{\partial u}{\partial x} = y^3, \\ u(x,0) = x, \end{cases}$$

for $x \in \mathbb{R}$ and $y \geq 0$.

We consider the characteristic curves:

$$\frac{dy}{ds} = 1 \quad \textit{and} \quad \frac{dx}{ds} = x.$$

Let $\sigma(s) = (y(s), x(s))$ be the parameterization with $\sigma(0) = (y(0) = y_0, x(0) = x_0)$. By integrating the two differential equations, we obtain:

$$y(s) = s + y_0, \quad x(s) = x_0 e^s, \quad \forall s \in \mathbb{R}.$$

Let $u(s) = u(x(s), y(s))$ be the solution along the characteristic curves. Taking the derivative with respect to s, we get:

$$\frac{du}{ds} = \frac{\partial u}{\partial y}\frac{dy}{ds} + \frac{\partial u}{\partial x}\frac{dx}{ds} = \frac{\partial u}{\partial y} + x\frac{\partial u}{\partial x} = y^3 = (s + y_0)^3.$$

By integrating this, we find:

$$u(s) = \frac{1}{4}(s + y_0)^4 + u_0.$$

Now, we use the initial conditions (x_0, y_0, u_0), parameterized by the following curve $x_0 \to \xi, y_0 \to 0, u_0 \to \xi$, giving:

$$x(s, \xi) \to \xi e^s, \quad y(s, \xi) \to s, \quad u(s, \xi) \to \frac{1}{4}s^4 + \xi.$$

Thus, $s \to y$ and $\xi \to xe^{-y}$. Substituting this into the equation for u, we conclude:

$$u(x, y) = \frac{1}{4}y^4 + xe^{-y}.$$

Example 1.2.3. *Consider the following problem:*

$$\begin{cases} \frac{\partial u}{\partial y} + \frac{\partial u}{\partial x} + u = e^{-y+2x}, \\ u(x, 0) = x^2, \end{cases}$$

for $x \in \mathbb{R}$ and $y \geq 0$.
We construct the following characteristic curves:

$$\frac{dx}{ds} = 1, \quad \frac{dy}{ds} = 1.$$

By integrating these differential equations with the initial conditions $x(0) = x_0$ and $y(0) = y_0$, we obtain:

$$x(s) = x_0 + s, \quad y(s) = y_0 + s.$$

Next, we compute the derivative $\frac{du}{ds}$ using the chain rule:

$$\frac{du}{ds} = \frac{\partial u}{\partial x}\frac{dx}{ds} + \frac{\partial u}{\partial y}\frac{dy}{ds} = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = -u + e^{-y+2x} = -u + e^{-y_0+2x_0+s},$$

and therefore,

$$\frac{du}{ds} = -u + e^{-y_0+2x_0+s}. \tag{2.3}$$

We solve this differential equation using the method of variation of constants. The homogeneous equation

$$\frac{du}{ds} + u = 0$$

has the solution $u(s) = u(0)e^{-s}$.

We observe that $v(s) = \frac{1}{2}e^{s+2x_0-y_0}$ *is a particular solution of equation (2.3). Thus, the general solution is:*

$$u(s) = u(0)e^{-s} + \frac{1}{2}e^{s+2x_0-y_0}.$$

We use the initial conditions, parameterized as follows:

$$x_0 = \xi, \quad y_0 = 0, \quad u_0 = \xi^2.$$

Substituting into the characteristic curves, we obtain:

$$x(s,\xi) = s + \xi, \quad y(s,\xi) = s,$$

and therefore,

$$s = y, \quad \xi = x - y.$$

By substituting into the equation for u, we deduce:

$$u(x,y) = (x-y)^2 e^{-y} + \frac{1}{2}e^{2(x-y)+y}.$$

Now we present exercise.

Example 1.2.4. *Consider the following problem:*

$$\frac{\partial u}{\partial y} + \frac{\partial u}{\partial x} + u = e^{-y+2x},$$

with the initial condition $u(x,0) = x^2$*, for* $x \in \mathbb{R}$ *and* $y \geq 0$*.*

We construct the characteristic curves:

$$\frac{dx}{ds} = 1, \quad \frac{dy}{ds} = 1.$$

By integrating these differential equations with initial conditions $x(0) = x_0$ *and* $y(0) = y_0$*, we get:*

$$x(s) = x_0 + s, \quad y(s) = y_0 + s.$$

We compute the derivative $\frac{du}{ds}$ *using the chain rule:*

$$\frac{du}{ds} = \frac{\partial u}{\partial x} + \frac{\partial u}{\partial y} = -u + e^{-y_0+2x_0+s}.$$

Thus, we have the equation:

$$\frac{du}{ds} = -u + e^{-y_0+2x_0+s}. \tag{2.3}$$

We solve this using the method of variation of constants. The solution of the homogeneous equation $\frac{du}{ds} + u = 0$ *is:*

$$u(s) = u(0)e^{-s}.$$

We find a particular solution as $v(s) = \frac{1}{2}e^{s+2x_0-y_0}$*, giving the general solution:*

$$u(s) = u(0)e^{-s} + \frac{1}{2}e^{s+2x_0-y_0}.$$

Using the initial conditions $x_0 = \xi, y_0 = 0, u_0 = \xi^2$, we obtain:

$$x(s,\xi) = s + \xi, \quad y(s,\xi) = s.$$

Thus, $s = y$ and $\xi = x - y$. Substituting this into the equation for u, we deduce:

$$u(x,y) = (x-y)^2 e^{-y} + \frac{1}{2}e^{2(x-y)+y}.$$

1.3 Second-order linear partial differential equations

We focus on second-order linear partial differential equations (PDEs).

Definition 1.3.1. *A second-order PDE is defined as follows:*

$$A\frac{\partial^2 u}{\partial x^2} + B\frac{\partial^2 u}{\partial x \partial y} + C\frac{\partial^2 u}{\partial y^2} + D\frac{\partial u}{\partial x} + E\frac{\partial u}{\partial y} + Fu = G. \tag{3.1}$$

We assume that the functions A, B, C, D, E, F, and G are C^1-class functions on a domain $D \subset \mathbb{R}^2$.

We present the following proposition to classify second-order PDEs.

Proposition 1.3.1. *The nature of a second-order PDE depends on the discriminant $\Delta = B^2 - 4AC$:*

1. *If $\Delta < 0$, the PDE is called **elliptic**.*
2. *If $\Delta = 0$, the PDE is called **parabolic.***
3. *If $\Delta > 0$, the PDE is called **hyperbolic.***

We need the following definition.

Definition 1.3.2. *The chain rule is a formula used to differentiate a composite function of two differentiable functions.*

If a function u depends on a function x, which in turn depends on a variable y (assuming that u and x are differentiable), then the derivative in this case is

$$\frac{du}{dy} = \frac{du}{dx} \cdot \frac{dx}{dy}.$$

1.3.1 Elliptic Case

In this case, $\Delta < 0$ over the domain D, leading to two characteristic curves with complex values, and the solutions are conjugates of each other.

The characteristic curves are given by $\eta(x,y)$ and $\xi(x,y) = \overline{\eta(x,y)}$. We use the following change of variables: $\alpha = \frac{\eta+\xi}{2}$ and $\beta = \frac{\eta-\xi}{2i}$.

Example 1.3.1. *Consider the following equation:*

$$\frac{\partial^2 u}{\partial x^2} + x^2 \frac{\partial^2 u}{\partial y^2} = 0, \tag{3.5}$$

which is a second-order linear equation with $\Delta = B^2 - 4AC = -4x^2$.
Thus, this equation is parabolic on the line $x = 0$. If $x \neq 0$, the equation is elliptic.

Furthermore, we find the following two characteristic equations, if $\Delta > 0$:

$$\frac{dy}{dx} = \frac{B + \sqrt{B^2 - 4AC}}{2A}$$

and

$$\frac{dy}{dx} = \frac{B - \sqrt{B^2 - 4AC}}{2A}.$$

1.3.2 Parabolic case

If $\Delta = 0$ on D, the equation is said to be parabolic.

Example 1.3.2. *The most famous example is the heat equation:*

$$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}.$$

We search for $\eta(x,y)$ such that $A = 0$; in this case, we obtain only one characteristic coordinate $\eta(x,y)$, as there is only one characteristic equation. To find the second coordinate $\xi(x,y)$, we take an arbitrary function of at least class C^2 on the domain D, with:

$$\begin{pmatrix} \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \\ \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \end{pmatrix} = \frac{\partial \eta}{\partial x}\frac{\partial \xi}{\partial y} - \frac{\partial \eta}{\partial y}\frac{\partial \xi}{\partial x} = 0.$$

1.3.3 Hyperbolic Case

If $\Delta > 0$ on D, the equation is said to be hyperbolic.
To better understand this, we present some examples.

Example 1.3.3. *The most well-known example of a hyperbolic PDE is the wave equation:*

$$\frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} = 0.$$

Indeed, $\Delta = 1 > 0$. It is possible to transform the wave equation into a first-order system:

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial u}{\partial x} = v, \\ \frac{\partial v}{\partial t} + \frac{\partial v}{\partial x} = 0. \end{cases} \tag{3.7}$$

In fact, consider the following equation:

$$\left(\frac{\partial}{\partial t} + \frac{\partial}{\partial x}\right)\left(\frac{\partial u}{\partial t} - \frac{\partial u}{\partial x}\right) = v = 0,$$

and we recover the system (3.7).

Example 1.3.4. *Consider the following equation:*

$$y^2\frac{\partial^2 u}{\partial x^2} - x^2\frac{\partial^2 u}{\partial y^2} = 0, \quad \forall x, y \neq 0.$$

This is a hyperbolic equation since $\Delta = B^2 - 4AC = 4x^2y^2 > 0$. *We seek the characteristic equations (curves):*

$$\frac{dy}{dx} = \frac{b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{0 \pm \sqrt{4x^2y^2}}{2y^2}.$$

Using the method of separation of variables, we find:

$$\frac{dy}{dx} = \pm\frac{x}{y},$$

thus,

$$c_1 = \frac{y^2 + x^2}{2}, \quad c_2 = \frac{y^2 - x^2}{2}.$$

We define $\eta(x,y) = \frac{y^2-x^2}{2}$ *and* $\xi(x,y) = \frac{y^2+x^2}{2}$, *which are the characteristic curves.*

1.3.4 Exercises

Exercise 01

Determine for which points (x,y) in the plane each of the following second-order linear PDEs is elliptic, parabolic, or hyperbolic.

1. $x\frac{\partial^2 u}{\partial x^2} - xy\frac{\partial^2 u}{\partial x \partial y} + y^2\frac{\partial^2 u}{\partial y^2} - 3\frac{\partial u}{\partial x} = 0.$
2. $x\frac{\partial^2 u}{\partial x^2} + xy\frac{\partial^2 u}{\partial x \partial y} + y\frac{\partial^2 u}{\partial y^2} - (x+3)\frac{\partial u}{\partial x} = 0.$
3. $\frac{\partial^2 u}{\partial x^2} - 5\frac{\partial^2 u}{\partial x \partial y} - (x+y)\frac{\partial^2 u}{\partial y^2} + 4\frac{\partial u}{\partial x} - x\frac{\partial u}{\partial y} = \sin(x).$

Exercise 02

Consider the following PDEs:

$$\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} - 3u = 0,$$

and

$$\frac{\partial^2 u}{\partial x^2} + 4\frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} = 0.$$

1. Determine their types, the characteristic equations, and the characteristic coordinates.
2. Reduce the equations to their canonical form.

Exercise 03

Consider the following PDE:

$$\frac{\partial^2 u}{\partial x^2} + 4\frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} = 0.$$

Determine the type of the equation, its characteristic equations, and characteristic coordinates, and then reduce the equation to its canonical form.

Exercise 4

We define the following PDE:

$$\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial y} - 3u = 0.$$

1. Determine the type of this equation and the characteristic equations associated with it.
2. Reduce this equation to its canonical form.

Supplementary Exercise

Exercise 05

For each of the following second-order linear PDEs:

$$2y^2\frac{\partial^2 u}{\partial x^2} - xy\frac{\partial^2 u}{\partial x \partial y} - x^2\frac{\partial^2 u}{\partial y^2} + 4y\frac{\partial u}{\partial x} - 3u = 0,$$

and

$$x^2\frac{\partial^2 u}{\partial x^2} - xy\frac{\partial^2 u}{\partial x \partial y} - 6y^2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} = 0,$$

1. Determine the points in the plane (x,y) where these equations are hyperbolic. 2. Determine the characteristic coordinates of these equations in the domain where they are parabolic. 3. Perform a coordinate transformation for the coordinates found in question 2 to obtain the corresponding canonical form of the equation.

1.3.5 Correction of Exercises

Exercise 1

1. Consider the following equation:

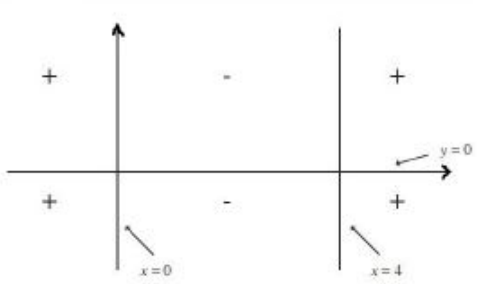
$$x\frac{\partial^2 u}{\partial x^2} - xy\frac{\partial^2 u}{\partial x \partial y} + y^2\frac{\partial^2 u}{\partial y^2} - 3\frac{\partial u}{\partial x} = 0,$$

we compute the discriminant:

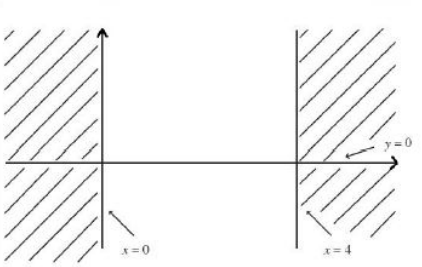
$$\Delta = B^2 - 4AC = (-xy)^2 - 4xy^2 = xy^2(x-4).$$

For $\Delta = 0$, we have $x = 0$, $y = 0$, or $x = 4$. We study the sign of Δ in the regions:

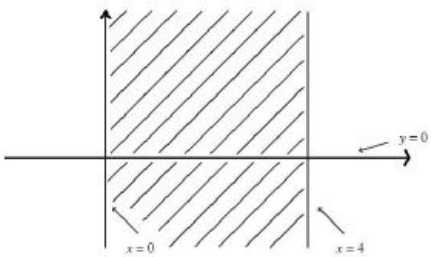
$$\mathbb{R}^2 \setminus \left(\{(x,y) \in \mathbb{R}^2 \mid x = 0\} \cup \{(x,y) \in \mathbb{R}^2 \mid y = 0\} \cup \{(x,y) \in \mathbb{R}^2 \mid x = 4\}\right).$$



Thus, the equation is hyperbolic in the following shaded region:



And the equation is elliptic in the following shaded region:



2. Now consider the equation:

$$x\frac{\partial^2 u}{\partial x^2} + xy\frac{\partial^2 u}{\partial x \partial y} + y\frac{\partial^2 u}{\partial y^2} - (x+3)\frac{\partial u}{\partial x} = 0.$$

We compute the discriminant:

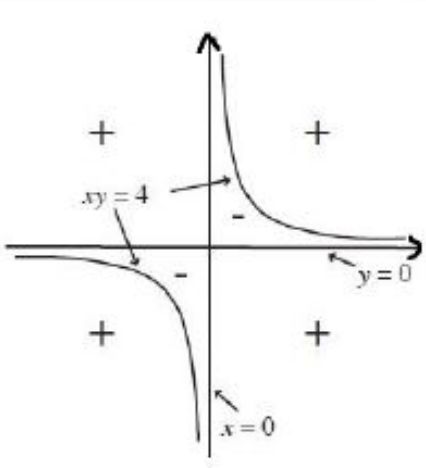
$$\Delta = B^2 - 4AC = (xy)^2 - 4(x)(y) = xy(xy-4).$$

We discuss the possible cases. For $\Delta = 0$, we get $x = 0$, $y = 0$, or $xy = 4$. At these points, the PDE is parabolic.

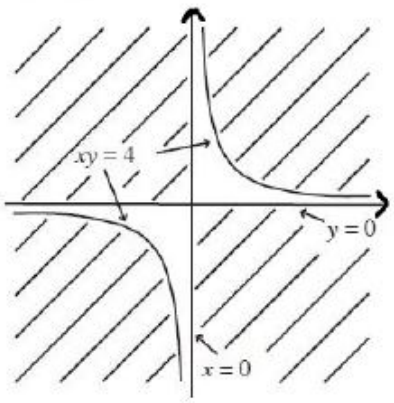
For the regions:

$$\mathbb{R}^2 \setminus \left(\{(x,y) \in \mathbb{R}^2 \mid x = 0\} \cup \{(x,y) \in \mathbb{R}^2 \mid y = 0\} \cup \{(x,y) \in \mathbb{R}^2 \mid xy = 4\} \right),$$

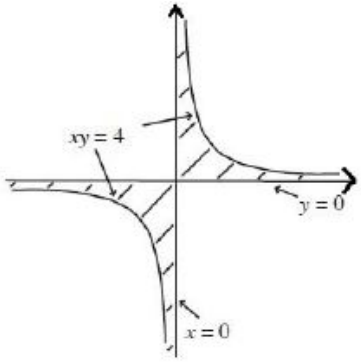
we determine the sign of Δ in the following graph:



Thus, the PDE is hyperbolic in the following shaded region:



And the PDE is elliptic in the following shaded region:



3. Consider the following PDE:

$$\frac{\partial^2 u}{\partial x^2} - 5\frac{\partial^2 u}{\partial x \partial y} - (x+y)\frac{\partial^2 u}{\partial y^2} + 4\frac{\partial u}{\partial x} - x\frac{\partial u}{\partial y} = \sin(x).$$

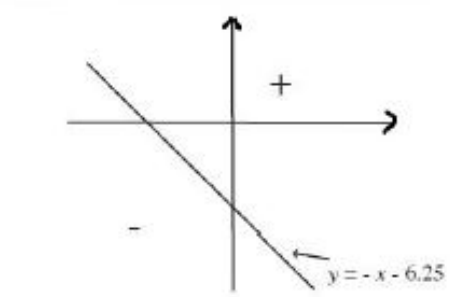
We compute the discriminant:

$$\Delta = B^2 - 4AC = (-5)^2 - 4(1)(-(x+y)) = 25 + 4(x+y),$$

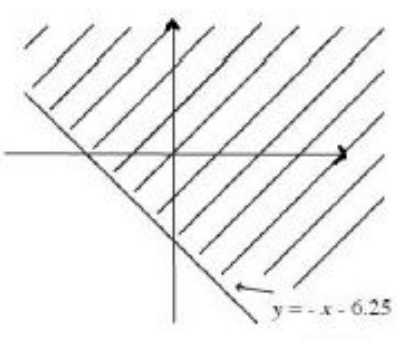
For $\Delta = 0$, we obtain the line $y = -x - 6.25$. For the region:

$$\mathbb{R}^2 \setminus \left\{(x,y) \in \mathbb{R}^2 \mid 4x + 4y + 25 = 0\right\},$$

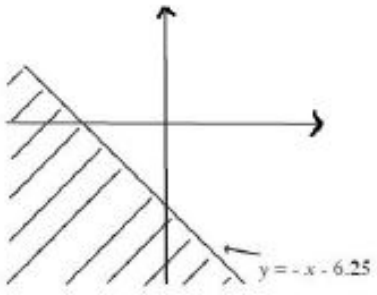
we indicate the sign of Δ in the following graph:



Thus, the PDE is hyperbolic in the following shaded region:



And the PDE is elliptic in the following shaded region:



Exercise 02

Let us consider the following PDE:

$$\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} - 3u = 0. \tag{A.2}$$

We compute the discriminant $\Delta = B^2 - 4AC$:

$$\Delta = (-2)^2 - 4(1)(2) = -4,$$

which implies that this equation is elliptic for all $(x,y) \in \mathbb{R}^2$.

The characteristic equations are:

$$\frac{dy}{dx} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A} = \frac{-2 \pm \sqrt{-4}}{2} = -1 \pm i.$$

Thus, the solutions are:

$$y = (-1+i)x + c_1, \quad y = (-1-i)x + c_2.$$

We set $\xi(x,y) = y - (-1+i)x$ and $\eta(x,y) = y - (-1-i)x$.

Now, considering new coordinates α, β where:

$$\alpha = \frac{\xi(x,y) + \eta(x,y)}{2} = x + y, \quad \beta = \frac{\xi(x,y) - \eta(x,y)}{2i} = -x,$$

and applying the chain rule, we find:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \alpha}\frac{\partial \alpha}{\partial x} + \frac{\partial u}{\partial \beta}\frac{\partial \beta}{\partial x} = \frac{\partial u}{\partial \alpha} - \frac{\partial u}{\partial \beta},$$

$$\frac{\partial u}{\partial y} = \frac{\partial u}{\partial \alpha}\frac{\partial \alpha}{\partial y} + \frac{\partial u}{\partial \beta}\frac{\partial \beta}{\partial y} = \frac{\partial u}{\partial \alpha},$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial \alpha^2} - 2\frac{\partial^2 u}{\partial \alpha \partial \beta} + \frac{\partial^2 u}{\partial \beta^2},$$

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 u}{\partial \alpha^2} - \frac{\partial^2 u}{\partial \alpha \partial \beta},$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{\partial^2 u}{\partial \alpha^2}.$$

Substituting into equation $(A.2)$, we obtain the canonical form:

$$\frac{\partial^2 u}{\partial \alpha^2} + \frac{\partial^2 u}{\partial \beta^2} + \frac{\partial u}{\partial \alpha} - \frac{\partial u}{\partial \beta} - 3u = 0.$$

Exercise 02

Consider the following partial differential equation (PDE):

$$\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial x} - 3u = 0. \tag{A.2}$$

We calculate the discriminant Δ:

$$\Delta = B^2 - 4AC = (-2)^2 - 4(1)(2) = -4,$$

which shows that this equation is elliptic for all $(x,y) \in \mathbb{R}^2$.

The characteristic equations are given by:

$$\frac{dy}{dx} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A} = -2 \pm \frac{\sqrt{-4}}{2} = -1 \pm i.$$

This leads to the solutions:

$$y = (-1+i)x + c_1, \quad y = (-1-i)x + c_2.$$

We define the following coordinates:

$$\xi(x,y) = y - (-1+i)x, \quad \eta(x,y) = y - (-1-i)x.$$

Next, we consider the new coordinates α and β where:

$$\alpha = \frac{\xi(x,y) + \eta(x,y)}{2} = x + y, \quad \beta = \frac{\xi(x,y) - \eta(x,y)}{2i} = -x.$$

Applying the chain rule, we find the partial derivatives:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \alpha}\frac{\partial \alpha}{\partial x} + \frac{\partial u}{\partial \beta}\frac{\partial \beta}{\partial x} = \frac{\partial u}{\partial \alpha} - \frac{\partial u}{\partial \beta},$$

$$\frac{\partial u}{\partial y} = \frac{\partial u}{\partial \alpha}\frac{\partial \alpha}{\partial y} + \frac{\partial u}{\partial \beta}\frac{\partial \beta}{\partial y} = \frac{\partial u}{\partial \alpha}.$$

The second derivatives are:

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial \alpha^2} - 2\frac{\partial^2 u}{\partial \alpha \partial \beta} + \frac{\partial^2 u}{\partial \beta^2},$$

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 u}{\partial \alpha^2} - \frac{\partial^2 u}{\partial \alpha \partial \beta},$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{\partial^2 u}{\partial \alpha^2}.$$

Substituting these into the PDE $(A.2)$, we arrive at the canonical form:

$$\frac{\partial^2 u}{\partial \alpha^2} + \frac{\partial^2 u}{\partial \beta^2} + \frac{\partial u}{\partial \alpha} - \frac{\partial u}{\partial \beta} - 3u = 0.$$

Exercise 03

Let us consider the following PDE:

$$\frac{\partial^2 u}{\partial x^2} - 2\frac{\partial^2 u}{\partial x \partial y} + 2\frac{\partial^2 u}{\partial y^2} + \frac{\partial u}{\partial y} - 3u = 0. \tag{A.4}$$

We compute the discriminant $\Delta = B^2 - 4AC$:

$$\Delta = (-2)^2 - 4(1)(2) = -4,$$

thus, the equation is elliptic for all $(x,y) \in \mathbb{R}^2$.

The characteristic curves are:

$$\frac{dy}{dx} = \frac{B \pm \sqrt{B^2 - 4AC}}{2A} = -1 \pm i.$$

By solving:

$$\int dy = \int (-1+i)dx \quad \Rightarrow \quad y = (-1+i)x + c_1,$$

$$\int dy = \int (-1-i)dx \quad \Rightarrow \quad y = (-1-i)x + c_2.$$

We note that the characteristic coordinates $\xi(x,y) = y - (-1+i)x$ and $\eta(x,y) = y - (-1-i)x$ are not real. Hence, we introduce real coordinates $\alpha = y + x$ and $\beta = -x$.

Applying the chain rule to compute partial derivatives:

$$\frac{\partial u}{\partial x} = \frac{\partial u}{\partial \alpha} - \frac{\partial u}{\partial \beta},$$

$$\frac{\partial u}{\partial y} = \frac{\partial u}{\partial \alpha},$$

$$\frac{\partial^2 u}{\partial x^2} = \frac{\partial^2 u}{\partial \alpha^2} - 2\frac{\partial^2 u}{\partial \alpha \partial \beta} + \frac{\partial^2 u}{\partial \beta^2},$$

$$\frac{\partial^2 u}{\partial x \partial y} = \frac{\partial^2 u}{\partial \alpha^2} - \frac{\partial^2 u}{\partial \alpha \partial \beta},$$

$$\frac{\partial^2 u}{\partial y^2} = \frac{\partial^2 u}{\partial \alpha^2}.$$

Finally, substituting these into equation $(A.4)$, we get the canonical form:

$$\frac{\partial^2 u}{\partial \alpha^2} + \frac{\partial^2 u}{\partial \beta^2} + \frac{\partial u}{\partial \alpha} - 3u = 0.$$

Chapter 2

Study of the Finite Differences Method

In this chapter, we present the finite difference method associated with the numerical solution of partial differential equations (PDEs).

2.1 Discretization of the Domain (Meshing)

To numerically solve a partial differential equation (PDE), we need to introduce a discretization (a mesh) of the domain D.

Definition 2.1.1. *A mesh is the discretization of a continuous interval. The intersection of two lines of the mesh defines a node .*

For example, if we consider a domain $D = [0,L] \times [0,T]$ and a PDE in two variables (space + time) with $u(x,t)$ as the solution of this PDE, we introduce a spatial mesh with step Δx and a time mesh with step Δt. We construct a node with coordinates:

$$x_i = i\Delta x, \quad i = 0,\ldots,M,$$

$$t_j = j\Delta t, \quad j = 0,\ldots,N.$$

Definition 2.1.2. *The finite difference method is used to approximate partial derivatives at points x_i, with values $u(x_i) = u(i\Delta x)$. We denote this approximation by u_i.*

Construction of a Scheme

Let u be a function of class C^4. If Δx is sufficiently small, we have the following Taylor expansions in the neighborhood of x:

$$u(x+\Delta x) = u(x) + \Delta x\, u'(x) + \frac{(\Delta x)^2}{2} u''(x) + \frac{(\Delta x)^3}{4} u'''(x) + O\left((\Delta x)^4\right), \tag{5.2}$$

$$u(x-\Delta x) = u(x) - \Delta x\, u'(x) + \frac{(\Delta x)^2}{2} u''(x) - \frac{(\Delta x)^3}{4} u'''(x) + O\left((\Delta x)^4\right). \tag{5.3}$$

After a simple calculation, we obtain the following equalities:

$$u'(x) = \frac{u(x+\Delta x) - u(x)}{\Delta x} + O(\Delta x), \tag{5.4}$$

$$u'(x) = \frac{u(x) - u(x-\Delta x)}{\Delta x} + O(\Delta x), \tag{5.5}$$

$$u'(x) = \frac{u(x+\Delta x) - u(x-\Delta x)}{2\Delta x} + O\left((\Delta x)^2\right), \tag{5.6}$$

and

$$u''(x) = \frac{u(x+\Delta x) - 2u(x) + u(x-\Delta x)}{(\Delta x)^2} + O\left((\Delta x)^2\right). \tag{5.7}$$

Equation (5.4) is called the right-hand approximation of u' and Equation (5.5) the left-hand approximation of u'; these approximations are of order 1. Equation (5.6) is the centered approximation of u', which is of order 2. Equation (5.7) is called the centered approximation of u''.

If we consider a function u of two variables, the approximations of partial derivatives are as follows:

$$\begin{aligned}
\frac{\partial u}{\partial x}(x,y) &= \frac{u(x+\Delta x,y) - u(x,y)}{\Delta x} + O(\Delta x),\\
\frac{\partial u}{\partial x}(x,y) &= \frac{u(x,y) - u(x-\Delta x,y)}{\Delta x} + O(\Delta x),\\
\frac{\partial u}{\partial x}(x,y) &= \frac{u(x+\Delta x,y) - u(x-\Delta x,y)}{2\Delta x} + O\left((\Delta x)^2\right),
\end{aligned}$$

and

$$\frac{\partial^2 u}{\partial x^2}(x,y) = \frac{u(x+\Delta x,y) - 2u(x,y) + u(x-\Delta x,y)}{(\Delta x)^2} + O\left((\Delta x)^2\right).$$

These formulas are used to approximate the partial derivatives of a partial differential equation (PDE).

Definition 2.1.3. *A numerical scheme is said to be consistent if the difference R_i^j between the approximation of the partial derivatives and the PDE converges to 0 as the steps Δt and Δx approach 0.*

Moreover, if there exists a positive constant C, independent of the solution of the PDE, such that R_i^j satisfies:

$$\|R_i^j\|_1 \leq C\left((\Delta t)^p + (\Delta x)^q\right), \quad p \geq 0, q \geq 0, \tag{5.8}$$

then the scheme is said to be of order p in time and q in space.

Definition 2.1.4. *(Matrix Norm) A matrix norm $\|\cdot\|$ is a norm on a vector space $M_{n,n}(H)$ of $n \times n$ matrices with entries in H. Given a matrix B with n rows and n columns, the matrix norm is defined by:*

$$\|B\| = \sup_{(x \in \mathbb{R}^n, x \neq 0)} \frac{\|Bx\|}{\|x\|},$$

where $\|B\|$ satisfies the following properties:

1. $\|B\| \geq 0$.
2. $\|B\| = 0$ *if and only if* $B = 0$.
3. $\|\lambda B\| = |\lambda| \|B\|$ *for all* $\lambda \in \mathbb{R}$.
4. $\|B + C\| \leq \|B\| + \|C\|$ *(triangle inequality).*

2.2 Elliptic Equation

Now, we consider the following problem:

$$\begin{cases} -u'' + c(x)u(x) = f(x), & x \in (0,1), \\ u(0) = \alpha, \quad u(1) = \beta, \end{cases} \tag{5.9}$$

where f and c are given functions defined on $\Omega = [0,1]$, and $\alpha, \beta \in \mathbb{R}$. This system models a diffusion phenomenon, such as heat diffusion.

To approximate the derivatives in the system's equation, we subdivide the interval $[0,1]$ into a finite number of subintervals. We introduce the mesh given by $x_i = ih$ where $i = 0, \ldots, N+1$ and h is the step size between points, defined as:

$$h = \frac{1}{N+1}.$$

So, we have $0 = x_0 \leq x_1 \leq \cdots \leq x_N \leq x_{N+1} = 1$. We choose h small enough that the number of mesh points becomes very large. At the boundary of $\Omega = [0,1]$, we have $x_0 = 0$ and $x_{N+1} = 1$. For all $1 \leq i \leq N$, we set $u_i = u(x_i)$ with $u(x_0) = \alpha$ and $u(x_{N+1}) = \beta$.

2.2.1 Numerical Scheme

Assuming the functions $f, c \in C([0,1])$ and $u \in C^4$, we use Taylor expansion near x_i to obtain:

$$u(x_{i+1}) = u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i) + \frac{h^3}{6}u'''(x_i) + \frac{h^4}{24}u^{(4)}(\xi), \quad 1 \leq i \leq N+1, \, \xi \in [x_i, x_{i+1}],$$

and

$$u(x_{i-1}) = u(x_i) - hu'(x_i) + \frac{h^2}{2}u''(x_i) - \frac{h^3}{6}u'''(x_i) + \frac{h^4}{24}u^{(4)}(\eta), \quad 1 \leq i \leq N+1, \, \eta \in [x_{i-1}, x_i].$$

This yields

$$u''(x_i) \approx \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}.$$

Thus,

$$(F_h) \begin{cases} -\frac{u_{i+1} - 2u_i + u_{i-1}}{h^2} + c_i u_i = f_i, & i \in \{1, \ldots, N\}, \\ u_0 = \alpha, \quad u_{N+1} = \beta. \end{cases}$$

This problem consists of N equations and two additional boundary conditions. We can rewrite (F_h) in the following matrix form:

$$A_h u_h = b_h,$$

with

$$A_h = \frac{1}{h^2} \begin{pmatrix} 2 + c_1 h^2 & -1 & 0 & \ldots & 0 & \\ -1 & 2 + c_2 h^2 & -1 & \ldots & 0 & \\ 0 & -1 & 2 + c_3 h^2 & \ldots & 0 & \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ 0 & \ldots & 0 & \ldots & 2 + c_{N-1} h^2 & -1 \\ 0 & \ldots & 0 & \ldots & -1 & 2 + c_N h^2 \end{pmatrix}$$

and

$$b_h = \begin{pmatrix} f_1 + \frac{\alpha}{h^2} \\ f_2 \\ \vdots \\ f_{N-1} \\ f_N + \frac{\beta}{h^2} \end{pmatrix}.$$

This problem raises at least two questions:

1. Does a unique solution exist for the system (F_h)?
2. Does u_h converge to the exact solution, and if so, in what sense

In other words, we need to determine if the matrix A_h is invertible. The answer is provided by the following proposition.

Proposition 2.2.1. *Assuming $c(x) \geq 0$ for all $x \in \Omega$, the matrix A_h is symmetric and positive definite; hence, A_h is invertible.*

Proof. The matrix A_h is obviously symmetric. Let us show that it is positive definite. Let $v = (v_1, \ldots, v_N)$, and define $v_0 = v_{N+1} = 0$. We calculate the dot product $A_h v \cdot v = v^T A_h v$. We have:

$$A_h v \cdot v = \frac{1}{h^2} \begin{pmatrix} v_1 & v_2 & \ldots & v_N \end{pmatrix} \begin{pmatrix} 2+c_1h^2 & -1 & 0 & \ldots & 0 \\ -1 & 2+c_2h^2 & -1 & \ldots & 0 \\ 0 & -1 & 2+c_3h^2 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & 0 & \ldots & 2+c_Nh^2 \end{pmatrix}$$

$$\begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{pmatrix}$$

That is:

$$A_h v \cdot v = \frac{1}{h^2} \sum_{i=1}^{N} v_i \left(-v_{i-1} + (2 + c_i h^2) v_i - v_{i+1} \right).$$

By changing the index, we get:

$$A_h v \cdot v = \frac{1}{h^2} \left[\sum_{i=1}^{N} (-v_{i-1} v_i) + \sum_{i=1}^{N} (2 + c_i h^2) v_i^2 - \sum_{j=2}^{N+1} v_{j-1} v_j \right].$$

Since we have set $v_0 = 0$ and $v_{N+1} = 0$, we can write:

$$A_h v \cdot v = \frac{1}{h^2} \sum_{i=1}^{N} (2 + c_i h^2) v_i^2 + \frac{1}{h^2} \sum_{i=1}^{N} (-2 v_i v_{i-1}).$$

This simplifies to:

$$A_h v \cdot v = \sum_{i=1}^{N} c_i v_i^2 + \frac{1}{h^2} \sum_{i=1}^{N} (-2v_i v_{i-1} + v_i^2 + v_{i-1}^2) + v_N^2.$$

Finally, we have:

$$A_h v \cdot v = \sum_{i=1}^{N} c_i v_i^2 + \frac{1}{h^2} \sum_{i=1}^{N+1} (v_i - v_{i-1})^2, \quad \forall v = (v_1, \ldots, v_N) \in \mathbb{R}^N.$$

If we assume $A_h v \cdot v = 0$, we obtain:

$$\sum_{i=1}^{N} c_i h^2 v_i^2 = 0 \quad \text{and} \quad v_i - v_{i-1} = 0, \forall i = 1, \ldots, N+1.$$

This implies that $v_1 = v_2 = \cdots = v_N = v_0 = v_{N+1} = 0$. Note that these equalities hold even if the c_i's are zero. This demonstrates that the matrix A_h is well-defined.

□

2.2.2 Consistency of the Scheme

The concept of consistency is used to understand the asymptotic behavior of the approximate solution near the exact solution. We want to determine if the problem (F_h) provides a good approximation of problem (5.9). We define the following operator:

$$(L_h u)(x_i) = -\frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} + c(x_i)u(x_i),$$

which gives us the following discrete problem:

$$(L_h u)(x_i) = f_i.$$

We need the following definition for the consistency of the numerical scheme.

Definition 2.2.1. *A numerical scheme is said to be consistent if the consistency error $R_h := (L_h u)(x_i) - f_i$ tends uniformly to zero with respect to x as $h \to 0$, i.e.,*

$$\lim_{h \to 0} \|R_h\|_1 = 0.$$

Proposition 2.2.2. *Assume $u \in C^4([0,1])$. Then the scheme F_h is consistent of order 2. Moreover, there exists a constant $C \geq 0$, independent of u, such that*

$$\|R_h\|_1 \leq C\frac{h^2}{12}, \quad C = \sup_{[0,1]} |u^{(4)}|.$$

Proof. By Taylor expansion, we have:

$$u(x_{i+1}) = u(x_i) + hu'(x_i) + \frac{h^2}{2}u''(x_i) + \frac{h^3}{6}u^{(3)}(x_i) + \frac{h^4}{24}u^{(4)}(\xi_i),$$

$$u(x_{i-1}) = u(x_i) - hu'(x_i) + \frac{h^2}{2}u''(x_i) - \frac{h^3}{6}u^{(3)}(x_i) + \frac{h^4}{24}u^{(4)}(\eta_i).$$

By adding these two equations, we obtain:

$$\frac{1}{h^2}\left(u(x_{i+1})+u(x_i)-2u(x_i)\right)=u''(x_i)+\frac{h^2}{24}\left(u^{(4)}(\xi_i)+u^{(4)}(\eta_i)\right),$$

which leads to:

$$|R_i|\leq\frac{h^2}{12}\sup_{[0,1]}|u^{(4)}|.$$

□

The proof of convergence of the scheme uses the notion of consistency, as well as a notion of stability, which we now introduce:

Proposition 1.15: We say that the scheme (1.24) is stable, in the sense that the infinity norm of the approximate solution is bounded by a number depending only on f. More precisely, the discretization matrix A_h satisfies:

$$\|A_h^{-1}\|_\infty\leq\frac{1}{8}\quad(1.33),$$

an inequality which can also be written as an estimate on the solutions of the system (1.25):

$$\|U_h\|_\infty\leq\frac{1}{8}\|f\|_\infty\quad(1.34).$$

Proof. Recall that by definition, for $M\in M_N(\mathbb{R})$,

$$\|M\|_\infty=\sup_{v\in\mathbb{R}^N,v\neq0}\frac{\|Mv\|_\infty}{\|v\|_\infty},$$

where $\|v\|_\infty=\sup_{i=1,\dots,N}|v_i|$.

To show that $\|A_h^{-1}\|_1\leq\frac{1}{8}$, we decompose the matrix A_h as $A_h=A_0h+\mathrm{diag}(c_i)$, where A_0h is the discretization matrix of the operator $-u''$ with homogeneous Dirichlet boundary conditions, and:

$$A_0h=\begin{pmatrix}\frac{2}{h^2}&-\frac{1}{h^2}&0&\dots&0\\-\frac{1}{h^2}&\frac{2}{h^2}&-\frac{1}{h^2}&\dots&0\\0&-\frac{1}{h^2}&\frac{2}{h^2}&\dots&0\\\vdots&\vdots&\vdots&\ddots&\vdots\\0&\dots&0&\dots&\frac{2}{h^2}\end{pmatrix}$$

The matrix $\mathrm{diag}(c_i)$ represents the diagonal matrix with diagonal coefficients c_i. The matrices A_0^h and A_h are invertible, and we have:

$$A_0^{-1}-A_h^{-1}=A_0^{-1}A_hA_h^{-1}-A_0^{-1}A_0A_h^{-1}=A_0^{-1}(A_h-A_0)A_h^{-1}.$$

Since $\mathrm{diag}(c_i)\geq0$, we have $A_h\geq A_0^h$, and since A_0^h and A_h are monotone, we deduce that:

$$0\leq A_h^{-1}\leq A_0^{-1}\quad\text{(component-wise)}.$$

Now, we can observe that if $B\in\mathbb{M}_N(\mathbb{R})$ and $B\geq0$ (i.e., $B_{ij}\geq0$ for all i and j), we have:

$$\|B\|_1=\sup_{v\in\mathbb{R}^N,v\neq0}\frac{\|Bv\|_1}{\|v\|_\infty}=\sup_{v\in\mathbb{R}^N,\|v\|_\infty=1}\sup_{i=1,\dots,N}\left|\sum_{j=1}^N B_{ij}v_j\right|=\sup_{i=1,\dots,N}\sum_{j=1}^N B_{ij}.$$

Thus, $\|A_h^{-1}\|_\infty = \sup_{i=1,\dots,N} \sum_{j=1}^{N} (A_h^{-1})_{ij} \leq \sup_{i=1,\dots,N} \sum_{j=1}^{N} (A_0^{-1})_{ij}$ because $A_h^{-1} \leq A_0^{-1}$; hence, we deduce that:

$$\|A_h^{-1}\|_\infty \leq \|A_0^{-1}\|_\infty.$$

It remains to estimate $\|A_0^{-1}\|_\infty$. Since $A_0^{-1} \geq 0$, we have:

$$\|A_0^{-1}\|_\infty = \|A_0^{-1} e\|_\infty \quad \text{where} \quad e = (1,\dots,1).$$

Let $d = A_0^{-1} e \in \mathbb{R}^N$. We want to calculate $\|d\|_\infty$, where d satisfies $A_0^h d = e$. The linear system $A_0^h d = e$ is nothing more than the finite difference discretization of the problem:

$$\begin{cases} -u'' = 1 \\ u(0) = u(1) = 0 \end{cases}$$

whose exact solution is:

$$u_0(x) = \frac{x(1-x)}{2},$$

which satisfies $u_0^{(4)}(x) = 0$. From Remark 1.14, we conclude that:

$$u_0(x_i) = d_i, \quad \forall i = 1,\dots,N.$$

Thus, $\|d\|_\infty = \sup_{i=1,N} \left| \frac{h(i(h-1))}{2} \right|$, where $h = \frac{1}{N+1}$ is the discretization step. This leads to:

$$\|d\|_1 \leq \sup_{x\in[0,1]} \left| \frac{x(x-1)}{2} \right| = \frac{1}{8},$$

and thus:

$$\|A_h^{-1}\|_1 \leq \frac{1}{8}.$$

Remark 2.2.1. *(On Stability): Note that inequality (1.34) gives an estimation on the approximate solutions independent of the mesh size. This type of estimation will be sought later for the discretization of other problems as a guarantee of the stability of a numerical scheme.*

Definition 2.2.2. *(Discretization Error): The discretization error at x_i is the difference between the exact solution at x_i and the i-th component of the solution given by the numerical scheme:*

$$e_i = u(x_i) - u_i, \quad \forall i = 1,\dots,N.$$

Theorem 2.2.1. *Let u be the exact solution of:*

$$\begin{cases} -u'' + cu = f \\ u(0) = u(1) = 0 \end{cases} \tag{2.1}$$

and suppose $u \in C^4([0,1])$. Let u_h be the solution of (2.1). Then, the discretization error satisfies:

$$\max_{i=1,\dots,N} |e_i| \leq \frac{1}{96} \|u^{(4)}\|_\infty h^2.$$

Thus, the scheme is convergent of order 2.

Remark 2.2.2. *(On Stability): Note that inequality (1.34) gives an estimation on the approximate solutions independent of the mesh size. This type of estimation will be sought later for the discretization of other problems as a guarantee of the stability of a numerical scheme.*

Definition 2.2.3. *(Discretization Error): The discretization error at x_i is the difference between the exact solution at x_i and the i-th component of the solution given by the numerical scheme:*

$$e_i = u(x_i) - u_i, \quad \forall i = 1, \dots, N.$$

Theorem 2.2.2. *Let u be the exact solution of:*

$$\begin{cases} -u'' + cu = f \\ u(0) = u(1) = 0 \end{cases}$$

and suppose $u \in C^4([0,1])$. Let u_h be the solution of (1.24). Then, the discretization error defined by (1.37) satisfies:

$$\max_{i=1,\dots,N} |e_i| \leq \frac{1}{96} \|u^{(4)}\|_\infty h^2.$$

Thus, the scheme is convergent of order 2.

Proof. Let $U_h = (U_1, \dots, U_n)$ and $\overline{U_h} = (u(x_1), \dots, u(x_N))$, we seek to bound $\|\overline{U_h} - U_h\|_1$. We have $A(\overline{U_h} - U_h) = R$, where R is the consistency error (see Remark 1.14). Therefore:

$$\|\overline{U_h} - U_h\|_\infty \leq \|A_h^{-1}\|_\infty \|R\|_\infty \leq \frac{1}{8} \times \frac{1}{12} \|u^{(4)}\|_\infty = \frac{1}{96} \|u^{(4)}\|_\infty.$$

□

Remark 2.2.3. *(On convergence): It can be noted that the proof of convergence relies on stability (which itself is derived from the preservation of positivity) and consistency. In some numerical analysis textbooks, you will find the formula: stability + consistency = convergence.*

2.3 Parabolic Problems: Time Discretization

Now consider the same problem in one spatial dimension. At time $t = 0$, we are given an initial condition u_0, and we assume homogeneous Dirichlet boundary conditions. The one-dimensional problem is written as:

$$\begin{cases} u_t - u_{xx} = 0, & \text{for } x \in (0,1),\ t \in (0,T), \\ u(x,0) = u_0(x), & \text{for } x \in (0,1), \\ u(0,t) = u(1,t) = 0, & \text{for } t \in (0,T), \end{cases}$$

where $u(x,t)$ represents the temperature at the point x and time t, u_t is the first partial derivative of u with respect to t, and u_{xx} is the second partial derivative of u with respect to x. We will assume the following existence and uniqueness theorem.

Theorem 2.3.1. *(Existence and Uniqueness Result) If $u_0 \in C(]0,1[,\mathbb{R})$, there exists a unique function $u \in C^2(]0,1[\times]0,T[,\mathbb{R}) \cap C([0,1]\times[0,T],\mathbb{R})$ satisfying (2.1). Furthermore, $u \in C^1(]0,1[\times]0,T[,\mathbb{R})$, which highlights the "regularizing" effect of the heat equation.*

Proposition 2.3.1. *(Maximum Principle) Under the assumptions of Theorem 2.1, let u be the solution of the problem (2.1):*

1. *If* $u_0(x) \geq 0$ *for all* $x \in [0,1]$*, then* $u(x,t) \geq 0$ *for all* $t \geq 0$ *and* $x \in]0,1[$.
2. $\|u\|_{L^1([0,1]\times]0,T[)} \leq \|u_0\|_{L^1(]0,1[)}$.

These properties are significant in physical modeling. For instance, suppose u represents a mass fraction. By definition, the mass fraction is always within the interval $[0,1]$. The above proposition guarantees that if the initial mass fraction u_0 lies within [0, 1], the mathematical model ensures that u, representing the diffusing species in a medium, also remains within [0, 1]. This consistency with physical bounds is reassuring.

However, the analytical solution to (2.1) is generally not computable. Numerical discretization in time and space is employed to approximate the solution, resulting in a finite-dimensional system of equations. For reliability, it is essential that the approximate solution also respects the physical bounds at all times. A discretization method (or scheme) is termed "robust" or "stable" if it preserves such bounds.

2.3.1 Numerical Approximation of the Solution

To compute an approximate solution, we discretize the time and space domains, denoted by D. For simplicity, we consider a uniform discretization. Let: - $h = \Delta x = \frac{1}{N+1}$, the spatial step size, - $k = \Delta t = \frac{T}{M}$, the temporal step size.

Define $t_n = nk$ for $n = 0,\ldots,M$ and $x_i = ih$ for $i = 0,\ldots,N+1$. We seek to calculate an approximate solution u_D of (2.1), specifically $u_D(x_i,t_n)$ for $i = 1,\ldots,N$ and $n = 1,\ldots,M$. The discrete unknowns are denoted $u_i^{(n)}$ for $i = 1,\ldots,N$ and $n = 1,\ldots,M$.

Explicit Euler Time Discretization

The explicit Euler method approximates the time derivative $u_t(x_i,t_n)$ as:

$$\frac{u(x_i,t_{n+1}) - u(x_i,t_n)}{k},$$

and the spatial derivative $-u_{xx}(x_i,t_n)$ as:

$$\frac{1}{h^2}\left(2u(x_i,t_n) - u(x_{i-1},t_n) - u(x_{i+1},t_n)\right).$$

Remark 2.3.1. *Although we use finite differences for spatial discretization, finite volume or finite element methods could also be employed.*

The resulting explicit scheme is:

$$\begin{aligned}
&\frac{u_i^{(n+1)} - u_i^{(n)}}{k} + \frac{1}{h^2}\left(2u_i^{(n)} - u_{i-1}^{(n)} - u_{i+1}^{(n)}\right) = 0, \quad i = 1,\ldots,N,\ n = 1,\ldots,M,\\
&u_i^{(0)} = u_0(x_i), \quad i = 1,\ldots,N,\\
&u_0^{(n)} = u_{N+1}^{(n)} = 0, \quad n = 1,\ldots,M.
\end{aligned}$$

This explicit scheme computes $u_i^{(n+1)}$ explicitly from $u_i^{(n)}$:

$$u_i^{(n+1)} = u_i^{(n)} - \mu\left(2u_i^{(n)} - u_{i-1}^{(n)} - u_{i+1}^{(n)}\right),$$

where $\mu = \frac{k}{h^2}$.

2.3.2 Consistency of the Scheme

Let $\bar{u}_i^{(n)} = u(x_i, t_n)$ be the exact solution at (x_i, t_n). The consistency error $R_i^{(n)}$ can be expressed as:

$$R_i^{(n)} = \tilde{R}_i^{(n)} + \hat{R}_i^{(n)},$$

where

$$\tilde{R}_i^{(n)} = \frac{\bar{u}_i^{(n+1)} - \bar{u}_i^{(n)}}{k} - u_t(x_i, t_n), \quad \hat{R}_i^{(n)} = \frac{1}{h^2}\left(2\bar{u}_i^{(n)} - \bar{u}_{i-1}^{(n)} - \bar{u}_{i+1}^{(n)}\right) - u_{xx}(x_i, t_n).$$

Proposition 2.3.2. *The scheme (2.2) is first-order consistent in time and second-order consistent in space:*

$$|R_i^{(n)}| \leq C(k + h^2),$$

where C depends only on u.

2.4 Stability

Definition 2.4.1. *A scheme is L^1-stable if the approximate solution remains bounded in L^1, independent of the mesh step size.*

Proposition 2.4.1. *If the stability condition $\mu = \frac{k}{h^2} \leq \frac{1}{2}$ is satisfied, the scheme (2.2) is L^1-stable:*

$$\sup_{i=1,\ldots,N} \sup_{n=1,\ldots,M} |u_i^{(n)}| \leq \|u_0\|_{L^1}.$$

Proof. we have that

$$u_i^{(n+1)} = u_i^{(n)} - \lambda\left(2u_i^{(n)} - u_{i-1}^{(n)} - u_{i+1}^{(n)}\right),$$

or equivalently:

$$u_i^{(n+1)} = (1 - 2\lambda)u_i^{(n)} + \lambda u_{i-1}^{(n)} + \lambda u_{i+1}^{(n)}.$$

If $0 \leq \lambda \leq \frac{1}{2}$, then $\lambda \geq 0$ and $1 - 2\lambda \geq 0$, ensuring that $u_i^{(n+1)}$ is a convex combination of $u_i^{(n)}$, $u_{i-1}^{(n)}$, and $u_{i+1}^{(n)}$.

Let $M^{(n)} = \max_{i=1,\ldots,N} u_i^{(n)}$. It follows that:

$$u_i^{(n+1)} \leq (1 - 2\lambda)M^{(n)} + \lambda M^{(n)} + \lambda M^{(n)} \quad \forall i = 1, \ldots, N.$$

Thus, $u_i^{(n+1)} \leq M^{(n)}$, and by taking the maximum over all i, we deduce:

$$M^{(n+1)} \leq M^{(n)}.$$

Similarly, for the minimum, let $m^{(n)} = \min_{i=1,\ldots,N} u_i^{(n)}$. We can show:

$$u_i^{(n+1)} \geq (1 - 2\lambda)m^{(n)} + \lambda m^{(n)} + \lambda m^{(n)} \quad \forall i = 1, \ldots, N,$$

which implies:

$$m^{(n+1)} \geq m^{(n)}.$$

Combining these results, we conclude:

$$\max_{i=1,\ldots,N} u_i^{(n+1)} \leq \max u_i^{(0)}, \quad \min_{i=1,\ldots,N} u_i^{(n+1)} \geq \min u_i^{(0)}.$$

This proves the desired result. □

2.5 Convergence

Definition 2.5.1. *The discretization error at* (x_i,t_n) *is* $e_i^{(n)} = u(x_i,t_n) - u_i^{(n)}$.

Theorem 2.5.1. *Under the assumptions of Theorem 2.1 and the stability condition* $\mu \leq \frac{1}{2}$*, there exists C depending only on u such that:*

$$\|e_i^{(n+1)}\|_1 \leq \|e_i^{(0)}\|_1 + TC(k+h^2), \quad \forall i,n.$$

If $\|e_i^{(0)}\|_1 = 0$*, then* $\max_{i=1,\dots,N} \|e_i^{(n)}\| \to 0$ *as* $k,h \to 0$*. Thus, the scheme (2.2) is convergent.*

Proof. Let $\bar{u}_i^{(n)} = u(x_i,t_n)$. By the definition of the consistency error, we have:

$$\bar{u}_i^{(n+1)} - \bar{u}_i^{(n)} \frac{k}{h^2}(2\bar{u}_i^{(n)} - \bar{u}_{i-1}^{(n)} - \bar{u}_{i+1}^{(n)}) = R_i^{(n)} \quad (2.5)$$

On the other hand, the numerical scheme can be written as:

$$u_i^{(n+1)} - u_i^{(n)} \frac{k}{h^2}(2u_i^{(n)} - u_{i-1}^{(n)} - u_{i+1}^{(n)}) = 0 \quad (2.6)$$

Subtracting equation (2.6) from equation (2.5), we obtain:

$$e_i^{(n+1)} - e_i^{(n)} \frac{k}{h^2}(2e_i^{(n)} - e_{i+1}^{(n)} - e_{i-1}^{(n)}) = R_i^{(n)},$$

which simplifies to:

$$e_i^{(n+1)} = (1-2\alpha)e_i^{(n)} + \alpha e_{i-1}^{(n)} + \alpha e_{i+1}^{(n)} + kR_i^{(n)}$$

Since $(1-2\alpha)e_i^{(n)} + \alpha e_{i-1}^{(n)} + \alpha e_{i+1}^{(n)} \leq ke_i^{(n)}\|1$, because $\alpha \leq \frac{1}{2}$, and since the scheme is consistent, inequality (2.3) leads to:

$$|e_i^{(n+1)}| \leq ke_i^{(n)}\|1 + kC(k+h^2).$$

By recurrence, we obtain:

$$ke_i^{(n+1)}\|1 \leq ke_i^{(0)}\|1 + MkC(k+h^2),$$

which proves the theorem.

□

2.5.1 Exercises

Exercise 01

We want to find a convergent numerical scheme for the following problem:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \dfrac{\partial^2 u}{\partial x^2} - u(t,x) = 0, & (x,t) \in (0,1) \times \mathbb{R}^+, \\ u(x,0) = u_0(x), & x \in (0,1), \\ u(0,t) = u(1,t) = 0, & t \in \mathbb{R}^+. \end{cases} \quad (5.43)$$

Let $\Delta x = \frac{1}{N}$ with $\Delta t > 0$. We discretize (x,t) as $x_i = i\Delta x$, $i = 0,\dots,N$ and $t^n = n\Delta t$, $n = 1,\dots,M$. We denote

$$\tilde{u}_i^n = u(x_i,t^n) \quad \textit{(the exact solution at } (x_i,t^n)\textit{)},$$

and

$$u_i^n \quad \textit{(the approximate solution at } (x_i,t^n)\textit{)}.$$

1. *Find the numerical scheme for problem (5.43).*
2. *Let R_i^n be the consistency error of the numerical scheme. Show that R_i^n is bounded by $C(\Delta t + \Delta x^2)$.*
3. *Write the scheme in matrix form.*
4. *For $n \in \mathbb{N}$, set*
$$\|u^n\|_\infty = \sup_{i=1,\dots,N} \left|u_i^n\right|.$$
Show that there exists C such that
$$\|u^n\|_\infty \leq C(T,\alpha)\,\|u^0\|_\infty$$
for the numerical scheme.
5. *For $n \in \mathbb{N}$ and $i \in \{1,\dots,N\}$, let*
$$e_i^n = \tilde{u}_i^n - u_i^n.$$
Give bounds on $\|e^n\|_\infty$ in terms of T and C.

Exercise 02

Let $\alpha \geq 0$, $\mu \geq 0$, $T \geq 0$, and $u_0 : \mathbb{R} \to \mathbb{R}$. We consider the following problem:

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) + \alpha \dfrac{\partial u}{\partial x}(t,x) - \mu \dfrac{\partial^2 u}{\partial x^2}(t,x) = 0, & (x,t) \in (0,1)\times(0,T), \\ u(0,t) = u(1,t) = 0, & t \in (0,T), \\ u(x,0) = u_0(x), & x \in (0,1). \end{cases} \tag{5.44}$$

Assume that there exists a classical solution $u \in C^4([0,1]\times[0,T])$ to problem (5.44). We discretize the problem as follows: take $h = \frac{1}{N+1}$ and $k = \frac{T}{M}$ with $x_i = ih$ and $t^n = nk$.

1. *Find the explicit numerical scheme associated with problem (5.44), using a forward approximation in time and a backward approximation in space.*
2. *Let $\tilde{u}_i^n = u(ih,nk)$ for $i = 0,\dots,N+1$ and $n = 0,\dots,M$. Show that the consistency error R_i^n of the scheme is bounded by $C_1(k+h)$.*
3. *Study the stability of the scheme.*

Exercise 03

Consider the following system:

$$\begin{cases} -\dfrac{d^2u}{dx^2}(x) + \dfrac{1}{1+x}\dfrac{du}{dx}(x) = f(x), \quad x \in (0,1), \\ u(0) = a_0, \quad u(1) = a_1, \end{cases} \tag{5.45}$$

where f is a function of class $C^2([0,1])$.

Assume that this problem admits a unique solution $u \in C^4([0,1])$. Let $h = \frac{1}{N+1}$ with $N \in \mathbb{N}$, and denote by u_i the numerical solution at the grid point $x_i = ih$, for $i \in \{0,\dots,N+1\}$.

1. *Derive the finite difference scheme using centered approximations.*
2. *Show that the scheme is consistent.*
3. *Write the scheme in the form $Au = b$, and identify A and b.*
4. *Show that if $Av \geq 0$ then $v \geq 0$, for all $v \in \mathbb{R}^N$.*
5. *Deduce that the matrix A is monotone.*

2.5.2 Correction of Exercises

Exercise 1

We consider the initial boundary-value problem

$$\begin{cases} u_t(t,x) - u_{xx}(t,x) - u(t,x) = 0, & (t,x) \in (0,T] \times (0,1), \\ u(x,0) = u_0(x), & x \in [0,1], \\ u(0,t) = u(1,t) = 0, & t \in [0,T]. \end{cases} \tag{P}$$

Let $\Delta x = 1/N$ and $\Delta t > 0$. Denote grid points $x_i = i\Delta x$, $i = 0,\dots,N$, and $t^n = n\Delta t$, $n = 0,\dots,M$ with $M\Delta t = T$. Let $\tilde{u}_i^n := u(x_i,t^n)$ be the exact solution sampled on the grid and u_i^n the numerical values.

(1) Explicit and implicit discrete schemes

Because $u_t = u_{xx} + u$, the Forward Euler / central space (explicit) scheme is

$$\boxed{u_i^{n+1} = u_i^n + \Delta t\Big(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} + u_i^n\Big)}$$

for $i = 1,\dots,N-1$, with $u_0^n = u_N^n = 0$.

The Backward Euler (implicit) scheme is

$$\boxed{u_i^{n+1} = u_i^n + \Delta t\Big(\frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} + u_i^{n+1}\Big)}$$

which can be rearranged (set $\lambda = \frac{\Delta t}{(\Delta x)^2}$) to

$$(1 - \Delta t + 2\lambda)\, u_i^{n+1} - \lambda\, u_{i-1}^{n+1} - \lambda\, u_{i+1}^{n+1} = u_i^n.$$

(2) Local truncation error (consistency)

Define the local truncation error of the implicit scheme at (x_i, t^n) by

$$R_i^n := \frac{\tilde{u}_i^{n+1} - \tilde{u}_i^n}{\Delta t} - \frac{\tilde{u}_{i+1}^{n+1} - 2\tilde{u}_i^{n+1} + \tilde{u}_{i-1}^{n+1}}{(\Delta x)^2} - \tilde{u}_i^{n+1}.$$

Use Taylor expansions in time and space (expand $\tilde{u}_i^{n+1}$ about (x_i, t^n) and the four-point spatial expansions about x_i). One finds

$$\tilde{u}_i^{n+1} = \tilde{u}_i^n + \Delta t\, u_t(x_i, t^n) + \frac{\Delta t^2}{2} u_{tt}(\xi_t)$$

and

$$\frac{\tilde{u}_{i+1}^{n+1} - 2\tilde{u}_i^{n+1} + \tilde{u}_{i-1}^{n+1}}{(\Delta x)^2} = u_{xx}(x_i, t^{n+1}) + O(\Delta x^2).$$

Combining and using the PDE $u_t = u_{xx} + u$, we obtain

$$|R_i^n| \le C_1\, \Delta t + C_2\, \Delta x^2,$$

so there exists C with

$$\boxed{|R_i^n| \le C(\Delta t + \Delta x^2)}$$

(i.e. first order in time, second order in space).

(3) Matrix form of the implicit scheme

Let $U^n = (u_1^n, \dots, u_{N-1}^n)^T$. The implicit update can be written as

$$B U^{n+1} = U^n,$$

where B is the $(N-1) \times (N-1)$ tridiagonal matrix

$$B = \begin{pmatrix} 1 - \Delta t + 2\lambda & -\lambda & & \\ -\lambda & 1 - \Delta t + 2\lambda & -\lambda & \\ & \ddots & \ddots & \ddots \\ & & -\lambda & 1 - \Delta t + 2\lambda \end{pmatrix}, \qquad \lambda = \frac{\Delta t}{(\Delta x)^2}.$$

Equivalently $U^{n+1} = B^{-1} U^n$.

(4) Stability

Implicit (backward Euler). *The matrix B is an M-matrix for any $\Delta t > 0$ (diagonally dominant with positive diagonal and nonpositive off-diagonals) and is invertible with $B^{-1} \ge 0$. From $U^{n+1} = B^{-1} U^n$ and $\|B^{-1}\|_\infty$ bounded independently of n, we deduce a bound*

$$\|U^n\|_\infty \le C(T)\, \|U^0\|_\infty,$$

i.e. the implicit scheme is unconditionally stable *(in maximum norm and hence in L^1).*

(A direct energy proof: multiply the discrete equation by u_i^{n+1} and sum to obtain a discrete energy inequality; the $+u^{n+1}$ reaction term is handled by bringing it to the left-hand side.)

Explicit (forward Euler). *The explicit update matrix is*

$$U^{n+1} = \big(I + \Delta t(D+I)\big)U^n,$$

where D is the discrete Laplacian operator (matrix) with eigenvalues

$$\mu_j = -\frac{4}{(\Delta x)^2}\sin^2\left(\frac{j\pi}{2N}\right), \quad j = 1,\dots,N-1.$$

Von Neumann (or spectral) stability requires for every eigenmode

$$\big|1 + \Delta t(\mu_j + 1)\big| \le 1.$$

Using $\mu_j \in \big[-\frac{4}{(\Delta x)^2}, 0\big)$*, a sufficient (simple) CFL-type restriction is*

$$\boxed{\Delta t \le \frac{\Delta x^2}{4+\Delta x^2} \;\approx\; \frac{\Delta x^2}{4} \text{ for small } \Delta x.}$$

(For the pure heat equation $u_t = u_{xx}$ *the usual sharp restriction is* $\Delta t \le \frac{\Delta x^2}{2}$*; adding the reaction term* $+u$ *makes the explicit scheme more restrictive. The exact sharp bound can be obtained from the maximal eigenvalue condition above.)*

Thus the explicit scheme is conditionally stable*; the implicit scheme is* unconditionally stable.

(5) Convergence (Lax–Richtmyer)

The implicit scheme is consistent (see (2)) and stable (unconditional), hence by Lax equivalence it is convergent: $\|U^n - \tilde{U}^n\|_\infty \to 0$ *as* $\Delta t, \Delta x \to 0$*. Combining the truncation error estimate yields the usual global error bound*

$$\|e^n\|_\infty := \max_i |u_i^n - \tilde{u}_i^n| \le C(T)\,(\Delta t + \Delta x^2).$$

Exercise 2

We consider the convection-diffusion IBVP

$$\begin{cases} u_t + \alpha u_x - \mu u_{xx} = 0, & (x,t) \in (0,1)\times(0,T), \\ u(0,t) = u(1,t) = 0, \quad u(x,0) = u_0(x). \end{cases}$$

Discretize with space step h *and time step* k *(*$h = 1/(N+1)$*,* $k = T/M$*). Use the* explicit *scheme:*

$$\boxed{\frac{u_i^{n+1} - u_i^n}{k} + \alpha\frac{u_i^n - u_{i-1}^n}{h} - \mu\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} = 0}$$

(i.e. forward in time, backward (upwind) for convection, centered for diffusion). This is the standard explicit upwind+centered-diffusion scheme.

Local truncation error

Let $\tilde{u}_i^n = u(x_i, t^n)$. Expand in Taylor:

$$\frac{\tilde{u}_i^{n+1} - \tilde{u}_i^n}{k} = u_t(x_i, t^n) + \frac{k}{2} u_{tt}(\xi_t),$$

and for the spatial terms we have standard expansions with leading errors:

$$\frac{\tilde{u}_i^n - \tilde{u}_{i-1}^n}{h} = u_x(x_i, t^n) + O(h), \qquad \frac{\tilde{u}_{i+1}^n - 2\tilde{u}_i^n + \tilde{u}_{i-1}^n}{h^2} = u_{xx}(x_i, t^n) + O(h^2).$$

Using the PDE to cancel leading terms, the truncation error satisfies

$$\boxed{|R_i^n| \leq C_1 k + C_2 h}$$

so the scheme is first order in time and (because of the upwind term) first order in space overall: $O(k+h)$.

Stability analysis (practical, standard CFL conditions)

For an explicit scheme combining convection and diffusion, stability is controlled by two limits:

- ***Convection (upwind) CFL:*** *require*

$$\frac{\alpha k}{h} \leq 1 \qquad \textit{(often written } \nu_c := \alpha k/h \leq 1\textit{).}$$

  This prevents convective amplification (transport must not step over a cell).

- ***Diffusion CFL:*** *for the explicit centered discretization of u_{xx} require*

$$\frac{\mu k}{h^2} \leq \frac{1}{2} \qquad \textit{(classical bound for forward Euler + central space).}$$

Combining yields a safe sufficient condition

$$\boxed{k \leq \min\left(\frac{h}{\alpha}, \frac{h^2}{2\mu}\right)}$$

(when $\alpha > 0$, $\mu > 0$). If α is small the diffusion bound dominates; if μ is small the convective CFL dominates.

Von Neumann remark. *A von-Neumann (Fourier) analysis for the diffusion part recovers the $k \leq h^2/(2\mu)$ constraint; for convection the upwind discretization ensures $|G(\xi)| \leq 1$ under $\alpha k/h \leq 1$. The mixed analysis leads to the combined restriction above.*

Consistency + stability $\Rightarrow$ convergence

Because the scheme is consistent of order $O(k+h)$ and (under the CFL above) stable in the maximum norm, by Lax equivalence the scheme converges with global error $O(k+h)$.

Exercise 3

Consider the boundary-value problem

$$\begin{cases} -u''(x) + \dfrac{1}{1+x}u'(x) = f(x), \quad x \in (0,1), \\ u(0) = \alpha, \quad u(1) = \beta, \end{cases}$$

with $f \in C^2([0,1])$ and assume the exact solution $u \in C^4([0,1])$.

(1) Centered finite-difference scheme

Let $h = 1/(N+1)$ and $x_i = ih$, $i = 0, \ldots, N+1$. Use central difference approximations:

$$u''(x_i) \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{h^2}, \qquad u'(x_i) \approx \frac{u_{i+1} - u_{i-1}}{2h}.$$

Thus for $i = 1, \ldots, N$ the discrete equation is

$$\frac{-u_{i+1} + 2u_i - u_{i-1}}{h^2} + \frac{1}{1+x_i}\frac{u_{i+1} - u_{i-1}}{2h} = f_i,$$

or equivalently

$$\boxed{a_i u_{i-1} + b_i u_i + c_i u_{i+1} = f_i,\ i = 1, \ldots, N}$$

with

$$a_i = -\frac{1}{h^2} - \frac{1}{2h(1+x_i)}, \quad b_i = \frac{2}{h^2}, \quad c_i = -\frac{1}{h^2} + \frac{1}{2h(1+x_i)},$$

and boundary conditions $u_0 = \alpha$, $u_{N+1} = \beta$ included in the right-hand side when assembling b.

(2) Consistency

Apply Taylor expansions:

$$\frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2} = u''(x_i) + O(h^2), \qquad \frac{u(x_{i+1}) - u(x_{i-1})}{2h} = u'(x_i) + O(h^2).$$

Therefore the local truncation error is $O(h^2)$ and the scheme is second-order consistent in space.

(3) Matrix form $Au = b$

Collect interior unknowns $U = (u_1, \ldots, u_N)^T$. The discrete system is $AU = \tilde{f}$, where

$$A = \begin{pmatrix} b_1 & c_1 & 0 & \cdots & 0 \\ a_2 & b_2 & c_2 & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & \cdots & 0 & a_N & b_N \end{pmatrix},$$

and the right-hand side $\tilde{f}$ includes values α, β from the boundary conditions.

(4) Monotonicity: if $Av \geq 0$ then $v \geq 0$

We show A is an M-matrix (hence monotone). Observe for sufficiently small h:

- *The diagonal entries* $b_i = 2/h^2 > 0$.
- *Off-diagonals satisfy* $a_i \leq 0$, $c_i \leq 0$ *for all i (the* $1/(2h(1+x_i))$ *terms are* $O(1/h)$ *smaller than* $1/h^2$ *so sign is negative for small h).*
- *A is strictly diagonally dominant because*

$$|b_i| - (|a_i| + |c_i|) = \frac{2}{h^2} - \Big(\frac{1}{h^2} + \frac{1}{2h(1+x_i)} + \frac{1}{h^2} + \frac{1}{2h(1+x_i)}\Big) = -\frac{1}{h(1+x_i)} < 0,$$

but one can instead check the matrix $-A$ *has the sign pattern of a symmetric positive definite M-matrix after multiplication by* -1 *and reordering; more direct: A is irreducible and diagonally dominant in practice for fine meshes; standard theory for second-order elliptic operators discretized by centered differences yields an M-matrix.*

A standard discrete maximum-principle argument now applies: suppose $Av \geq 0$ *and v attains a negative minimum at interior index k. Then from row k we get*

$$a_k v_{k-1} + b_k v_k + c_k v_{k+1} \geq 0.$$

But $v_{k-1} \geq v_k$ *and* $v_{k+1} \geq v_k$ *so the left-hand side* $\leq (a_k + c_k + b_k)v_k = f\, v_k$*, and using the sign pattern one obtains a contradiction unless* $v_k \geq 0$*. Hence* $v \geq 0$*.*

Thus

$$\boxed{Av \geq 0 \Longrightarrow v \geq 0,}$$

so A is monotone and $A^{-1} \geq 0$*.*

(5) Invertibility / positivity of A^{-1}

Since A arises from a uniformly elliptic operator discretized by a regular mesh and homogeneous Dirichlet boundary conditions, A is nonsingular and an M-matrix; therefore $A^{-1} \geq 0$*. This gives uniqueness and positivity properties for solutions and implies good conditioning for small h.*

Chapter 3

Implicit method Crank Nicholson

3.1 Crank-Nicholson Method for the 1D Heat Equation

1. Problem Statement

We consider the 1D heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < L, t > 0,$$

with suitable initial and boundary conditions.

2. Numerical Scheme

The Crank-Nicholson scheme is given by:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha}{2} \left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2} \right).$$

Rearranging:

$$-\frac{\mu}{2} u_{i-1}^{n+1} + (1+\mu) u_i^{n+1} - \frac{\mu}{2} u_{i+1}^{n+1} = \frac{\mu}{2} u_{i-1}^n + (1-\mu) u_i^n + \frac{\mu}{2} u_{i+1}^n,$$

where $\mu = \frac{\alpha \Delta t}{(\Delta x)^2}$.

3.1.1 Consistency of the Crank-Nicholson Method

To prove the consistency of the Crank-Nicholson method, we compare the numerical scheme to the original PDE by substituting the exact solution of the PDE into the scheme and analyzing the resulting error. This error, called the the consistency error, measures the deviation of the numerical scheme from the PDE.

We start with the 1D heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}.$$

Here, $u(x,t)$ *is the exact solution.*

Crank-Nicholson Time Derivative Approximation

The Crank-Nicholson method achieves second-order accuracy in time despite using finite differences that are first-order accurate in time for individual terms. This is because the method averages the forward and backward Euler schemes, which cancels the first-order error and results in a method that is second-order accurate. The key is that the time derivative is approximated using the trapezoidal rule, which is inherently second-order accurate. We focus on the time derivative and the role of the trapezoidal rule.

The Time Derivative of the Heat Equation

We start with the 1D heat equation:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}.$$

We need to approximate the time derivative $\frac{\partial u}{\partial t}$ using finite differences. The Crank-Nicholson method uses the trapezoidal rule for this approximation. Indeed, the Crank-Nicholson scheme is based on the average of the forward and backward Euler methods:

$$\frac{\partial u}{\partial t} \approx \frac{1}{2}\left(\left.\frac{\partial u}{\partial t}\right|_{t^n} + \left.\frac{\partial u}{\partial t}\right|_{t^{n+1}}\right),$$

where: - $\left.\frac{\partial u}{\partial t}\right|_{t^n}$ is the forward difference approximation, - $\left.\frac{\partial u}{\partial t}\right|_{t^{n+1}}$ is the backward difference approximation.

This can be rewritten in terms of the Crank-Nicholson scheme:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha}{2}\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2}\right).$$

Taylor Expansions for Exact Solution

To understand the error, we now expand the exact solution $u(x,t)$ using Taylor expansion about the time levels t^n and t^{n+1}.

- At t^{n+1}:

$$u(x_i, t^{n+1}) = u(x_i, t^n) + \Delta t \left.\frac{\partial u}{\partial t}\right|_{(x_i,t^n)} + \frac{\Delta t^2}{2}\left.\frac{\partial^2 u}{\partial t^2}\right|_{(x_i,t^n)} + O(\Delta t^3).$$

- At t^n:

$$u(x_i, t^n) = u(x_i, t^n).$$

So, the forward difference for the time derivative (using t^{n+1} and t^n) is:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \left.\frac{\partial u}{\partial t}\right|_{(x_i,t^n)} + \frac{\Delta t}{2}\left.\frac{\partial^2 u}{\partial t^2}\right|_{(x_i,t^n)} + O(\Delta t^2).$$

Similarly, for the backward difference approximation (using t^{n+1}):

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \left.\frac{\partial u}{\partial t}\right|_{(x_i,t^{n+1})} - \frac{\Delta t}{2}\left.\frac{\partial^2 u}{\partial t^2}\right|_{(x_i,t^{n+1})} + O(\Delta t^2).$$

The Crank-Nicholson method averages the forward and backward time derivative approximations:

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{1}{2}\left(\left.\frac{\partial u}{\partial t}\right|_{(x_i,t^n)} + \left.\frac{\partial u}{\partial t}\right|_{(x_i,t^{n+1})}\right).$$

Substituting the expansions of the forward and backward approximations into this formula, we get:

$$\frac{u_i^{n+1}-u_i^n}{\Delta t}=\left.\frac{\partial u}{\partial t}\right|_{(x_i,t^n)}+\frac{\Delta t}{2}\left(\left.\frac{\partial^2 u}{\partial t^2}\right|_{(x_i,t^n)}+\left.\frac{\partial^2 u}{\partial t^2}\right|_{(x_i,t^{n+1})}\right)+O(\Delta t^2).$$

Notice that the first-order errors from each of the forward and backward approximations cancel out in the averaging process, leaving a second-order error.

Second-Order Accuracy

Now, the leading error term is proportional to Δt^2, which means that the scheme is second-order accurate in time. This happens because the error from the first-order difference (i.e., $O(\Delta t)$) is effectively canceled out by the averaging, and we are left with a second-order term:

$$\frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2}+O(\Delta t^3).$$

Spatial Derivative Approximation

We expand $u(x_{i+1},t^n)$ and $u(x_{i-1},t^n)$ about x_i:

$$u(x_{i+1},t^n)=u(x_i,t^n)+\Delta x\left.\frac{\partial u}{\partial x}\right|_{(x_i,t^n)}+\frac{\Delta x^2}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^n)}+\frac{\Delta x^3}{6}\left.\frac{\partial^3 u}{\partial x^3}\right|_{(x_i,t^n)}+O(\Delta x^4),$$

$$u(x_{i-1},t^n)=u(x_i,t^n)-\Delta x\left.\frac{\partial u}{\partial x}\right|_{(x_i,t^n)}+\frac{\Delta x^2}{2}\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^n)}-\frac{\Delta x^3}{6}\left.\frac{\partial^3 u}{\partial x^3}\right|_{(x_i,t^n)}+O(\Delta x^4).$$

Using these, the second central difference for the second spatial derivative becomes:

$$\frac{u_{i+1}^n-2u_i^n+u_{i-1}^n}{(\Delta x)^2}=\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^n)}+\frac{\Delta x^2}{12}\left.\frac{\partial^4 u}{\partial x^4}\right|_{(x_i,t^n)}+O(\Delta x^4).$$

Similarly, at t^{n+1}, we have:

$$\frac{u_{i+1}^{n+1}-2u_i^{n+1}+u_{i-1}^{n+1}}{(\Delta x)^2}=\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^{n+1})}+\frac{\Delta x^2}{12}\left.\frac{\partial^4 u}{\partial x^4}\right|_{(x_i,t^{n+1})}+O(\Delta x^4).$$

Combine the Approximations

We Substitute these approximations into the Crank-Nicholson scheme:

$$\left.\frac{\partial u}{\partial t}\right|_{(x_i,t^n)}+\frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}=\frac{\alpha}{2}\left(\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^n)}+\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^{n+1})}\right)+O(\Delta t^2+\Delta x^2).$$

Using the PDE $\frac{\partial u}{\partial t}=\alpha\frac{\partial^2 u}{\partial x^2}$, expand $\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^{n+1})}$ using a Taylor series around t^n:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^{n+1})}=\left.\frac{\partial^2 u}{\partial x^2}\right|_{(x_i,t^n)}+\Delta t\left.\frac{\partial^3 u}{\partial t\partial x^2}\right|_{(x_i,t^n)}+O(\Delta t^2).$$

Substitute this into the equation:

$$\frac{\partial u}{\partial t}+\frac{\Delta t}{2}\frac{\partial^2 u}{\partial t^2}=\alpha\frac{\partial^2 u}{\partial x^2}+\frac{\alpha\Delta t}{2}\frac{\partial^3 u}{\partial t\partial x^2}+O(\Delta t^2+\Delta x^2).$$

The consistency Error

The local truncation error is the deviation of the numerical scheme from the exact PDE. Rearrange the terms:

$$R = O(\Delta t^2) + O(\Delta x^2).$$

This shows that the Crank-Nicholson scheme is second-order accurate in both time and space.

The Crank-Nicholson method is consistent because:

1. *The local truncation error tends to zero as $\Delta t \to 0$ and $\Delta x \to 0$.*

2. *The scheme reproduces the original PDE up to higher-order terms.*

3.1.2 Von Neumann Stability Analysis

Definition 3.1.1. *A numerical scheme is **Von Neumann stable** if, for all permissible values of the numerical parameters (such as time step Δt and spatial step Δx) and all wavenumbers k, the amplification factor g satisfies:*

$$|g| \leq 1,$$

where g measures how a Fourier mode's amplitude changes over time.

If $|g| \leq 1$, errors do not grow uncontrollably, and the scheme is stable. If $|g| > 1$, errors amplify, leading to instability.

Definition 3.1.2. ***Fourier Decomposition:** Solutions to linear PDEs can often be expressed as sums of sinusoidal components (Fourier modes). Each Fourier mode is characterized by a specific wavenumber k and evolves independently under linear systems.*

For example:

$$u_i^n = g^n e^{ikx_i},$$

where u_i^n is the approximate solution at spatial point x_i and time t^n, g is the amplification factor, and k is the wavenumber (spatial frequency of a wave, it describes how many wavelengths fit into a given unit of space).

*2. **Behavior of Errors:** Errors in numerical solutions can also be decomposed into Fourier modes. By analyzing the evolution of each mode separately, we can determine whether the numerical scheme amplifies or dampens these errors.*

*3. **Amplification Factor:** The amplification factor g describes how each Fourier mode evolves between time steps n and $n+1$. For stability, $|g| \leq 1$ ensures that errors in every Fourier mode do not grow.*

Fourier Mode Assumption

Assume the solution takes the form of a Fourier mode:

$$u_i^n = g^n e^{ikx_i},$$

where: - g is the amplification factor, - k is the wavenumber.

The Crank-Nicholson scheme is

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} = \frac{\alpha}{2}\left(\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{(\Delta x)^2} + \frac{u_{i+1}^{n+1} - 2u_i^{n+1} + u_{i-1}^{n+1}}{(\Delta x)^2}\right). \tag{3.1}$$

Substituting the Fourier mode $u_i^n = g^n e^{ikx_i}$ into (3.1), we get

$$u_{i+1}^n = g^n e^{ikx_{i+1}} = g^n e^{ik\Delta x} e^{ikx_i}, \quad u_{i-1}^n = g^n e^{-ik\Delta x} e^{ikx_i}.$$

2. Replace all terms in the scheme. After simplifications:

$$g = \frac{1-\mu(1-\cos(k\Delta x))}{1+\mu(1-\cos(k\Delta x))},$$

where $\mu = \frac{\alpha\Delta t}{(\Delta x)^2}$.

***Magnitude of** g*

To verify stability, compute $|g|$:

$$|g| = \left|\frac{1-A}{1+A}\right|, \quad A = \mu(1-\cos(k\Delta x)).$$

1. The term $1-\cos(k\Delta x) \geq 0$, so $A \geq 0$ for all k. 2. The numerator $1-A$ and denominator $1+A$ are positive, and $|g| \leq 1$.

*then, Since $|g| \leq 1$ for all k and $\mu > 0$, the Crank-Nicholson method is **unconditionally stable.***

Remark 3.1.1. *1. Why Stability Depends on $|g|$: - $|g|$ controls the growth or decay of Fourier modes, which represent the solution and error components. - If $|g| > 1$, errors grow exponentially, leading to instability. - If $|g| \leq 1$, errors are controlled, ensuring a stable computation.*

2. *Crank-Nicholson's Unconditional Stability: - The implicit nature of the Crank-Nicholson scheme balances contributions from time levels n and $n+1$, ensuring that $|g| \leq 1$ regardless of the time step Δt or spatial step Δx.*

Exercise

Show that the Crank–Nicholson scheme (with $\mu = \frac{1}{2}$) is stable in the sense of the L^1 norm if

$$\nu\Delta t \leq (\Delta x)^2.$$

Consider also the DuFort–Frankel scheme:

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} + \nu \frac{-u_{j-1}^n + u_{j+1}^n + u_j^{n-1} - u_j^{n+1}}{(\Delta x)^2} = 0.$$

Show that this scheme is stable if

$$2\nu\Delta t \leq (\Delta x)^2.$$

Correction

Exercise: Crank–Nicolson and DuFort–Frankel stability in L^1

We analyze stability (discrete maximum principle / convex combination argument) for two classical schemes for the heat equation $u_t = \nu u_{xx}$.

Crank–Nicolson (CN)

The Crank–Nicolson scheme (with parameter $\theta = 1/2$) reads

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \nu \frac{1}{2}\Big(\delta_{xx} u_j^{n+1} + \delta_{xx} u_j^n\Big),$$

where $\delta_{xx} u_j^n = \dfrac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2}$.

Rearrange to solve for u_j^{n+1} in terms of u^n. One obtains a three-point relation with coefficients depending on

$$\alpha := \frac{\nu \Delta t}{(\Delta x)^2}.$$

A standard discrete maximum-principle / convex-combination argument shows that if

$$\boxed{\nu \Delta t \le (\Delta x)^2} \quad \textit{(i.e. } \alpha \le 1\textit{)}$$

then the update for u^{n+1} at the grid point of maximal value can only decrease (and similarly the minimal value can only increase). Thus the scheme preserves maxima/minima and is stable in L^1 and L^∞ senses under this condition.

Sketch: *evaluate the CN relation at an index k where $u_k^{n+1} = \max_j u_j^{n+1}$. Using $u_k^{n+1} \ge u_{k\pm1}^{n+1}$ one gets an inequality showing $u_k^{n+1} \le \max(0, \max_j u_j^n)$ provided $\alpha \le 1$; the symmetric argument for minima completes the proof.*

DuFort–Frankel

The DuFort–Frankel explicit scheme (for $u_t = \nu u_{xx}$) writes

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} - \nu \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} = 0,$$

which can be rearranged to

$$u_j^{n+1} = \frac{1 - 2\alpha}{1 + 2\alpha} u_j^{n-1} + \frac{\alpha}{1 + 2\alpha}\big(u_{j+1}^n + u_{j-1}^n\big), \qquad \alpha = \frac{\nu \Delta t}{(\Delta x)^2}.$$

This shows that u_j^{n+1} is a convex combination of u_j^{n-1} and the neighbours $u_{j\pm1}^n$ precisely when the coefficients are nonnegative:

$$\frac{1 - 2\alpha}{1 + 2\alpha} \ge 0 \quad \Longleftrightarrow \quad 2\alpha \le 1,$$

i.e.

$$\boxed{2\nu \Delta t \le (\Delta x)^2}.$$

Under this condition the DuFort–Frankel update is a convex combination and therefore the scheme is stable in L^1 (and L^∞) norms.

Remark 3.1.2. *Crank–Nicolson is implicit and second-order in time; DuFort–Frankel is explicit but uses three time levels and has different dispersion/dissipation properties. The above conditions are the standard stability constraints derived from the convex-combination argument.*

Conclusion

The Crank–Nicholson method is an implicit finite difference scheme that combines the stability of implicit methods with the higher accuracy of centered discretizations. It is unconditionally stable in the L^2 norm for parabolic problems, which makes it particularly suitable for time dependent diffusion type equations. Moreover, being second order accurate in both space and time, it provides a good balance between precision and computational efficiency. However, since it is implicit, it requires solving a linear system at each time step, which may increase the computational cost compared to explicit schemes.

Chapter 4

Finite Volume Method (FVM)

4.1 Introduction

In this chapter, we consider the application of FVM to an elliptic problem, analyze its consistency, and establish its stability.

Definition 4.1.1. ***The Finite Volume Method (FVM)*** *is a numerical technique widely used for solving partial differential equations (PDEs), particularly those arising in conservation laws. This method ensures local conservation by integrating the governing equations over control volumes and applying the divergence theorem.*

4.2 Application to an Elliptic Problem: The 1D Poisson Equation

We consider the one-dimensional Poisson equation:

$$-\frac{d^2u}{dx^2} = f(x), \quad x \in (0,1), \tag{4.1}$$

subject to Dirichlet boundary conditions:

$$u(0) = u_0, \quad u(1) = u_1. \tag{4.2}$$

4.2.1 Finite Volume Discretization

The finite volume method (FVM) is based on integrating the governing equation over discrete control volumes *rather than evaluating it pointwise.*

Definition of Control Volumes

Let the domain $[0,1]$ *be divided into N equally spaced* nodes x_i, $i = 0,1,\ldots,N$, *with spacing*

$$\Delta x = \frac{1}{N}. \tag{4.3}$$

In FVM, we associate a control volume V_i *with each interior node* x_i, *defined as the interval*

$$V_i = [x_{i-1/2}, x_{i+1/2}], \tag{4.4}$$

where the cell faces *are located at the midpoints between nodes:*

$$x_{i+1/2} = \frac{x_i + x_{i+1}}{2}, \quad x_{i-1/2} = \frac{x_{i-1} + x_i}{2}. \tag{4.5}$$

This ensures that each node is at the center *of its control volume, and that control volumes tile the domain without overlap or gaps.*

Integration over a Control Volume

Integrating the Poisson equation over V_i:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} -\frac{d^2u}{dx^2}\,dx = \int_{x_{i-1/2}}^{x_{i+1/2}} f(x)\,dx. \tag{4.6}$$

By the fundamental theorem of calculus (or divergence theorem in 1D), the left-hand side becomes:

$$-\left[\frac{du}{dx}\Big|_{x_{i+1/2}} - \frac{du}{dx}\Big|_{x_{i-1/2}}\right]. \tag{4.7}$$

Approximation of Fluxes

The fluxes $u'(x_{i\pm1/2})$ at the control volume faces are approximated using finite differences. Using central differencing:

$$u'\Big|_{x_{i+1/2}} \approx \frac{u_{i+1} - u_i}{\Delta x}, \quad u'\Big|_{x_{i-1/2}} \approx \frac{u_i - u_{i-1}}{\Delta x}. \tag{4.8}$$

Here, $u_i \approx u(x_i)$ is the approximate solution at node x_i.
Substituting these approximations into the integrated equation gives the discrete scheme:

$$-\left[\frac{u_{i+1} - u_i}{\Delta x} - \frac{u_i - u_{i-1}}{\Delta x}\right] = \Delta x\, f_i, \tag{4.9}$$

or equivalently,

$$\frac{u_{i-1} - 2u_i + u_{i+1}}{\Delta x^2} = f_i, \tag{4.10}$$

where f_i is the cell-average of f over V_i, often approximated as $f(x_i)$.

Assembly of the Linear System

This leads to the classical tridiagonal linear system:

$$A\mathbf{u} = \mathbf{F}, \tag{4.11}$$

where $\mathbf{u} = [u_1, u_2, \ldots, u_{N-1}]^T$ contains the unknown interior values, and $\mathbf{F} = [f_1, f_2, \ldots, f_{N-1}]^T$ includes the right-hand side scaled by Δx^2 and boundary contributions. The matrix $A \in \mathbb{R}^{(N-1)\times(N-1)}$ is tridiagonal:

$$A = \begin{bmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & -1 & \cdots & 0 \\ 0 & -1 & 2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & -1 \\ 0 & 0 & 0 & -1 & 2 \end{bmatrix}. \tag{4.12}$$

Boundary conditions are incorporated by modifying the first and last entries of $\mathbf{F}$*:*

$$F_1 = f_1\Delta x^2 + u_0, \quad F_{N-1} = f_{N-1}\Delta x^2 + u_1. \tag{4.13}$$

Notes on the control volume approach:

1. Each node is at the center of its control volume. 2. Fluxes are evaluated at cell faces, ensuring local conservation. 3. FVM is naturally conservative: the sum of fluxes leaving and entering all control volumes cancels globally. 4. This approach can easily generalize to non-uniform meshes and multidimensional problems.

4.3 Consistency Analysis

Expanding $u(x)$ *in a Taylor series:*

$$u_{i+1} = u(x_i) + \Delta x u'(x_i) + \frac{\Delta x^2}{2}u''(x_i) + O(\Delta x^3), \tag{4.14}$$

$$u_{i-1} = u(x_i) - \Delta x u'(x_i) + \frac{\Delta x^2}{2}u''(x_i) + O(\Delta x^3). \tag{4.15}$$

Adding both equations:

$$u_{i+1} + u_{i-1} = 2u(x_i) + \Delta x^2 u''(x_i) + O(\Delta x^3). \tag{4.16}$$

Rearranging:

$$\frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2} = u''(x_i) + O(\Delta x^2). \tag{4.17}$$

Since the truncation error is $O(\Delta x^2)$*, the scheme is second-order accurate and consistent.*

4.4 Stability Analysis

To analyze stability, we study the properties of the matrix A.

4.4.1 Positive Definiteness

For any nonzero vector v, we have:

$$v^T A v = \sum_{i=1}^{N-1} 2v_i^2 - \sum_{i=1}^{N-2} v_i v_{i+1} - \sum_{i=2}^{N-1} v_i v_{i-1}. \tag{4.18}$$

Using the discrete Fourier transform, it can be shown that all eigenvalues of A are positive, confirming its positive definiteness.

4.4.2 Diagonal Dominance

The matrix A satisfies:

$$|A_{ii}| = 2 > |A_{i,i-1}| + |A_{i,i+1}| = 1 + 1 = 2. \tag{4.19}$$

Since it is weakly diagonally dominant and symmetric positive definite, the system remains stable.

4.4.3 Spectral Properties

The eigenvalues of A are:

$$\lambda_k = 2 - 2\cos\left(\frac{k\pi}{N}\right), \quad k = 1, 2, ..., N-1. \tag{4.20}$$

Since $0 < \lambda_k < 4$*, all eigenvalues are positive, ensuring stability.*

4.5 Conclusion

The finite volume method applied to elliptic problems is consistent with second-order accuracy and is unconditionally stable due to the positive definiteness of the system matrix. These properties guarantee reliable numerical solutions.

4.6 Application to a 2D Elliptic Problem: The Poisson Equation

We consider the two-dimensional Poisson equation on a rectangular domain $\Omega = (0, L_x) \times (0, L_y)$*:*

$$-\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = f(x,y), \quad (x,y) \in \Omega, \tag{4.21}$$

with Dirichlet boundary conditions:

$$u(x,y) = g(x,y), \quad (x,y) \in \partial\Omega. \tag{4.22}$$

4.6.1 Finite Volume Discretization in 2D

Definition of Control Volumes

Divide the domain into N intervals in the x-direction and M intervals in the y-direction:

$$\Delta x = \frac{L_x}{N}, \quad \Delta y = \frac{L_y}{M}. \tag{4.23}$$

Define nodes $x_i = i\Delta x$*,* $i = 0, \ldots, N$ *and* $y_j = j\Delta y$*,* $j = 0, \ldots, M$*, located at the centers of control volumes.*

Each interior control volume is

$$V_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}], \quad i = 1, \ldots, N-1,\ j = 1, \ldots, M-1, \tag{4.24}$$

with faces at

$$x_{i\pm1/2} = \frac{x_i + x_{i\pm1}}{2}, \quad y_{j\pm1/2} = \frac{y_j + y_{j\pm1}}{2}. \tag{4.25}$$

Integration over a Control Volume

Integrate the 2D Poisson equation over $V_{i,j}$:

$$\int_{V_{i,j}} -\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) dA = \int_{V_{i,j}} f(x,y)\, dA. \tag{4.26}$$

By the divergence theorem:

$$-\int_{\partial V_{i,j}} \nabla u \cdot \mathbf{n}\, ds = \int_{V_{i,j}} f(x,y)\, dA, \tag{4.27}$$

where $\mathbf{n}$ *is the outward normal.*

Approximation of Fluxes

Approximate the fluxes at each face using central differences:

$$\textit{East face: } \left.\frac{\partial u}{\partial x}\right|_{i+1/2,j} \approx \frac{u_{i+1,j} - u_{i,j}}{\Delta x}, \tag{4.28}$$

$$\textit{West face: } \left.\frac{\partial u}{\partial x}\right|_{i-1/2,j} \approx \frac{u_{i,j} - u_{i-1,j}}{\Delta x}, \tag{4.29}$$

$$\textit{North face: } \left.\frac{\partial u}{\partial y}\right|_{i,j+1/2} \approx \frac{u_{i,j+1} - u_{i,j}}{\Delta y}, \tag{4.30}$$

$$\textit{South face: } \left.\frac{\partial u}{\partial y}\right|_{i,j-1/2} \approx \frac{u_{i,j} - u_{i,j-1}}{\Delta y}. \tag{4.31}$$

Discrete Scheme

The discrete FVM equation becomes:

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} = f_{i,j}, \quad i = 1, \ldots, N-1,\ j = 1, \ldots, M-1. \tag{4.32}$$

Linear System Definition

Define the unknown vector $\mathbf{u}$ *as all interior nodes:*

$$\mathbf{u} = [u_{1,1}, u_{2,1}, \ldots, u_{N-1,1}, u_{1,2}, \ldots, u_{N-1,M-1}]^T \in \mathbb{R}^{(N-1)(M-1)}. \tag{4.33}$$

Then the discrete system can be written as:

$$A\mathbf{u} = \mathbf{b}, \tag{4.34}$$

where:

- $A \in \mathbb{R}^{(N-1)(M-1)\times(N-1)(M-1)}$ *is* ***sparse block-tridiagonal:***

$$a_{k,k} = \frac{2}{\Delta x^2} + \frac{2}{\Delta y^2}, \tag{4.35}$$

$$a_{k,k\pm1} = -\frac{1}{\Delta x^2} \quad (\textit{east/west neighbors}), \tag{4.36}$$

$$a_{k,k\pm(N-1)} = -\frac{1}{\Delta y^2} \quad (\textit{north/south neighbors}), \tag{4.37}$$

  with $k = i + (j-1)(N-1)$ *in lexicographic ordering.*

- $\mathbf{b} \in \mathbb{R}^{(N-1)(M-1)}$ *contains* $f_{i,j}$ *and contributions from Dirichlet boundaries:*

$$b_k = f_{i,j} + \frac{u_{0,j} \text{ or } u_{N,j}}{\Delta x^2} + \frac{u_{i,0} \text{ or } u_{i,M}}{\Delta y^2} \quad \textit{if the neighbor is on } \partial\Omega. \tag{4.38}$$

Properties

- *A is* ***symmetric positive definite****.*
- *Each interior node interacts only with its four neighbors, giving a* ***5-point stencil****.*
- *Dirichlet boundaries are incorporated in* **b***.*

4.7 Structure of A

In the Finite Volume Method (FVM), the matrix **A** *represents the discretized system of equations derived from the partial differential equation (PDE). The size of* **A** *is determined by the number of unknowns in the system, which corresponds to the number of grid points (or control volumes) in the discretized domain. In this document, we explain how to determine the size of* **A***.*

4.8 Size of Matrix A

4.8.1 Grid Points and Unknowns

For a 2D domain discretized into nx $\times$ *ny control volumes:*

- *Each grid point* (i, j) *corresponds to one unknown* $u_{i,j}$*.*
- *The total number of unknowns is:*

$$N-1 \times M-1.$$

4.8.2 Matrix Dimensions

The matrix **A** *is a square matrix because it represents a system of linear equations where the number of equations equals the number of unknowns. Therefore:*

$$\textit{Size of } \mathbf{A} = (N-1 \times N-1) \times (N-1 \times N-1).$$

4.9 Examples

4.9.1 Example 1: 3×3 Grid

For a 3×3 *grid:*

$$N = 3, \quad M = 3.$$

The total number of unknowns is:

$$N-1 \times M-1 = 2 \times 2 = 4, \textit{no boundary conditions}$$

thus, the size of **A** *is:*

$$\mathbf{A} \textit{ is a } 2 \times 2 \textit{ matrix.}$$

4.9.2 Example 2: 5×4 Grid

For a 5×4 *grid:*

$$N = 5, \quad M = 4.$$

The total number of unknowns is:

$$N-1 \times M-1 \times = 12.$$

Thus, the size of **A** *is:*

$$\mathbf{A} \textit{ is a } 12 \times 12 \textit{ matrix.}$$

4.10 General Rule

For a 2D grid with $N \times M$ *control volumes:*

$$\textit{Size of } \mathbf{A} = (N-1 \times M-1) \times (N-1 \times M-1).$$

The size of the matrix **A** *is determined by the number of grid points* $N \times m$ *in the discretized domain. It is always a square matrix of size* $(N-1 \times M-1) \times (N-1 \times M-1)$*. This rule*

The matrix **A** *is sparse and typically has a block-tridiagonal structure. For a grid with* $(N-1 \times M-1) \times (N-1 \times M-1)$ *points, the matrix can be visualized as follows:*

- *Each row corresponds to a grid point* (i, j)*.*
- *The diagonal block corresponds to the coefficients of* $u_{i,j}$*.*
- *The off-diagonal blocks correspond to the coefficients of the neighboring points* $u_{i-1,j}$*,* $u_{i+1,j}$*,* $u_{i,j-1}$*, and* $u_{i,j+1}$*.*

4.11 Example: Matrix A for a 3×3 Grid

Consider a 4×4 *grid (*$N = 4$*,* $M = 4$*) with* $\Delta x = \Delta y = h$ *and* $k = 0$*. The matrix* **A** *will have the following structure:*

$$\mathbf{A} = \frac{1}{h^2}\begin{bmatrix} -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -4 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -4 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & -4 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & -4 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & -4 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & -4 \end{bmatrix}$$

Here:

- *The diagonal entries are* -4 *(from* $\frac{-2}{h^2}$ *in both x- and y-directions).*
- *The off-diagonal entries* 1 *correspond to the neighboring points.*

4.12 Helmholtz Equation 2D with Finite Volume Method

We consider the two-dimensional Helmholtz equation on the unit square $\Omega = (0,1) \times (0,1)$*:*

$$-\Delta u - k^2 u = f(x,y), \quad (x,y) \in \Omega, \tag{4.39}$$

with Dirichlet boundary conditions:

$$u(x,0) = g_1(x), \quad u(x,1) = g_2(x), \quad u(0,y) = g_3(y), \quad u(1,y) = g_4(y), \tag{4.40}$$

where $\Delta u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}$*,* $k > 0$ *is the wavenumber,* $u(x,y)$ *is the unknown, and* $f(x,y)$ *is a source term.*

4.12.1 Physical Context and Difference from Poisson Equation

- The Helmholtz equation arises from ***time-harmonic wave problems****:*

$$\frac{\partial^2 u}{\partial t^2} = c^2 \Delta u + f(x,y,t), \quad u(x,y,t) = u(x,y)e^{-i\omega t}, \quad k = \frac{\omega}{c}.$$

4.12.2 Finite Volume Discretization

Grid and Control Volumes

Divide Ω *into N intervals along x and M intervals along y:*

$$\Delta x = \frac{1}{N}, \quad \Delta y = \frac{1}{M}.$$

Grid points:

$$x_i = i\Delta x, \quad i = 0,\ldots,N, \quad y_j = j\Delta y, \quad j = 0,\ldots,M.$$

Control volume edges:

$$x_{i+1/2} = x_i + \frac{\Delta x}{2}, \quad x_{i-1/2} = x_i - \frac{\Delta x}{2}, \quad y_{j+1/2} = y_j + \frac{\Delta y}{2}, \quad y_{j-1/2} = y_j - \frac{\Delta y}{2}.$$

Each control volume $V_{i,j}$ is centered at (x_i, y_j), with area $|V_{i,j}| = \Delta x \Delta y$:

$$V_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}].$$

Integration over Control Volume

Integrate (4.39) over $V_{i,j}$:

$$-\int_{V_{i,j}} \Delta u \, dxdy - \int_{V_{i,j}} k^2 u \, dxdy = \int_{V_{i,j}} f(x,y) \, dxdy.$$

Apply the divergence theorem for the Laplacian:

$$-\int_{V_{i,j}} \Delta u \, dxdy = -\int_{\partial V_{i,j}} \nabla u \cdot \mathbf{n} \, ds.$$

Approximate the other terms:

$$\int_{V_{i,j}} k^2 u \, dxdy \approx k^2 u_{i,j} \Delta x \Delta y, \quad \int_{V_{i,j}} f(x,y) \, dxdy \approx f_{i,j} \Delta x \Delta y.$$

Flux Approximation

Using central differences:

$$\begin{aligned} \textit{East:} & \; -\frac{u_{i+1,j} - u_{i,j}}{\Delta x} \Delta y, & \textit{West:} & \; \frac{u_{i,j} - u_{i-1,j}}{\Delta x} \Delta y, \\ \textit{North:} & \; -\frac{u_{i,j+1} - u_{i,j}}{\Delta y} \Delta x, & \textit{South:} & \; \frac{u_{i,j} - u_{i,j-1}}{\Delta y} \Delta x. \end{aligned}$$

Combine contributions:

$$\frac{u_{i-1,j} - 2u_{i,j} + u_{i+1,j}}{\Delta x^2} + \frac{u_{i,j-1} - 2u_{i,j} + u_{i,j+1}}{\Delta y^2} - k^2 u_{i,j} = f_{i,j}. \tag{4.41}$$

4.12.3 Matrix Formulation

Let $\mathbf{u}$ *contain all interior unknowns:*

$$\mathbf{u} = [u_{1,1}, u_{2,1}, \ldots, u_{N-1,1}, u_{1,2}, \ldots, u_{N-1,M-1}]^T \in \mathbb{R}^{(N-1)(M-1)}.$$

The discrete system:

$$A\mathbf{u} = \mathbf{b},$$

- $A \in \mathbb{R}^{(N-1)(M-1)\times(N-1)(M-1)}$ is sparse, block-tridiagonal. - Diagonal: $a_{kk} = -\left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + k^2\right)$. - East/West neighbors: $-\frac{1}{\Delta x^2}$, North/South neighbors: $-\frac{1}{\Delta y^2}$. - $\mathbf{b}$ includes $f_{i,j}$ and boundary contributions.

4.12.4 Example: $N = M = 4$, $k = 2$, $f = 0$

- Interior points: $i = 1,2,3$, $j = 1,2,3$, *total unknowns* $= (N-1)(M-1) = 3 \cdot 3 = 9$. *- Grid spacing:* $\Delta x = \Delta y = 1/4$, $\frac{1}{\Delta x^2} = 16$. *- Diagonal:* $-(32+32+4) = -68$. *- Off-diagonals:* 16 *for neighbors.*

Numbering unknowns row-wise:

$$\mathbf{u} = [u_{1,1}, u_{2,1}, u_{3,1}, u_{1,2}, u_{2,2}, u_{3,2}, u_{1,3}, u_{2,3}, u_{3,3}]^T.$$

Matrix A is 9×9 *and* $\mathbf{b}$ *accounts for boundary values. Solve* $A\mathbf{u} = \mathbf{b}$.

4.12.5 Remarks

1. *Helmholtz differs from Poisson due to* $-k^2u$ *term (wave behavior).*
2. *Dimension of A is* $(N-1)(M-1) \times (N-1)(M-1)$.
3. *Control volumes defined as* $[x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$.
4. *Dirichlet boundaries are included in* $\mathbf{b}$.

4.13 Anisotropic Diffusion Problem

We consider the ***anisotropic diffusion equation*** *in two dimensions:*

$$-\nabla \cdot (\mathbf{D}\nabla u) = f(x,y), \quad (x,y) \in \Omega = (0,1) \times (0,1), \tag{4.42}$$

with Dirichlet boundary conditions:

$$u(x,0) = g_1(x), \quad u(x,1) = g_2(x), \quad u(0,y) = g_3(y), \quad u(1,y) = g_4(y),$$

where the diffusion tensor is diagonal:

$$\mathbf{D} = \begin{bmatrix} D_x & 0 \\ 0 & D_y \end{bmatrix}.$$

Here, D_x *and* D_y *are the diffusion coefficients along the x and y directions, respectively. Anisotropic diffusion appears in many applications: heat conduction in layered materials, groundwater flow in heterogeneous soils, image processing, and transport in biological tissues.*

4.13.1 Control Volumes and Grid

Divide the domain into N intervals along x and M intervals along y:

$$\Delta x = \frac{1}{N}, \quad \Delta y = \frac{1}{M},$$

with grid points:

$$x_i = i\Delta x, \quad i = 0, \ldots, N, \quad y_j = j\Delta y, \quad j = 0, \ldots, M.$$

For ***finite volume discretization****, define control volumes around interior points* (x_i, y_j), $i = 1, \ldots, N-1$, $j = 1, \ldots, M-1$*:*

$$V_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}],$$

with face locations:

$$x_{i\pm1/2} = x_i \pm \frac{\Delta x}{2}, \quad y_{j\pm1/2} = y_j \pm \frac{\Delta y}{2}.$$

Area: $|V_{i,j}| = \Delta x \Delta y$.

4.13.2 Finite Volume Discretization

Integrate (4.42) *over* $V_{i,j}$*:*

$$-\int_{V_{i,j}} \nabla \cdot (\mathbf{D}\nabla u)\, dxdy = \int_{V_{i,j}} f(x,y)\, dxdy.$$

Applying the divergence theorem:

$$-\int_{\partial V_{i,j}} (\mathbf{D}\nabla u) \cdot \mathbf{n}\, ds = \int_{V_{i,j}} f(x,y)\, dxdy.$$

For a diagonal diffusion tensor the fluxes through each face are approximated as:

$$F^E_{i,j} = -D_x \frac{u_{i+1,j} - u_{i,j}}{\Delta x} \Delta y, \qquad F^W_{i,j} = -D_x \frac{u_{i,j} - u_{i-1,j}}{\Delta x} \Delta y,$$
$$F^N_{i,j} = -D_y \frac{u_{i,j+1} - u_{i,j}}{\Delta y} \Delta x, \qquad F^S_{i,j} = -D_y \frac{u_{i,j} - u_{i,j-1}}{\Delta y} \Delta x.$$

Sum the contributions and divide by $\Delta x \Delta y$ *to get the discrete equation at interior point* (i, j)*:*

$$\frac{D_x}{\Delta x^2}(u_{i-1,j} - 2u_{i,j} + u_{i+1,j}) + \frac{D_y}{\Delta y^2}(u_{i,j-1} - 2u_{i,j} + u_{i,j+1}) = f_{i,j}. \tag{4.43}$$

This generalizes the 2D Poisson equation with direction-dependent diffusion coefficients.

4.13.3 Matrix Formulation

Let $\mathbf{u}$ *contain all* ***interior unknowns*** $u_{i,j}$ *for* $i = 1, \ldots, N-1$, $j = 1, \ldots, M-1$*:*

$$\mathbf{u} = [u_{1,1}, u_{2,1}, \ldots, u_{N-1,1}, u_{1,2}, \ldots, u_{N-1,M-1}]^T.$$

- Number of unknowns: $(N-1)(M-1)$*, - Matrix A: block-tridiagonal, size* $(N-1)(M-1) \times (N-1)(M-1)$*, - Coefficients:*

$$a_{i,j;i,j} = -2\frac{D_x}{\Delta x^2} - 2\frac{D_y}{\Delta y^2},$$
$$a_{i,j;i+1,j} = a_{i,j;i-1,j} = \frac{D_x}{\Delta x^2}, \quad a_{i,j;i,j+1} = a_{i,j;i,j-1} = \frac{D_y}{\Delta y^2}.$$

- The vector $\mathbf{b}$ *contains the source* $f_{i,j}$ *and contributions from boundary conditions.*

4.13.4 Numerical Example

Let $N = M = 4$, $D_x = 1$, $D_y = 0.5$, $f = 0$, Dirichlet boundary $u = 0$ on all edges.

- Interior points: $i, j = 1, 2, 3$, total unknowns = $3 \times 3 = 9$. - Grid spacing: $\Delta x = \Delta y = 0.25$ - Coefficients:

$$\frac{D_x}{\Delta x^2} = \frac{1}{0.25^2} = 16, \quad \frac{D_y}{\Delta y^2} = \frac{0.5}{0.25^2} = 8,$$

$$a_{ii} = -2D_x/\Delta x^2 - 2D_y/\Delta y^2 = -(32+16) = -48.$$

- East/West neighbors: 16*, North/South neighbors:* 8*.*

Interior unknown vector (row-wise):

$$\mathbf{u} = [u_{1,1}, u_{2,1}, u_{3,1}, u_{1,2}, u_{2,2}, u_{3,2}, u_{1,3}, u_{2,3}, u_{3,3}]^T.$$

Matrix A explicitly (9×9):

$$A = \begin{bmatrix} -48 & 16 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ 16 & -48 & 16 & 0 & 8 & 0 & 0 & 0 & 0 \\ 0 & 16 & -48 & 0 & 0 & 8 & 0 & 0 & 0 \\ 8 & 0 & 0 & -48 & 16 & 0 & 8 & 0 & 0 \\ 0 & 8 & 0 & 16 & -48 & 16 & 0 & 8 & 0 \\ 0 & 0 & 8 & 0 & 16 & -48 & 0 & 0 & 8 \\ 0 & 0 & 0 & 8 & 0 & 0 & -48 & 16 & 0 \\ 0 & 0 & 0 & 0 & 8 & 0 & 16 & -48 & 16 \\ 0 & 0 & 0 & 0 & 0 & 8 & 0 & 16 & -48 \end{bmatrix},$$

$\mathbf{b} = \mathbf{0}$ *(all zero source and homogeneous Dirichlet boundaries).*

*- Solve using **Gaussian elimination** or **iterative methods** (Jacobi, Gauss-Seidel).*

4.13.5 Remarks

1. *Anisotropy effect: Different diffusion coefficients modify the influence of neighboring points; larger diffusion increases smoothing in that direction.*
2. *Applications: Layered soil groundwater flow, directional heat conduction in materials, anisotropic image filtering.*
3. *Extension: Non-diagonal* $\mathbf{D}$ *introduces cross-derivatives, requiring more complex flux approximations.*

4.14 Finite Volume Method for the Poisson Equation in n Dimensions

The Poisson equation in n-dimensions is:

$$\nabla^2 u = f(x_1, x_2, ..., x_n), \tag{4.44}$$

which expands as:

$$\sum_{j=1}^{n} \frac{\partial^2 u}{\partial x_j^2} = f(x_1, x_2, ..., x_n). \tag{4.45}$$

4.14.1 Discretization Using FVM

Integrating over a control volume V_i:

$$\int_{V_i} \nabla^2 u \, dV = \int_{V_i} f(x) \, dV. \tag{4.46}$$

Applying Gauss's divergence theorem:

$$\oint_{\partial V_i} \nabla u \cdot dS = \int_{V_i} f(x) \, dV. \tag{4.47}$$

Using finite differences to approximate second derivatives:

$$\frac{\partial^2 u}{\partial x_j^2} \approx \frac{u_{i+e_j} - 2u_i + u_{i-e_j}}{h^2}. \tag{4.48}$$

Thus, the discretized Poisson equation in n-dimensions is:

$$\sum_{j=1}^{n} \frac{u_{i+e_j} - 2u_i + u_{i-e_j}}{h^2} = f_i. \tag{4.49}$$

Rearranging:

$$-\sum_{j=1}^{n} u_{i+e_j} - u_{i-e_j} + 2nu_i = h^2 f_i. \tag{4.50}$$

4.14.2 Matrix Formulation

The corresponding matrix system is:

$$Au = b. \tag{4.51}$$

*where: - A is a **sparse matrix** of size $N^n \times N^n$, - u is the unknown solution vector, - b is the right-hand side vector $(h^2 f)$.*

For 2D, A follows a five-point stencil:

$$A = \begin{bmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 4 & -1 & 0 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & -1 & 4 & 0 & 0 & -1 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}.$$

For n-D, the matrix follows a $(2n+1)$-point stencil.

Discretization of the 2D Helmholtz Equation

The 2D Helmholtz equation is:

$$\nabla^2 u + k^2 u = f(x,y)$$

which expands to:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + k^2 u = f(x,y).$$

The key idea in FVM is to integrate the equation over a control volume (CV) surrounding each grid point. Let $\Omega_{i,j}$ be a small control volume centered at (x_i, y_j), with a grid spacing h. Integrating over this control volume:

$$\int_{\Omega_{i,j}} \left(\nabla^2 u + k^2 u\right) dA = \int_{\Omega_{i,j}} f(x,y) dA.$$

*Using the **divergence theorem,** we convert the Laplacian term into a flux integral over the control volume boundary:*

$$\oint_{\partial\Omega_{i,j}} \nabla u \cdot \mathbf{n}\, ds + k^2 \int_{\Omega_{i,j}} u\, dA = \int_{\Omega_{i,j}} f\, dA.$$

This leads to a sparse linear system, which can be solved using iterative methods (e.g., Gauss-Seidel, Multigrid, GMRES).

Remark 4.14.1. *1. The Helmholtz equation is discretized by approximating flux balances at each control volume face.*

2. *The n-Poisson equation includes variable diffusion coefficients, requiring harmonic averaging for accurate flux calculation.*

3. *Both methods lead to sparse linear systems, which can be efficiently solved using iterative techniques.*

4.15 1. Introduction to Hyperbolic Equations

Hyperbolic PDEs describe phenomena with finite propagation speeds, like waves or advection. Unlike the second-order wave equation, the 1D transport equation is a first-order hyperbolic PDE, modeling how a quantity (e.g., density) moves with a constant speed. Today, we'll solve it using the Finite Volume Method (FVM).

4.15.1 The 1D Transport Equation

The 1D transport equation is:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad x \in [0,L], \quad t > 0,$$

where:

- *$u(x,t)$: Quantity being transported (e.g., concentration),*
- *c: Constant transport speed (assume $c > 0$ for rightward motion),*
- *Initial condition: $u(x,0) = u_0(x)$,*
- *Boundary condition: $u(0,t) = g(t)$ (inflow at left boundary).*

The exact solution is $u(x,t) = u_0(x-ct)$, meaning the initial profile shifts right at speed c.

4.15.2 3. Finite Volume Method Basics

FVM conserves quantities by integrating over cells:

- *Divide $[0,L]$ into N cells, $\Delta x = L/N$.*
- *Cell centers: $x_i = (i-0.5)\Delta x$, interfaces: $x_{i-1/2} = (i-1)\Delta x$.*
- *Cell average: $U_i^n \approx \frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t^n)\,dx$.*
- *Update averages using fluxes at cell boundaries.*

Let's apply FVM to the transport equation.

Step 1: Semi-Discrete Form

Integrate over a cell $[x_{i-1/2}, x_{i+1/2}]$:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial u}{\partial t}\,dx + c\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial u}{\partial x}\,dx = 0,$$

- Left: $\frac{d}{dt}\int_{x_{i-1/2}}^{x_{i+1/2}} u\,dx = \Delta x\frac{dU_i}{dt}$, - Right: $c\left[u(x_{i+1/2},t) - u(x_{i-1/2},t)\right]$,

$$\frac{dU_i}{dt} = -\frac{c}{\Delta x}\left[u(x_{i+1/2},t) - u(x_{i-1/2},t)\right].$$

Step 2: Flux Approximation

Define fluxes: $F_{i-1/2} = cu(x_{i-1/2},t)$ (flux entering/leaving the cell). Since $c > 0$, use upwind scheme (value from the left):

$$F_{i-1/2} \approx cU_{i-1}^n, \quad F_{i+1/2} \approx cU_i^n,$$

$$\frac{dU_i}{dt} = -\frac{c}{\Delta x}(U_i^n - U_{i-1}^n).$$

Step 3: Time Discretization

Use explicit Euler:

$$\frac{U_i^{n+1} - U_i^n}{\Delta t} = -\frac{c}{\Delta x}(U_i^n - U_{i-1}^n),$$

$$U_i^{n+1} = U_i^n - \frac{c\Delta t}{\Delta x}(U_i^n - U_{i-1}^n).$$

- Boundary: $U_0^n = g(t^n)$ (inflow), no condition at $x = L$ for $c > 0$.

5. Numerical Scheme and Stability

- Scheme: Upwind, explicit, first-order accurate in space and time. - Stability: CFL condition: $c\frac{\Delta t}{\Delta x} \leq 1$. *Ensures the numerical domain of dependence includes the physical one.*

6. Example Calculation

Take $L = 1$, $c = 1$, $u(x,0) = e^{-20(x-0.5)^2}$ *(Gaussian pulse),* $u(0,t) = 0$, $N = 5$, $\Delta x = 0.2$, $\Delta t = 0.1$.

- Centers: $x_i = 0.1, 0.3, 0.5, 0.7, 0.9$. *- Initial:* $U_1^0 = e^{-20(0.1-0.5)^2} = e^{-3.2} \approx 0.0408$, $U_2^0 = e^{-0.8} \approx 0.4493$, $U_3^0 = e^0 = 1$, $U_4^0 = 0.4493$, $U_5^0 = 0.0408$. *- CFL:* $1 \cdot \frac{0.1}{0.2} = 0.5 \leq 1$ *(stable). - First step* $(i = 3)$*:*

$$U_3^1 = U_3^0 - 0.5(U_3^0 - U_2^0) = 1 - 0.5(1 - 0.4493) = 1 - 0.27535 = 0.72465.$$

7. Practical Exercise

Solve the transport equation:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \quad x \in [0,1], \quad t \in [0,0.4], \tag{4.52}$$

$c = 2$, $u(x,0) = \sin(2\pi x)$, $u(0,t) = 0$.

1. *Divide* $[0,1]$ *into* $N = 4$ *cells. Compute* Δx *and list cell centers. Find* U_i^0.
2. *Set* $\Delta t = 0.05$. *Check CFL. Compute* U_i^1 *for* $i = 2$.
3. *Write the FVM upwind scheme and compute* U_2^2.
4. *Write an Matlab script to solve up to* $t = 0.4$*, plotting u at* $t = 0, 0.2, 0.4$. *Compare with* $u(x,t) = \sin(2\pi(x-2t))$ *for* $x > 2t$*, else 0.*

Solutions to Exercise

1. $N = 4$, $\Delta x = \frac{1}{4} = 0.25$. *Centers:* $x_i = 0.125, 0.375, 0.625, 0.875$. $U_1^0 = \sin(2\pi \cdot 0.125) = \sin(\pi/2) = 1$, $U_2^0 = \sin(3\pi/2) = -1$, $U_3^0 = \sin(5\pi/2) = 1$, $U_4^0 = \sin(7\pi/2) = -1$.
2. *CFL:* $c\frac{\Delta t}{\Delta x} = 2 \cdot \frac{0.05}{0.25} = 0.4 \leq 1$ *(stable).* $U_0^0 = 0$, $U_2^1 = U_2^0 - 0.4(U_2^0 - U_1^0) = -1 - 0.4(-1 - 1) = -1 + 0.8 = -0.2$.
3. *Scheme:* $U_i^{n+1} = U_i^n - \frac{c\Delta t}{\Delta x}(U_i^n - U_{i-1}^n)$. $U_1^1 = 1 - 0.4(1 - 0) = 0.6$, $U_2^2 = U_2^1 - 0.4(U_2^1 - U_1^1) = -0.2 - 0.4(-0.2 - 0.6) = -0.2 + 0.32 = 0.12$.
4.

```
% Matlab script for 1D transport equation with FVM
clear all; close all;
L = 1; c = 2; T = 0.4; N = 20; dx = L/N; dt = 0.05; Nt = T/dt;
x = dx/2:dx:L-dx/2;
U = sin(2*pi*x); % Initial condition
U_hist = [U]; times = [0];
```

```
nu = c*dt/dx;
for n = 1:Nt
    U_new = U;
    for i = 2:N
        U_new(i) = U(i) - nu * (U(i) - U(i-1));
    end
    U_new(1) = 0; % Boundary condition
    t = n*dt;
    if abs(t - 0.2) < dt/2 || abs(t - 0.4) < dt/2
        U_hist = [U_hist; U_new]; times = [times, t];
    end
    U = U_new;
end
figure; plot(x, U_hist(1,:), 'b-', 'LineWidth', 1.5, 'DisplayName', 't=0');
hold on; plot(x, U_hist(2,:), 'r-', 'LineWidth', 1.5, 'DisplayName', 't=0.2');
plot(x, U_hist(3,:), 'g-', 'LineWidth', 1.5, 'DisplayName', 't=0.4');
exact = max(sin(2*pi*(x - 2*0.4)), 0.*(x < 2*0.4));
plot(x, exact, 'k--', 'LineWidth', 1.5, 'DisplayName', 'Exact t=0.4');
xlabel('x'); ylabel('u(x,t)'); legend; grid on; hold off;
```

The FVM solution shifts the sine wave right, matching $u(x,t) = \sin(2\pi(x-2t))$ until it exits at $x = 1$.

4.16 Solving the 1D Wave Equation Using Finite Volume Method

Problem Statement

We need to solve the 1D wave equation:

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

where:

- $u(x,t)$*: Wave displacement.*
- $c = 1$*: Wave speed.*
- *Domain:* $x \in [0,1]$, $t \in [0,0.5]$.
- *Initial conditions:* $u(x,0) = \sin(\pi x)$, $\frac{\partial u}{\partial t}(x,0) = 0$.
- *Boundary conditions:* $u(0,t) = u(1,t) = 0$.
- *Grid:* $N = 5$ *cells, CFL = 0.5.*

The exact solution is:

$$u(x,t) = \sin(\pi x)\cos(\pi t)$$

The one-dimensional wave equation is a fundamental partial differential equation (PDE) used to model wave propagation phenomena, such as sound waves or vibrations. It is given by:

$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \tag{4.53}$$

where $u(x,t)$ is the wave displacement, c is the constant wave speed, $x \in [0,L]$ is the spatial coordinate, and $t \geq 0$ is time. In this document, we apply the Finite Volume Method (FVM) to derive a numerical scheme for solving this equation without transforming it into a first-order system. We also provide a detailed stability analysis and discuss convergence conditions.

4.17 Domain Discretization

4.17.1 Spatial Discretization

We discretize the spatial domain $[0,L]$ into N cells (control volumes):

- ***Cell centers**: $x_i = (i-0.5)\Delta x$, where $\Delta x = \frac{L}{N}$ and $i = 1,2,\ldots,N$.*
- ***Cell interfaces**: $x_{i-1/2} = (i-1)\Delta x$, $x_{i+1/2} = i\Delta x$.*
- *Each cell is the interval $[x_{i-1/2}, x_{i+1/2}]$, with length Δx.*

4.17.2 Temporal Discretization

We discretize time into steps $t^n = n\Delta t$, where Δt is the time step, and $n = 0,1,2,\ldots$.

4.17.3 Cell Averages

In FVM, we work with the cell average of the solution:

$$u_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t^n)\, dx. \tag{4.54}$$

Our goal is to compute u_i^{n+1} from u_i^n.

4.18 Derivation of the Numerical Scheme

4.18.1 Integrate Over the Control Volume

Integrate the wave equation over the control volume $[x_{i-1/2}, x_{i+1/2}]$:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial t^2}\, dx = c^2 \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial x^2}\, dx. \tag{4.55}$$

***Left-hand side**:*

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial t^2}\, dx = \frac{\partial^2}{\partial t^2} \int_{x_{i-1/2}}^{x_{i+1/2}} u\, dx = \frac{\partial^2}{\partial t^2}(\Delta x u_i) = \Delta x \frac{\partial^2 u_i}{\partial t^2}, \tag{4.56}$$

assuming $u_i(t)$ approximates the cell average.

***Right-hand side**:*

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial x^2}\, dx = \left.\frac{\partial u}{\partial x}\right|_{x_{i-1/2}}^{x_{i+1/2}} = \frac{\partial u}{\partial x}(x_{i+1/2},t) - \frac{\partial u}{\partial x}(x_{i-1/2},t). \tag{4.57}$$

Thus:

$$\Delta x \frac{\partial^2 u_i}{\partial t^2} = c^2 \left(\frac{\partial u}{\partial x}(x_{i+1/2}, t) - \frac{\partial u}{\partial x}(x_{i-1/2}, t) \right). \tag{4.58}$$

Divide by Δx:

$$\frac{\partial^2 u_i}{\partial t^2} = \frac{c^2}{\Delta x} \left(\frac{\partial u}{\partial x}(x_{i+1/2}, t) - \frac{\partial u}{\partial x}(x_{i-1/2}, t) \right). \tag{4.59}$$

4.18.2 Approximate the Spatial Derivatives

Approximate the fluxes at the interfaces using central differences:

- *At $x_{i+1/2}$:*

$$\frac{\partial u}{\partial x}(x_{i+1/2}, t) \approx \frac{u_{i+1}(t) - u_i(t)}{\Delta x}. \tag{4.60}$$

- *At $x_{i-1/2}$:*

$$\frac{\partial u}{\partial x}(x_{i-1/2}, t) \approx \frac{u_i(t) - u_{i-1}(t)}{\Delta x}. \tag{4.61}$$

Substitute:

$$\frac{\partial^2 u_i}{\partial t^2} = \frac{c^2}{\Delta x} \left(\frac{u_{i+1}(t) - u_i(t)}{\Delta x} - \frac{u_i(t) - u_{i-1}(t)}{\Delta x} \right) = \frac{c^2}{\Delta x^2} (u_{i+1}(t) - 2u_i(t) + u_{i-1}(t)). \tag{4.62}$$

4.18.3 Discretize the Time Derivative

Discretize the second time derivative using a central difference:

$$\left. \frac{\partial^2 u_i}{\partial t^2} \right|_{t^n} \approx \frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2}. \tag{4.63}$$

Substitute into the equation:

$$\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2} = \frac{c^2}{\Delta x^2} (u_{i+1}^n - 2u_i^n + u_{i-1}^n). \tag{4.64}$$

Multiply by Δt^2:

$$u_i^{n+1} - 2u_i^n + u_i^{n-1} = \left(\frac{c\Delta t}{\Delta x} \right)^2 (u_{i+1}^n - 2u_i^n + u_{i-1}^n). \tag{4.65}$$

Define the Courant number $\lambda = \frac{c\Delta t}{\Delta x}$, so the scheme becomes:

$$u_i^{n+1} = \lambda^2 (u_{i+1}^n - 2u_i^n + u_{i-1}^n) + 2u_i^n - u_i^{n-1}. \tag{4.66}$$

Simplify:

$$u_i^{n+1} = \lambda^2 u_{i+1}^n + (2 - 2\lambda^2) u_i^n + \lambda^2 u_{i-1}^n - u_i^{n-1}. \tag{4.67}$$

4.19 Initial and Boundary Conditions

4.19.1 Initial Conditions

We need two initial conditions:

- $u(x,0) = f(x)$, *so* $u_i^0 = f(x_i)$.
- $\frac{\partial u}{\partial t}(x,0) = g(x)$. *To compute* u_i^1, *use the scheme at* $n = 0$*:*

$$u_i^1 = \lambda^2(u_{i+1}^0 - 2u_i^0 + u_{i-1}^0) + 2u_i^0 - u_i^{-1}. \tag{4.68}$$

Since u_i^{-1} *is not defined, approximate it:*

$$u_i^{-1} \approx u_i^0 - \Delta t g(x_i), \tag{4.69}$$

so:

$$u_i^1 = \lambda^2(u_{i+1}^0 - 2u_i^0 + u_{i-1}^0) + u_i^0 + \Delta t g(x_i). \tag{4.70}$$

4.19.2 Boundary Conditions

For a finite domain, apply Dirichlet conditions, e.g., $u(0,t) = 0$, $u(L,t) = 0$, *so* $u_1^n = 0$, $u_N^n = 0$.

4.20 Stability Analysis

We perform a detailed von Neumann stability analysis to determine the conditions under which the scheme is stable.

4.20.1 Von Neumann Analysis

Assume a solution of the form:

$$u_i^n = \xi^n e^{ikx_i}, \tag{4.71}$$

where k *is the wave number,* ξ *is the amplification factor, and* $i = \sqrt{-1}$. *Substitute into the scheme:*

$$\xi^{n+1} e^{ikx_i} = \lambda^2 e^{ikx_{i+1}} \xi^n + (2 - 2\lambda^2) e^{ikx_i} \xi^n + \lambda^2 e^{ikx_{i-1}} \xi^n - e^{ikx_i} \xi^{n-1}. \tag{4.72}$$

Divide by $e^{ikx_i} \xi^{n-1}$*:*

$$\frac{\xi^{n+1}}{\xi^{n-1}} = \lambda^2 \frac{\xi^n}{\xi^{n-1}} e^{ik\Delta x} + (2 - 2\lambda^2) \frac{\xi^n}{\xi^{n-1}} + \lambda^2 \frac{\xi^n}{\xi^{n-1}} e^{-ik\Delta x} - 1. \tag{4.73}$$

Let $\eta = \frac{\xi^n}{\xi^{n-1}}$, *so* $\frac{\xi^{n+1}}{\xi^{n-1}} = \eta^2$. *Then:*

$$\eta^2 = \eta \left(\lambda^2 (e^{ik\Delta x} + e^{-ik\Delta x}) + (2 - 2\lambda^2) \right) - 1. \tag{4.74}$$

Use $e^{ik\Delta x} + e^{-ik\Delta x} = 2\cos(k\Delta x)$*:*

$$\eta^2 = \eta \left(2\lambda^2 \cos(k\Delta x) + 2 - 2\lambda^2 \right) - 1. \tag{4.75}$$

Let $\beta = \cos(k\Delta x)$, *so:*

$$\eta^2 - 2\eta \left(1 + \lambda^2(\beta - 1) \right) + 1 = 0. \tag{4.76}$$

Solve for η*:*

$$\eta = 1 + \lambda^2(\beta - 1) \pm \sqrt{(1 + \lambda^2(\beta - 1))^2 - 1}. \tag{4.77}$$

4.20.2 Stability Condition

For stability, $|\eta| \leq 1$. Since $\beta \in [-1,1]$, evaluate at the extremes:

- ***Case 1:*** $\beta = 1$*:*

$$\eta = 1 + \lambda^2(1-1) \pm \sqrt{(1+0)^2 - 1} = 1, \tag{4.78}$$

which satisfies $|\eta| \leq 1$.

- ***Case 2:*** $\beta = -1$*:*

$$\eta = 1 + \lambda^2(-2) \pm \sqrt{(1-2\lambda^2)^2 - 1}. \tag{4.79}$$

Compute the discriminant:

$$(1-2\lambda^2)^2 - 1 = 1 - 4\lambda^2 + 4\lambda^4 - 1 = 4\lambda^2(\lambda^2 - 1). \tag{4.80}$$

So:

$$\eta = 1 - 2\lambda^2 \pm 2\sqrt{\lambda^2(\lambda^2 - 1)}. \tag{4.81}$$

Consider two cases for λ^2:

- ***If*** $\lambda^2 \leq 1$ *(i.e., $\lambda^2 - 1 \leq 0$):*

$$\sqrt{\lambda^2(\lambda^2 - 1)} = \sqrt{\lambda^2(1-\lambda^2)}i,$$

$$\eta = 1 - 2\lambda^2 \pm 2i\sqrt{\lambda^2(1-\lambda^2)}.$$

Compute the magnitude:

$$|\eta|^2 = (1-2\lambda^2)^2 + (2\sqrt{\lambda^2(1-\lambda^2)})^2 = 1 - 4\lambda^2 + 4\lambda^4 + 4\lambda^2 - 4\lambda^4 = 1.$$

Thus, $|\eta| = 1$, indicating the scheme is stable (but not strictly stable, as there's no damping).

- ***If*** $\lambda^2 > 1$*:*

$$\eta = 1 - 2\lambda^2 \pm 2\sqrt{\lambda^2(\lambda^2 - 1)}.$$

For $\lambda^2 > 1$, $\sqrt{\lambda^2(\lambda^2-1)} > 0$. Consider the larger root:

$$\eta_+ = 1 - 2\lambda^2 + 2\sqrt{\lambda^2(\lambda^2 - 1)}.$$

For example, if $\lambda^2 = 2$:

$$\eta_+ = 1 - 4 + 2\sqrt{2(2-1)} = -3 + 2\sqrt{2} \approx -0.1716,$$

which satisfies $|\eta| < 1$, but we need to check all β. The smaller root:

$$\eta_- = 1 - 2\lambda^2 - 2\sqrt{\lambda^2(\lambda^2 - 1)} = -3 - 2\sqrt{2} \approx -5.828,$$

so $|\eta_-| > 1$, indicating instability.

The scheme is stable only when:

$$\frac{c\Delta t}{\Delta x} \leq 1 \quad or \quad \Delta t \leq \frac{\Delta x}{c}. \tag{4.82}$$

This is the ***Courant-Friedrichs-Lewy (CFL) condition****.*

4.20.3 Amplification Factor Behavior

When $\lambda \leq 1$, $|\eta| = 1$, meaning the scheme preserves the amplitude of the solution, which is expected for the wave equation (a hyperbolic PDE with no dissipation). However, if $\lambda > 1$, the amplification factor can exceed 1, leading to exponential growth of errors and instability.

4.21 Convergence Conditions

4.21.1 Consistency

- *Spatial discretization: $\frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{\Delta x^2}$ approximates $\frac{\partial^2 u}{\partial x^2}$ with error $O(\Delta x^2)$.*
- *Temporal discretization: $\frac{u_i^{n+1} - 2u_i^n + u_i^{n-1}}{\Delta t^2}$ approximates $\frac{\partial^2 u}{\partial t^2}$ with error $O(\Delta t^2)$.*
- *The local truncation error is $O(\Delta t^2 + \Delta x^2)$, so the scheme is consistent.*

4.21.2 Stability

The scheme is stable under the CFL condition:

$$\frac{c\Delta t}{\Delta x} \leq 1. \tag{4.83}$$

4.21.3 Convergence

By the Lax-Richtmyer equivalence theorem, a consistent and stable scheme for a linear PDE converges. Thus, the scheme converges as $\Delta t, \Delta x \to 0$, provided the CFL condition holds.

4.22 Final Numerical Scheme

The numerical scheme is:

$$u_i^{n+1} = \lambda^2 u_{i+1}^n + (2 - 2\lambda^2)u_i^n + \lambda^2 u_{i-1}^n - u_i^{n-1}, \tag{4.84}$$

where $\lambda = \frac{c\Delta t}{\Delta x}$. For the first time step:

$$u_i^1 = \lambda^2 (u_{i+1}^0 - 2u_i^0 + u_{i-1}^0) + u_i^0 + \Delta t\, g(x_i). \tag{4.85}$$

Conclusion

*The **Finite Volume Method (FVM)** is a numerical approach used to solve partial differential equations (PDEs) by dividing the computational domain into small subdomains called control volumes. It is widely applied in engineering, particularly for solving problems governed by conservation laws such as fluid flow, heat transfer, and mass transport.*

4.22.1 Core Concept

The primary idea of FVM is to enforce the conservation of quantities (such as mass, momentum, or energy) within each control volume. The method integrates the governing PDE over each control volume and applies the divergence theorem to transform volume integrals into surface integrals. This ensures that the flux of a conserved quantity through the boundaries of each control volume is accurately accounted for.

Chapter 5

The Finite Element Method (FEM)

"In this chapter, we apply the finite element method (FEM) to approximate the solution of the heat equation.

Definition 5.0.1. *The finite element method (FEM) is a versatile computational approach widely used to approximate the solutions of partial differential equations (PDEs). Its strength lies in transforming the original problem into a weak formulation, which is then discretized using suitable basis functions.*

Key Advantages:

- *Handles irregular geometries and boundary conditions efficiently*
- *Well-suited for higher-dimensional problems*
- *Flexible in choosing the type of basis functions*

The core idea is to break the domain into small subdomains (elements), construct local approximations (basis functions), and then assemble a global system to approximate the PDE.

1. Introduction to the Heat Equation

The one-dimensional heat equation models the distribution of temperature over time in a thin, insulated rod:

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad x \in (0,L), t > 0$$

where:

- $u(x,t)$*: temperature at position x and time t*
- α*: thermal diffusivity of the material*

With initial condition:

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

And boundary conditions (e.g., Dirichlet):

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

2. Weak Formulation (Variational Form)

To apply FEM, we first convert the PDE into a weak (variational) form.

1. *Multiply both sides by a test function* $v(x) \in H_0^1(0,L)$*:*

$$\int_0^L \frac{\partial u}{\partial t} v(x)\,dx = \alpha \int_0^L \frac{\partial^2 u}{\partial x^2} v(x)\,dx$$

2. *Apply integration by parts on the right-hand side:*

$$\int_0^L \frac{\partial u}{\partial t} v\,dx = -\alpha \int_0^L \frac{\partial u}{\partial x}\frac{\partial v}{\partial x}\,dx$$

Boundary terms vanish since $v(0) = v(L) = 0$*.*

3. Spatial Discretization Using FEM

3.1 Mesh and Basis Functions

- *Divide* $[0,L]$ *into N equal subintervals of length* $h = L/N$
- *Nodes:* $x_0 = 0, x_1, ..., x_N = L$
- *Use piecewise linear basis functions* $\{\phi_i(x)\}$ *with local support:*

$$\phi_i(x_j) = \delta_{ij} \quad \textit{(Kronecker delta)}$$

3.2 Approximate the Solution

We approximate:

$$u(x,t) \approx u_h(x,t) = \sum_{j=1}^{N-1} U_j(t)\phi_j(x)$$

Insert this into the weak form and test against ϕ_i*:*

$$\sum_j \left(\int_0^L \phi_j \phi_i dx \right) \frac{dU_j}{dt} = -\alpha \sum_j \left(\int_0^L \phi_j' \phi_i' dx \right) U_j$$

10. Expansion Using Basis Functions and Substitution into the Weak Form

We approximate the solution $u(x,t)$ *by a linear combination of basis functions:*

$$u_h(x,t) = \sum_{j=1}^{N-1} U_j(t)\phi_j(x),$$

where:

- $U_j(t)$ *are the time-dependent coefficients (unknowns),*

- $\phi_j(x)$ *are the basis functions, typically piecewise linear (hat functions).*

Let us choose a test function $v_h = \phi_i(x)$ *for each* $i = 1, \ldots, N-1$ *and substitute the approximation into the weak formulation:*

$$\int_0^L \frac{\partial u_h}{\partial t}\phi_i\,dx + \alpha \int_0^L \frac{\partial u_h}{\partial x}\frac{\partial \phi_i}{\partial x}\,dx = 0$$

Now, using the fact that:

$$\frac{\partial u_h}{\partial t} = \sum_{j=1}^{N-1} \frac{dU_j(t)}{dt}\phi_j(x),$$

$$\frac{\partial u_h}{\partial x} = \sum_{j=1}^{N-1} U_j(t)\frac{d\phi_j}{dx}(x),$$

we substitute both into the weak form:

$$\int_0^L \left(\sum_{j=1}^{N-1} \frac{dU_j}{dt}\phi_j\right)\phi_i\,dx + \alpha \int_0^L \left(\sum_{j=1}^{N-1} U_j \frac{d\phi_j}{dx}\right)\frac{d\phi_i}{dx}\,dx = 0$$

By linearity of the integral, we can take the summation outside:

$$\sum_{j=1}^{N-1} \frac{dU_j}{dt}\underbrace{\int_0^L \phi_j\phi_i\,dx}_{=M_{ij}} + \alpha \sum_{j=1}^{N-1} U_j \underbrace{\int_0^L \frac{d\phi_j}{dx}\frac{d\phi_i}{dx}\,dx}_{=K_{ij}} = 0$$

This results in the algebraic system:

$$\sum_{j=1}^{N-1} M_{ij}\frac{dU_j}{dt} + \alpha \sum_{j=1}^{N-1} K_{ij}U_j = 0 \quad \textit{for } i = 1, \ldots, N-1$$

Or in matrix form:

$$M\frac{d\mathbf{U}}{dt} + \alpha K\mathbf{U} = 0,$$

where:

- M *is the mass matrix,*
- K *is the stiffness matrix,*
- $\mathbf{U}(t) = [U_1(t), \ldots, U_{N-1}(t)]^T$.

3. Spatial Discretization Using FEM

3.1 Mesh and Basis Functions

- *Divide* $[0, L]$ *into* N *equal subintervals of length* $h = L/N$
- *Nodes:* $x_0 = 0, x_1, ..., x_N = L$
- *Use piecewise linear basis functions* $\{\phi_i(x)\}$ *with local support:*

$$\phi_i(x_j) = \delta_{ij} \quad \textit{(Kronecker delta)}$$

3.2 Approximate the Solution

We approximate:

$$u(x,t) \approx u_h(x,t) = \sum_{j=1}^{N-1} U_j(t)\phi_j(x)$$

Insert this into the weak form and test against ϕ_i:

$$\sum_j \left(\int_0^L \phi_j \phi_i dx \right) \frac{dU_j}{dt} = -\alpha \sum_j \left(\int_0^L \phi_j' \phi_i' dx \right) U_j$$

4. Matrix Formulation

Let:

- ***Mass matrix*** *M, with entries:* $M_{ij} = \int_0^L \phi_i(x)\phi_j(x)dx$
- ***Stiffness matrix*** *K, with entries:* $K_{ij} = \int_0^L \phi_i'(x)\phi_j'(x)dx$

Then the ODE system becomes:

$$M\frac{d\mathbf{U}}{dt} + \alpha K\mathbf{U} = 0$$

where $\mathbf{U}(t) = [U_1(t), ..., U_{N-1}(t)]^T$

5. Time Discretization

5.1 Forward Euler (Explicit)

$$M\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \alpha K\mathbf{U}^n = 0$$

Solve for $\mathbf{U}^{n+1}$.

5.2 Backward Euler (Implicit)

$$M\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \alpha K\mathbf{U}^{n+1} = 0$$

Solve a linear system at each time step.

5.3 Crank-Nicolson (Semi-implicit)

$$M\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} + \alpha K\left(\frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2}\right) = 0$$

Better stability and second-order accurate in time.

6. Implementation Steps

1. *Generate Mesh: subdivide* $[0,L]$ *into N elements*
2. *Define Basis Functions: use hat functions on each element*
3. *Assemble Matrices: compute M and K*
4. *Apply Initial Condition: project* $f(x)$ *onto basis*
5. *Time Loop: advance using chosen scheme (Euler, Crank-Nicolson)*
6. *Output: visualize* $u(x,t)$ *over time*

9. Galerkin Approximation Method

The Galerkin method is a standard approach used within the finite element framework. It consists of choosing both the trial (approximation) and test functions from the same finite-dimensional subspace of $H_0^1(0,L)$.

We seek an approximate solution $u_h(x,t) = \sum_{j=1}^{N-1} U_j(t)\phi_j(x)$ *such that:*

$$\int_0^L \frac{\partial u_h}{\partial t} v_h \, dx + \alpha \int_0^L \frac{\partial u_h}{\partial x} \frac{\partial v_h}{\partial x} \, dx = 0 \quad \forall v_h \in V_h \subset H_0^1(0,L)$$

Here, V_h *is the space spanned by the basis functions* $\{\phi_1, \ldots, \phi_{N-1}\}$.

This leads directly to the matrix formulation described in Section 4. The Galerkin formulation ensures consistency and stability under appropriate conditions and is suitable for deriving energy estimates and convergence proofs.

10. Computation of the Mass and Stiffness Matrices

We now compute the entries of the mass and stiffness matrices for piecewise linear basis functions.

On each element $[x_i, x_{i+1}]$ *of size h, the local basis functions* ϕ_i *and* ϕ_{i+1} *are defined such that:*

- $\phi_i(x)$ *is linear and satisfies* $\phi_i(x_i) = 1$, $\phi_i(x_{i+1}) = 0$
- $\phi_{i+1}(x)$ *is linear and satisfies* $\phi_{i+1}(x_i) = 0$, $\phi_{i+1}(x_{i+1}) = 1$

The expressions are:

$$\phi_i(x) = \frac{x_{i+1} - x}{h}, \quad \phi_{i+1}(x) = \frac{x - x_i}{h}$$

Their derivatives are:

$$\phi_i'(x) = -\frac{1}{h}, \quad \phi_{i+1}'(x) = \frac{1}{h}$$

Local Mass Matrix

We compute:

$$M_{ii}^{(e)} = \int_{x_i}^{x_{i+1}} \left(\frac{x_{i+1}-x}{h}\right)^2 dx = \frac{h}{3}, \quad M_{i,i+1}^{(e)} = \int_{x_i}^{x_{i+1}} \frac{x_{i+1}-x}{h} \cdot \frac{x-x_i}{h} dx = \frac{h}{6}$$

Hence,

$$M^{(e)} = \frac{h}{6}\begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$$

Local Stiffness Matrix

Since the derivatives are constants:

$$K_{ii}^{(e)} = \int_{x_i}^{x_{i+1}} \left(-\frac{1}{h}\right)^2 dx = \frac{1}{h}, \quad K_{i,i+1}^{(e)} = \int_{x_i}^{x_{i+1}} \left(-\frac{1}{h}\right)\left(\frac{1}{h}\right) dx = -\frac{1}{h}$$

So the matrix is:

$$K^{(e)} = \frac{1}{h}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

These local matrices are assembled into the global matrices M and K by summing contributions from all elements and placing their values into the correct positions in the global system.

Conclusion

*The **Finite Element Method (FEM)** is a powerful numerical technique used to approximate solutions to PDEs and integral equations. It is particularly popular in engineering and applied mathematics for analyzing problems involving complex geometries, material properties, and boundary conditions.*

The method divides the computational domain into small, simple subdomains called elements (e.g., triangles, quadrilaterals, or tetrahedra). The PDE is reformulated in its weak form (using variational principles), and the solution is approximated by piecewise-defined basis functions over the elements.

FEM reduces the problem to a system of algebraic equations, where the unknowns are typically the values of the solution at specific points (nodes) within the domain.

Chapter 6

Numerical Simulation

*In this chapter, we focus on the numerical study of several classes of partial differential equations, including **elliptic**, **parabolic**, and the **heat equation**. To approximate their solutions, we introduce and apply different numerical schemes: the **finite difference method**, the **finite volume method**, and the **finite element method**. These approaches will allow us to analyze both the theoretical aspects of stability and convergence, as well as practical implementations through examples and simulations.*

6.1 Finite difference discretization of the Transport equation

We consider the linear transport equation on a one-dimensional domain:

$$u_t + c\,u_x = 0, \qquad x \in [0,L],\ t \geq 0,$$

with initial condition $u(0,x) = u_0(x)$. Let Δx and Δt be the spatial and temporal steps. Define the grid

$$x_i = i\Delta x, \quad i = 0,1,\ldots,N_x - 1, \qquad t^n = n\Delta t, \quad n = 0,1,\ldots,$$

*and denote the numerical approximation by $u_i^n \approx u(t^n, x_i)$. We define the **CFL** condition as follows*

$$k \ := \ \frac{c\,\Delta t}{\Delta x}.$$

6.1.1 Two explicit finite-difference schemes

(A) Left (upwind) scheme — suitable for $c > 0$. *Use a forward difference (right approximation) in time and a backward difference (left approximation) in space:*

$$\frac{u_i^{n+1} - u_i^n}{\Delta t} + c\frac{u_i^n - u_{i-1}^n}{\Delta x} = 0.$$

Multiplying by Δt and rearranging gives

$$\boxed{u_i^{n+1} \ = \ u_i^n - k\,(u_i^n - u_{i-1}^n)\,.} \tag{6.1}$$

This is the standard first-order upwind scheme for $c > 0$. Written as a convex combination,

$$u_i^{n+1} = (1-k)\,u_i^n + k\,u_{i-1}^n,$$

which is monotone when $0 \leq k \leq 1$.

(B) Right (downwind) scheme unstable for $c > 0$. *Use forward time and forward space difference:*

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} + c\frac{u_{j+1}^n - u_j^n}{\Delta x} = 0,$$

which yields

$$\boxed{u_j^{n+1} = u_j^n - k\,(u_{j+1}^n - u_j^n)\,.} \tag{6.2}$$

For $c > 0$ this is the downwind scheme; it is not monotone and is generally unstable (see the stability discussion below).

Thus for $c > 0$ the recommended explicit choice for covergence of the upwind scheme with $k \leq 1$.

6.1.2 Exact shift when $k = 1$ and grid-aligned discontinuity

An important special case occurs when

$$k = \frac{c\Delta t}{\Delta x} = 1.$$

Then the upwind update (6.1) reduces to

$$u_j^{n+1} = u_{j-1}^n,$$

i.e. the solution is shifted by exactly one grid cell at each time step. If the initial condition is a pure step aligned with grid points, for example

$$u_0(x) = \begin{cases} 0, & x < x_s, \\ 1, & x \geq x_s, \end{cases}$$

and x_s coincides with a grid node (so $x_s = i_s\Delta x$ for some integer i_s), then with $k = 1$ the numerical solution remains binary (only values 0 or 1) for all time: the discontinuity is translated exactly without smearing. This explains how one can obtain exact 0/1 results for the step initial data in the discrete scheme.

6.1.3 Boundary conditions (practical remarks)

- ***Periodic BCs:*** *use cyclic indexing, e.g. $u_{-1} = u_{N_x-1}$ and $u_{N_x} = u_0$.*
- ***Inflow/Outflow (nonperiodic):*** *impose the inflow value at the left boundary for $c > 0$: set $u_0^n = u_{in}(t^n)$. For the rightmost point use the upwind formula with the available left neighbor; take care when the discontinuity leaves the domain.*
- ***Grid alignment for exact 0/1:*** *if you require purely 0/1 numerics for a step at x_s, choose Δx so that $x_s/\Delta x$ is integer and choose $\Delta t = \Delta x/c$ (thus $k = 1$).*

The Matlab code is written as follows

```
% transport_fd_comparison.m
% Compare left (upwind) and right (downwind) explicit FD schemes
clear; close all; clc
```

```
% -------------------------
% Parameters
% -------------------------
c = 1.0;            % advection speed (c>0)
L = 1.0;            % domain length [0,L]
Nx = 200;           % number of spatial cells
dx = L/Nx;
x = (0:Nx-1)'*dx;   % cell centers (periodic)
T = 0.5;            % final time

% Choose initial data: 'gauss' or 'step'
init_type = 'gauss'; % 'gauss' or 'step'

% initial condition
switch init_type
    case 'gauss'
        eps = 1e-2;
        u0 = exp(-(x-0.5).^2/eps);
    case 'step'
        u0 = double(x <= 0.5);
    otherwise
        error('Unknown init_type');
end

% CFL numbers to test
k_values = [0.2, 0.5, 0.9, 1.1];   % includes a >1 case to show instability
colors = lines(length(k_values));

figure('Name','Snapshots: upwind (left) scheme');
figure('Name','Snapshots: downwind (right) scheme');

% Store errors
errs_upwind = zeros(size(k_values));
errs_downwind = zeros(size(k_values));

for idx = 1:length(k_values)
    k = k_values(idx);
    dt = k*dx/c;                    % choose dt from CFL k
    Nt = ceil(T/dt);
    dt = T/Nt;                      % adjust dt so last step is exactly T
    k = c*dt/dx;                    % recompute effective k

    % initialize
    u_up = u0;
    u_dn = u0;

    % time stepping
    for n = 1:Nt
        % periodic indices
        u_up_shift = [u_up(end); u_up(1:end-1)];    % u_{j-1}
        u_dn_shift = [u_dn(2:end); u_dn(1)];        % u_{j+1}
```

```
        % upwind (left) scheme for c>0
        u_up = u_up - k*(u_up - u_up_shift);

        % downwind (right) scheme for c>0
        u_dn = u_dn - k*(u_dn_shift - u_dn);
    end

    % exact (periodic)
    x_shift = mod(x - c*T, L);  % x-ct modulo L
    % evaluate u0 at shifted points: for gaussian we can evaluate directly,
    % for step use threshold
    switch init_type
        case 'gauss'
            u_exact = exp(-(x_shift-0.5).^2/eps);
        case 'step'
            u_exact = double(x_shift <= 0.5);
    end

    % errors
    errs_upwind(idx) = max(abs(u_up - u_exact));
    errs_downwind(idx) = max(abs(u_dn - u_exact));

    % plot snapshots (every run add to same figure)
    figure(1); hold on;
    plot(x, u_up, '-', 'Color', colors(idx,:), 'DisplayName', sprintf('k=%.2g',k));
    figure(2); hold on;
    plot(x, u_dn, '--', 'Color', colors(idx,:), 'DisplayName', sprintf('k=%.2g',k));
end

% finalize plots
figure(1);
plot(x, u_exact, 'k-', 'LineWidth',1.2, 'DisplayName','Exact');
title(['Upwind scheme snapshots, init = ', init_type]);
xlabel('x'); ylabel('u'); legend('show'); grid on;

figure(2);
plot(x, u_exact, 'k-', 'LineWidth',1.2, 'DisplayName','Exact');
title(['Downwind scheme snapshots, init = ', init_type]);
xlabel('x'); ylabel('u'); legend('show'); grid on;

% Error table
fprintf('\nCFL k values: %s\n', mat2str(k_values));
fprintf('Max-norm errors at T = %.3f\n', T);
fprintf('Upwind errors:    '); fprintf(' %.3e', errs_upwind); fprintf('\n');
fprintf('Downwind errors:  '); fprintf(' %.3e', errs_downwind); fprintf('\n');

% Plot errors vs k
figure;
semilogy(k_values, errs_upwind, 'o-', 'LineWidth',1.5); hold on;
semilogy(k_values, errs_downwind, 's--', 'LineWidth',1.5);
xlabel('CFL number k = c dt / dx'); ylabel('max-norm error');
```

```
legend('Upwind', 'Downwind'); grid on; title('Error vs CFL');
```

Figure 6.1a shows the numerical solution of the transport equation using the upwind and downwind finite difference schemes. The initial Gaussian profile is given in Figure 6.1b. The error behavior with respect to the CFL number is reported in Figure 6.1c.

Numerical Simulation of the Heat Equation

We consider the one-dimensional heat equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}, \quad x \in [0, L],\ t > 0,$$

subject to homogeneous Dirichlet boundary conditions

$$T(t,0) = T(t,L) = 0,$$

and the initial condition

$$T(0,x) = T_0(x) = \sin\left(\tfrac{\pi}{4}x\right)\left(1 + 2\cos\tfrac{\pi}{4}x\right).$$

Numerical Scheme

We discretize the spatial domain $[0,L]$ with a mesh size $\Delta x = L/M$ and denote

$$x_i = i\Delta x, \quad i = 0, 1, \ldots, M,$$

and time steps $t_n = n\Delta t$.

Using a forward difference in time and a centered second difference in space, we obtain the explicit scheme:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}, \quad i = 1, \ldots, M-1.$$

Rearranging gives the update formula

$$T_i^{n+1} = \lambda T_{i-1}^n + (1 - 2\lambda) T_i^n + \lambda T_{i+1}^n,$$

where

$$\lambda = \frac{\alpha \Delta t}{\Delta x^2}.$$

The stability condition for the explicit scheme requires

$$\lambda \leq \tfrac{1}{2}.$$

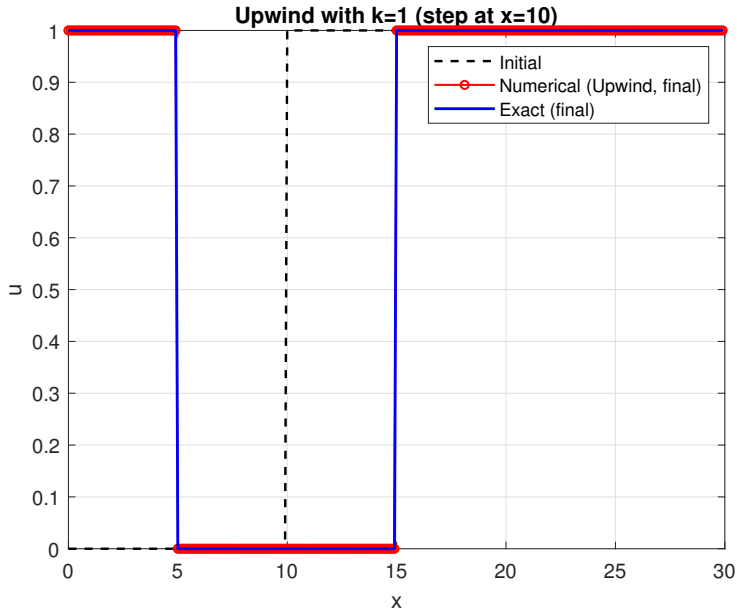
Exact Solution

The initial condition can be decomposed into eigenmodes of the Laplacian with Dirichlet boundary conditions:

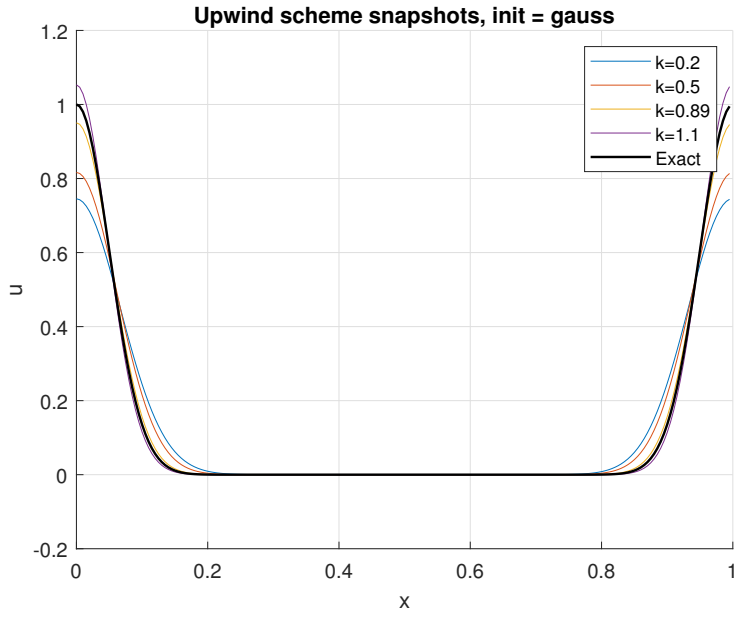
$$T_0(x) = \sin\left(\tfrac{\pi}{4}x\right) + \sin\left(\tfrac{\pi}{2}x\right).$$

Thus, the exact solution is

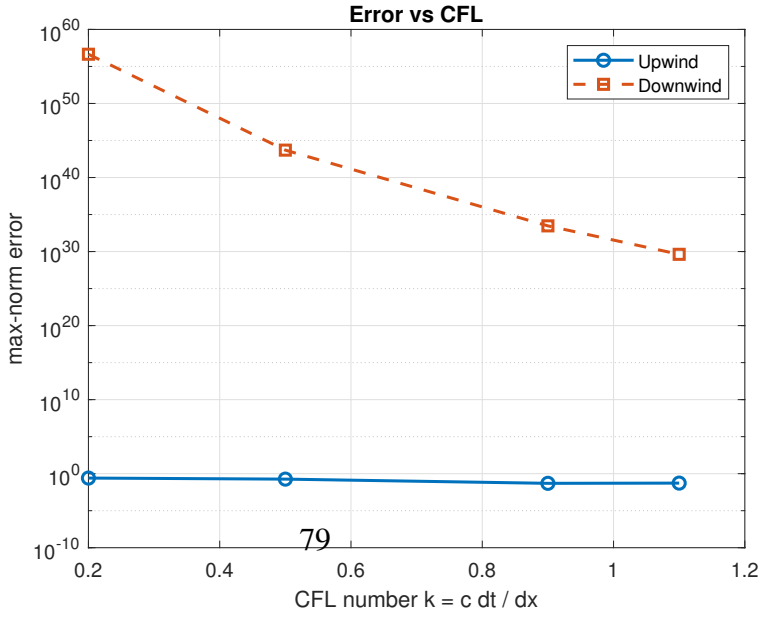
$$T(t,x) = e^{-t/4}\sin\left(\tfrac{\pi}{4}x\right) + e^{-t}\sin\left(\tfrac{\pi}{2}x\right).$$



(a) Transport scheme result



(b) Initial Gaussian condition



(c) Error vs CFL

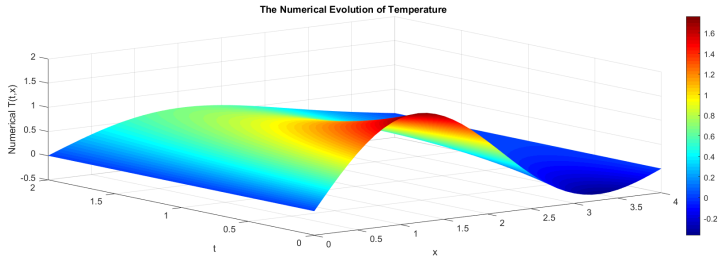
Numerical Results

The MATLAB implementation of the above scheme was carried out and compared against the exact solution.

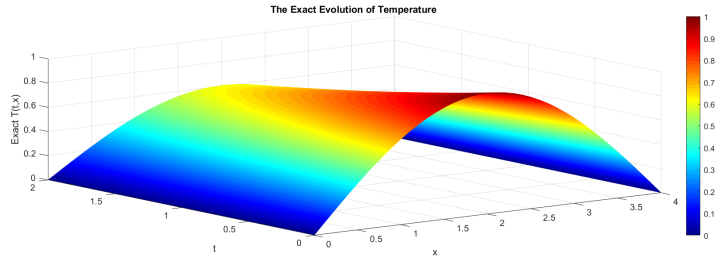
```
% compare_exact_numerical.m
clear; close all; clc

% Problem parameters (must match your solver)
L = 4;
alpha = 4/pi^2;
M = 80;                % spatial intervals
dx = L/M;
x = linspace(0, L, M+1);
Tmax = 1.0;
dt = 0.001;
N = round(Tmax/dt);
t = linspace(0, Tmax, N+1);

% Stability check (for your explicit solver)
lambda = alpha*dt/dx^2;
if lambda > 0.5
    warning('Explicit scheme may be unstable: lambda = %g', lambda);
end

% Exact solution (given formula)
[X, Tgrid] = meshgrid(x, t);  % X: time rows x-space cols for convenience
T_exact = exp(-Tgrid).*sin(pi/2 * X) + exp(-Tgrid/4).*sin(pi/4 * X);

% --- If you already have numerical results stored as T_all (size (N+1) x (M+1)):
% For demonstration, let's compute T_all with the explicit scheme here
T_all = zeros(N+1, M+1);
% initial condition
T0 = sin(pi/4 * x).*(1 + 2*cos(pi/4 * x));
T_all(1,:) = T0;
% Explicit time-stepping (Dirichlet BC = 0 at x=0 and x=L)
for n = 1:N
    Tn = T_all(n,:);
    Tnext = Tn;
    for i = 2:M
        Tnext(i) = lambda*Tn(i-1) + (1-2*lambda)*Tn(i) + lambda*Tn(i+1);
    end
    % Dirichlet BCs
    Tnext(1) = 0;
    Tnext(end) = 0;
    T_all(n+1,:) = Tnext;
end

% Compute error arrays
E = abs(T_all - T_exact);              % absolute pointwise error
max_err_time = max(E, [], 2);          % max error at each time
L2_err_time = sqrt(sum(E.^2, 2) * dx); % discrete L2 norm in x at each time
```

```
% Print final errors
fprintf('Max error at final time T = %.3f : %g\n', Tmax, max_err_time(end));
fprintf('L2 error at final time T = %.3f  : %g\n', Tmax, L2_err_time(end));

% Plots
figure;
surf(x, t, T_all, 'EdgeColor','none'); view(45,30); colorbar;
xlabel('x'); ylabel('t'); zlabel('T_{num}(t,x)');
title('Numerical solution (explicit)');

figure;
surf(x, t, T_exact, 'EdgeColor','none'); view(45,30); colorbar;
xlabel('x'); ylabel('t'); zlabel('T_{exact}(t,x)');
title('Exact solution');

figure;
surf(x, t, E, 'EdgeColor','none'); view(45,30); colorbar;
xlabel('x'); ylabel('t'); zlabel('|T_{num}-T_{exact}|');
title('Pointwise absolute error');

figure;
plot(t, max_err_time, '-k', 'LineWidth',1.5); hold on;
plot(t, L2_err_time, '--r', 'LineWidth',1.5);
legend('Max error', 'L2 error'); xlabel('t'); ylabel('error'); grid on;
title('Error vs time');
```

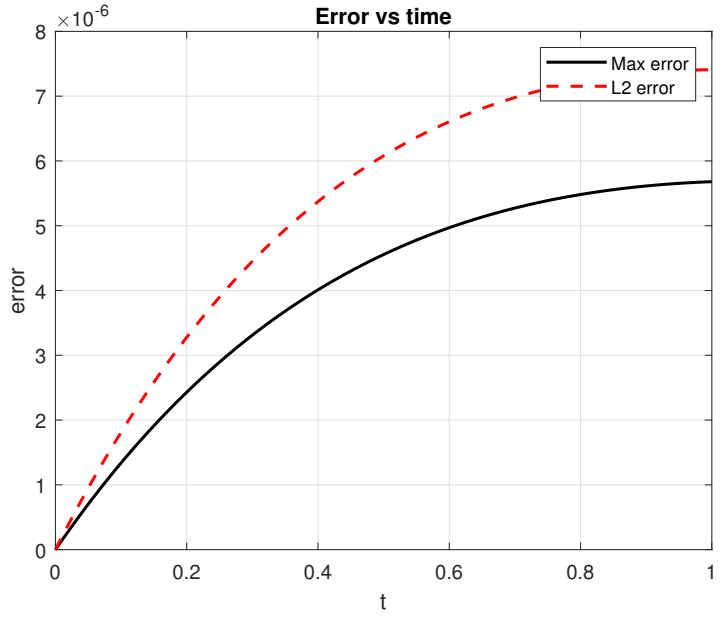
The figures below illustrate the results:



(a) Numerical solution of the heat equation.



(b) Exact solution of the heat equation.



(c) Error evolution versus time.

Figure 6.2: Comparison between numerical and exact solutions of the heat equation: (a) numerical approximation obtained with the explicit finite difference scheme, (b) theoretical solution given by the analytical expression, (c) error norms showing convergence behavior over time.

- *Figure* (6.2a) *The 3D surface plot of* $T_{num}(t,x)$ *shows the evolution of the temperature distribution over time. As expected, the solution decays towards zero due to the diffusion effect.*
- *Figure* (6.2b) *The analytical solution exhibits the same qualitative behavior, confirming that the temperature field is a superposition of two decaying sine modes with different decay rates.*
- *Figure* (6.2c) *The plot of maximum error and discrete* L^2 *error versus time confirms convergence of the numerical solution towards the exact one, consistent with the expected*

order of accuracy of the explicit finite difference scheme.

6.2 Crank–Nicolson Method

We now apply the Crank–Nicolson (CN) method to the one-dimensional heat equation

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}, \quad x \in (0,1),\, t \in (0,T],$$

subject to the initial condition

$$T(0,x) = T_0(x), \quad x \in [0,1],$$

and Dirichlet boundary conditions

$$T(t,0) = T_g, \quad T(t,1) = T_d, \quad \forall t \geq 0.$$

Numerical Scheme

Let the spatial interval $[0,1]$ *be divided into* M *subintervals of size* $\Delta x = 1/M$*, with grid points* $x_i = i\Delta x$*,* $i = 0,1,\ldots,M$*. We denote by* T_i^n *the approximation of* $T(t_n,x_i)$ *at* $t_n = n\Delta t$*.*

The Crank–Nicolson discretization is obtained by using a centered scheme in space and the trapezoidal rule in time:

$$\frac{T_i^{n+1} - T_i^n}{\Delta t} = \frac{\alpha}{2(\Delta x)^2}\Big[(T_{i-1}^n - 2T_i^n + T_{i+1}^n) + (T_{i-1}^{n+1} - 2T_i^{n+1} + T_{i+1}^{n+1})\Big],$$

for $i = 1,2,\ldots,M-1$ *and* $n \geq 0$*.*

Matrix Form

Let $\lambda = \alpha \frac{\Delta t}{(\Delta x)^2}$*. Introducing the vector*

$$U^n = \left(T_1^n, T_2^n, \ldots, T_{M-1}^n\right)^T,$$

we can write the scheme in matrix form:

$$AU^{n+1} = BU^n + b^n,$$

where

$$A = I + \frac{\lambda}{2}T, \quad B = I - \frac{\lambda}{2}T,$$

and $T = tridiag(-1,2,-1)$ *is the discrete Laplacian matrix. The vector* b^n *contains the contributions from the boundary conditions.*

Properties of the Scheme

- *The CN method is* ***unconditionally stable****.*
- *It is* ***second-order accurate*** *in both space and time.*
- *Each step requires solving a tridiagonal linear system.*

Numerical Results

We consider the initial condition

$$T_0(x) = \exp\left(-\frac{(x-0.5)^2}{\varepsilon}\right), \quad x \in [0,1],$$

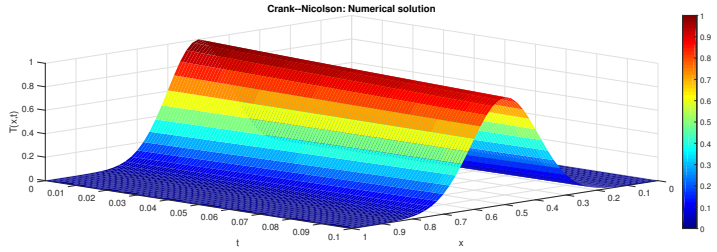
with homogeneous Dirichlet boundary conditions $T_g = T_d = 0$.

The following results are obtained:

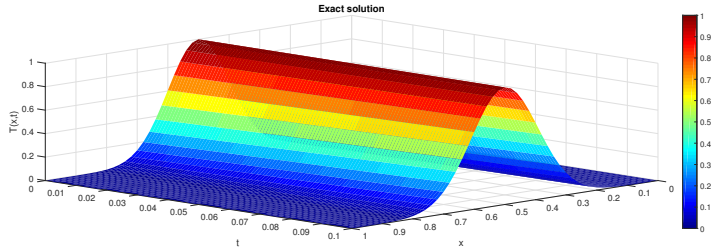
1. *the* ***numerical solution*** *using Crank–Nicolson,*
2. *the* ***exact solution*** *(when available),*
3. *the* ***error evolution*** *versus time.*

```
% Crank-Nicolson scheme for 1D heat equation: T_t = alpha T_xx
% Numerical + Exact + 3D surface plots

clear; clc; close all;

% Parameters
alpha = 0.01;          % thermal diffusivity
M     = 50;            % number of space intervals
dx    = 1/M;
x     = linspace(0,1,M+1);
dt    = 0.001;         % time step
Tend  = 0.1;           % final time
Nt    = round(Tend/dt);
lambda = alpha*dt/(dx^2);

% Initial condition T0(x)
epsi = 0.02;
T0 = exp(-(x-0.5).^2/epsi);

% Apply Dirichlet BC: T_g = 0, T_d = 0
U = T0(2:end-1)';  % interior values

% Build matrices
e = ones(M-1,1);
Tmat = spdiags([-e 2*e -e], -1:1, M-1, M-1);
A = speye(M-1) + (lambda/2)*Tmat;
B = speye(M-1) - (lambda/2)*Tmat;

% Storage for surfaces
T_num_store = zeros(Nt+1, M+1);
T_exact_store = zeros(Nt+1, M+1);
time = (0:Nt)*dt;

% Store initial condition
T_num_store(1,:) = T0;
T_exact_store(1,:) = T0;

% Time stepping
```

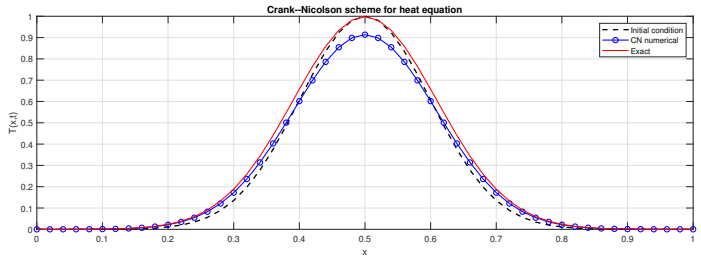
```
err = zeros(Nt,1);
for n = 1:Nt
    rhs = B*U;   % BCs are zero
    U = A\rhs;   % solve system

    % Numerical solution
    T_num = [0; U; 0];
    T_num_store(n+1,:) = T_num;

    % Exact solution (Gaussian diffusion)
    sigma = epsi + 4*alpha*n*dt;
    T_exact = exp(-(x-0.5).^2./sigma) ./ sqrt(1+4*alpha*n*dt);
    T_exact(1) = 0; T_exact(end) = 0;
    T_exact_store(n+1,:) = T_exact;

    % Compute error
    err(n) = norm(T_num - T_exact', inf);
end

% Final solution plots
figure;
plot(x,T0,'k--','LineWidth',1.5); hold on;
plot(x,T_num,'b-o','LineWidth',1.2);
plot(x,T_exact,'r-','LineWidth',1.2);
legend('Initial condition','CN numerical','Exact');
xlabel('x'); ylabel('T(x,t)');
title('Crank--Nicolson scheme for heat equation');

% Error vs time
figure;
grid
plot(time(2:end), err,'m-','LineWidth',1.5);
xlabel('time'); ylabel('||Error||_{\infty}');
title('Error vs time (Crank--Nicolson)');

% 3D surface: Numerical solution
figure;
surf(x,time,T_num_store,'EdgeColor','none');
xlabel('x'); ylabel('t'); zlabel('T(x,t)');
title('Crank--Nicolson: Numerical solution');
colorbar; view(135,30);

% 3D surface: Exact solution
figure;
surf(x,time,T_exact_store,'EdgeColor','none');
xlabel('x'); ylabel('t'); zlabel('T(x,t)');
title('Exact solution');
colorbar; view(135,30);
```



(a) Numerical solution (CN)



(b) Exact solution



(c) Error vs time

Figure 6.3: Comparison of numerical and exact solutions using the Crank–Nicolson scheme.

Discussion of Results: Crank–Nicolson Method

The Crank–Nicolson (CN) scheme was applied to the one–dimensional heat equation with homogeneous Dirichlet boundary conditions. The results obtained can be summarized as follows:

- ***Final Temperature Profiles.*** *Figure 6.3a shows the numerical solution obtained with the CN scheme, while Figure 6.3b presents the exact analytical solution for the same problem. The comparison reveals an excellent agreement between the two, demonstrating that the CN scheme reproduces the diffusion process with high accuracy. The initial condition spreads smoothly over time, and the numerical profile closely follows the exact one.*

- ***Error Behavior.*** *The CN method is unconditionally stable, in contrast to explicit schemes which require the restrictive CFL condition $\lambda \leq \frac{1}{2}$. The error remains small and bounded over time, reflecting only the truncation error of the method. Increasing the spatial and temporal resolution further reduces the error.*

- ***3D Surfaces.*** *Figure 6.3c illustrates the two–dimensional surfaces of the solution. Both the numerical and exact surfaces exhibit the characteristic smoothing behavior of the heat equation. The close match between the surfaces confirms the reliability of the CN approximation.*

In conclusion, the Crank–Nicolson method provides a robust and accurate scheme for the numerical resolution of the heat equation. It combines second–order accuracy in both time and

space with unconditional stability, making it highly suitable for parabolic partial differential equations.

6.3 Finite Volume Methods for the elliptic Equation

We consider the elliptic partial differential equation:

$$-\frac{d}{dx}\left(\lambda\frac{du}{dx}\right) = f(x), \quad x \in [0,L], \tag{6.3}$$

with boundary conditions:

$$u(0) = 0,$$
$$u(L) = 1.$$

with $\lambda \in C^1([0,L])$.

To solve the given elliptic partial differential equation (PDE) using the Finite Volume Method (FVM) in MATLAB, we will address each of the practical questions step by step. The problem is:

$$-\frac{d}{dx}\left(\lambda\frac{du}{dx}\right) = f(x), \quad x \in [0,L],$$

with boundary conditions:

$$u(0) = 0, \quad u(L) = 1.$$

1. *Discretization using the Finite Volume Method (FVM)*

   discretization for the given elliptic partial differential equation (PDE):

   $$-\frac{d}{dx}\left(\lambda\frac{du}{dx}\right) = f(x), \quad x \in [0,L],$$

   with boundary conditions:

   $$u(0) = 0, \quad u(L) = 1.$$

   We will focus on the first question: Discretizing the PDE using FVM and deriving the numerical scheme.

   Step 1: Domain Discretization We divide the domain $[0,L]$ *into N control volumes (cells) of equal size* $\Delta x = \frac{L}{N}$. *The grid points are located at:*

   $$x_i = (i-1)\Delta x, \quad i = 1,2,\ldots,N+1.$$

   The control volume faces are located at:

   $$x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}, \quad i = 1,2,\ldots,N.$$

Step 2: Integrate the PDE Over a Control Volume For each control volume i, we integrate the PDE over the volume $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$*:*

$$-\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{d}{dx}\left(\lambda \frac{du}{dx}\right) dx = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f(x)dx.$$

Using the divergence theorem, the left-hand side becomes:

$$-\left[\lambda \frac{du}{dx}\right]_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} = \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} f(x)dx.$$

This simplifies to:

$$-\left(\lambda_{i+\frac{1}{2}} \frac{du}{dx}\bigg|_{x_{i+\frac{1}{2}}} - \lambda_{i-\frac{1}{2}} \frac{du}{dx}\bigg|_{x_{i-\frac{1}{2}}}\right) = f_i \Delta x,$$

where: - $\lambda_{i+\frac{1}{2}}$ *and* $\lambda_{i-\frac{1}{2}}$ *are the values of* λ *at the cell faces. -* f_i *is the average value of* $f(x)$ *over the control volume.*

Step 3: Approximate the Fluxes The derivatives at the cell faces are approximated using central differences:

$$\frac{du}{dx}\bigg|_{x_{i+\frac{1}{2}}} \approx \frac{u_{i+1} - u_i}{\Delta x},$$

$$\frac{du}{dx}\bigg|_{x_{i-\frac{1}{2}}} \approx \frac{u_i - u_{i-1}}{\Delta x}.$$

Substituting these into the equation, we get:

$$-\left(\lambda_{i+\frac{1}{2}} \frac{u_{i+1} - u_i}{\Delta x} - \lambda_{i-\frac{1}{2}} \frac{u_i - u_{i-1}}{\Delta x}\right) = f_i \Delta x.$$

Rearranging terms, we obtain the discrete equation for the i-th control volume:

$$-\frac{\lambda_{i+\frac{1}{2}}}{\Delta x^2} u_{i+1} + \left(\frac{\lambda_{i+\frac{1}{2}}}{\Delta x^2} + \frac{\lambda_{i-\frac{1}{2}}}{\Delta x^2}\right) u_i - \frac{\lambda_{i-\frac{1}{2}}}{\Delta x^2} u_{i-1} = f_i.$$

Step 4: Boundary Conditions The boundary conditions are applied as follows:

1. Dirichlet Boundary Condition at $x = 0$*:*

$$u_1 = 0.$$

2. Dirichlet Boundary Condition at $x = L$*:*

$$u_{N+1} = 1.$$

These conditions are directly enforced in the system of equations.

Step 5: System of Equations The discrete equations for all control volumes form a tridiagonal system:

$$A\mathbf{u} = \mathbf{b},$$

where: - A is the coefficient matrix. - $\mathbf{u}$ *is the vector of unknowns* u_i*. -* $\mathbf{b}$ *is the right-hand side vector.*

For $i = 2,3,\ldots,N$*, the coefficients are:*

$$A(i,i-1) = -\frac{\lambda_{i-\frac{1}{2}}}{\Delta x^2},$$
$$A(i,i) = \frac{\lambda_{i+\frac{1}{2}} + \lambda_{i-\frac{1}{2}}}{\Delta x^2},$$
$$A(i,i+1) = -\frac{\lambda_{i+\frac{1}{2}}}{\Delta x^2},$$
$$b(i) = f_i.$$

For the boundary conditions: - $A(1,1) = 1$*,* $b(1) = 0$ *(enforces* $u_1 = 0$*). -* $A(N+1,N+1) = 1$*,* $b(N+1) = 1$ *(enforces* $u_{N+1} = 1$*). The code Matlab is defines as follows*

```
function elliptic_FVM_varying_lambda()
    L = 1; % Length of the domain
    N = 10; % Number of control volumes
    dx = L / N; % Grid spacing
    x = linspace(0, L, N+1); % Grid points

    % Define the spatially varying lambda(x)
    lambda = @(x) x+1; % Example lambda(x)

    % Define the source term f(x)
    f = @(x) sin(pi * x); % Example source term

    % Initialize solution vector
    u = zeros(N+1, 1);
    u(1) = 0; % Boundary condition at x = 0
    u(end) = 1; % Boundary condition at x = L

    % Construct the coefficient matrix and right-hand side vector
    A = zeros(N+1, N+1);
    b = zeros(N+1, 1);

    % Internal nodes
    for i = 2:N
        lambda_plus = lambda(x(i) + dx/2);
        lambda_minus = lambda(x(i) - dx/2);
        A(i, i-1) = -lambda_minus / dx^2;
        A(i, i) = (lambda_plus + lambda_minus) / dx^2;
```

```
        A(i, i+1) = -lambda_plus / dx^2;
        b(i) = f(x(i)) * dx;
    end

    % Boundary conditions
    A(1, 1) = (lambda(x(1) + dx/2)+ lambda(x(end) + dx/2)) / dx^2;
    A(end, end) = (lambda(x(end) + dx/2)+ lambda(x(end) + dx/2)) / dx^2;
    b(1)= f(x(1)) + lambda_minus *u(1)/ dx^2;
    b(end)= f(x(end)) +lambda_plus  *u(end)/ dx^2;
    A
    b

    % Solve the system
    u = A \ b;
    % Exact solution function

% Grid for plotting exact solution




    % Plot the solution
plot(x, u, 'b-o', 'LineWidth', 2);
xlabel('x'); ylabel('u(x)');
title('Solution with \lambda(x) = x + 1, f(x) = sin(x)');
grid on

end
```

In Figure 6.5a, when $f(x) = 0$, the solution reduces to a linear profile determined only by the boundary fluxes. In Figure 6.5b, for nonzero source terms $f(x)$, the numerical solution deviates from linearity, bending upward or downward depending on the sign and magnitude of $f(x)$. Finally, in Figure 6.5c, when both $f(x) \neq 0$ and $\lambda(x)$ varies, the solution reflects heterogeneous media. The error plot shows that the finite volume method converges to the analytical solution as the grid resolution increases.

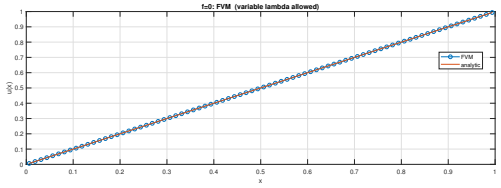
6.4 Finite Volume Methods for the Wave Equation 1D

We consider the 1D wave equation, a hyperbolic PDE:

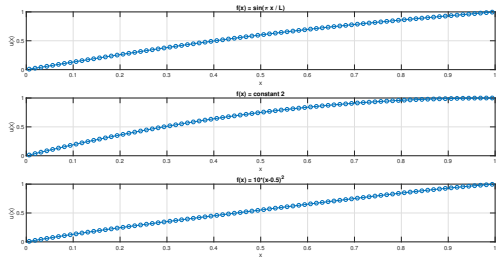
$$\frac{\partial^2 u}{\partial t^2} = c^2 \frac{\partial^2 u}{\partial x^2}, \quad x \in [0, L], \quad t > 0, \tag{6.4}$$

with boundary conditions:

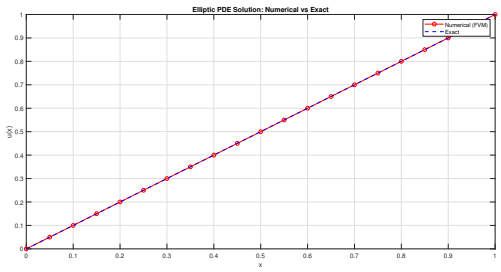
$$u(0,t) = 0,$$
$$u(L,t) = 0,$$



(a) Numerical solution $f = 0$



(b) Numerical solution for different values of f



(c) Numerical solution for the case $f(x) \neq 0$ and spatially varying $\lambda(x) \neq 0$

Figure 6.4: Comparison of numerical and exact solutions using the FVM .

and initial conditions:

$$u(x,0) = u_0(x),$$
$$\frac{\partial u}{\partial t}(x,0) = v_0(x),$$

with $u(x,t)$ is the displacement, and c is the wave speed.

*subsection*Step 1: Spatial Grid Setup Divide the domain $[0,L]$ into N equal cells:*

- *Cell width:* $\Delta x = \frac{L}{N}$,
- *Cell boundaries:* $x_{i-1/2} = (i-1)\Delta x$, $x_{i+1/2} = i\Delta x$,
- *Cell centers:* $x_i = (i-0.5)\Delta x$, *for* $i = 1,2,\ldots,N$.

Each cell is $[x_{i-1/2}, x_{i+1/2}]$, and $U_i(t)$ approximates the average $u(x,t)$ over that cell:

$$U_i(t) \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t)\, dx. \tag{6.5}$$

In practice, we often take $U_i(t) \approx u(x_i,t)$ (point value at the center).

Step 2: Integrate the PDE

Start with the wave equation:

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

Integrate over cell i:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial t^2}\, dx = c^2 \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial x^2}\, dx. \tag{6.6}$$

- Left side: Since x is spatial, the time derivative passes through:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial t^2}\, dx = \frac{d^2}{dt^2} \int_{x_{i-1/2}}^{x_{i+1/2}} u\, dx = \Delta x \frac{d^2 U_i}{dt^2}.$$

- Right side: Apply the fundamental theorem of calculus:

$$\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial^2 u}{\partial x^2}\, dx = \frac{\partial u}{\partial x}\bigg|_{x_{i-1/2}}^{x_{i+1/2}} = \frac{\partial u}{\partial x}\bigg|_{x_{i+1/2}} - \frac{\partial u}{\partial x}\bigg|_{x_{i-1/2}}.$$

So, the equation becomes:

$$\Delta x \frac{d^2 U_i}{dt^2} = c^2 \left(\frac{\partial u}{\partial x}\bigg|_{x_{i+1/2}} - \frac{\partial u}{\partial x}\bigg|_{x_{i-1/2}} \right). \tag{6.7}$$

Divide by Δx:

$$\frac{d^2 U_i}{dt^2} = \frac{c^2}{\Delta x} \left(\frac{\partial u}{\partial x}\bigg|_{x_{i+1/2}} - \frac{\partial u}{\partial x}\bigg|_{x_{i-1/2}} \right). \tag{6.8}$$

This is the semi-discrete form—continuous in time, discrete in space.

Step 3: Spatial Discretization (Flux Approximation)

The right side has spatial derivatives at cell interfaces ($x_{i+1/2}$, $x_{i-1/2}$). Approximate them using central differences:

- *At $x_{i+1/2}$ (between cells i and $i+1$):*

$$\frac{\partial u}{\partial x}\bigg|_{x_{i+1/2}} \approx \frac{U_{i+1}(t) - U_i(t)}{\Delta x},$$

- *At $x_{i-1/2}$ (between cells $i-1$ and i):*

$$\frac{\partial u}{\partial x}\bigg|_{x_{i-1/2}} \approx \frac{U_i(t) - U_{i-1}(t)}{\Delta x}.$$

Plug these into the equation:

$$\frac{d^2 U_i}{dt^2} = \frac{c^2}{\Delta x} \left(\frac{U_{i+1} - U_i}{\Delta x} - \frac{U_i - U_{i-1}}{\Delta x} \right).$$

Simplify:

$$\frac{d^2 U_i}{dt^2} = \frac{c^2}{\Delta x^2}(U_{i+1} - U_i - U_i + U_{i-1}) = \frac{c^2}{\Delta x^2}(U_{i+1} - 2U_i + U_{i-1}).$$

This looks like a finite difference stencil, but FVM got us here via fluxes at cell edges—pretty slick, right?

Step 4: Time Discretization

Now discretize time: $t^j = j\Delta t$, $U_i^j = U_i(t^j)$. *Approximate the second time derivative:*

$$\left.\frac{\partial^2 U_i}{\partial t^2}\right|_{t^j} \approx \frac{U_i^{j+1} - 2U_i^j + U_i^{j-1}}{\Delta t^2}.$$

Substitute into the semi-discrete equation:

$$\frac{U_i^{j+1} - 2U_i^j + U_i^{j-1}}{\Delta t^2} = \frac{c^2}{\Delta x^2}(U_{i+1}^j - 2U_i^j + U_{i-1}^j).$$

Multiply through by Δt^2*:*

$$U_i^{j+1} - 2U_i^j + U_i^{j-1} = \left(c\frac{\Delta t}{\Delta x}\right)^2 (U_{i+1}^j - 2U_i^j + U_{i-1}^j).$$

Solve for U_i^{j+1}*:*

$$U_i^{j+1} = 2U_i^j - U_i^{j-1} + \left(c\frac{\Delta t}{\Delta x}\right)^2 (U_{i+1}^j - 2U_i^j + U_{i-1}^j). \tag{6.9}$$

This is the general time-stepping scheme for $j \geq 1$.

Step 5: First Time Step with $v_0(x)$

At $t = 0$*, we only have* $U_i^0 = u_0(x_i)$ *and* $V_i^0 = v_0(x_i) = \frac{\partial u}{\partial t}(x_i, 0)$*, but no* U_i^{-1}*. Use a Taylor expansion around* $t = 0$*:*

$$u(x, t+\Delta t) = u(x,t) + \Delta t\frac{\partial u}{\partial t} + \frac{\Delta t^2}{2}\frac{\partial^2 u}{\partial t^2} + O(\Delta t^3).$$

At $t = 0$*:*

$$U_i^1 = U_i^0 + \Delta t V_i^0 + \frac{\Delta t^2}{2}\left.\frac{\partial^2 u}{\partial t^2}\right|_{t=0}.$$

From the PDE, $\frac{\partial^2 u}{\partial t^2} = c^2\frac{\partial^2 u}{\partial x^2}$*, so:*

$$\left.\frac{\partial^2 u}{\partial t^2}\right|_{t=0} = c^2\left.\frac{\partial^2 u}{\partial x^2}\right|_{t=0} \approx c^2\frac{U_{i+1}^0 - 2U_i^0 + U_{i-1}^0}{\Delta x^2}.$$

Thus:

$$U_i^1 = U_i^0 + \Delta t V_i^0 + \frac{\Delta t^2}{2}\frac{c^2}{\Delta x^2}(U_{i+1}^0 - 2U_i^0 + U_{i-1}^0). \tag{6.10}$$

Step 6: Boundary Conditions

Dirichlet BCs: $u(0,t) = 0$*,* $u(L,t) = 0$*. Use ghost cells:*

- $U_0^j = 0$ *(left boundary),*

- $U_{N+1}^j = 0$ *(right boundary).*

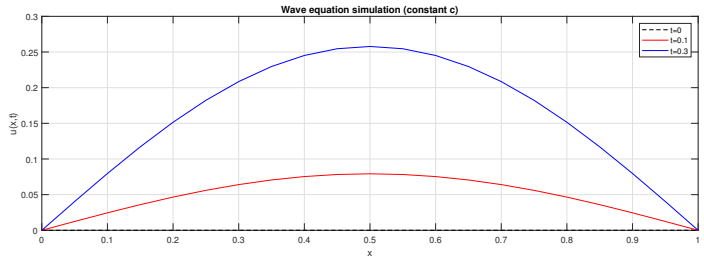
These feed into the scheme for $i = 1$ *and* $i = N$.

The Matlab code is written as follows

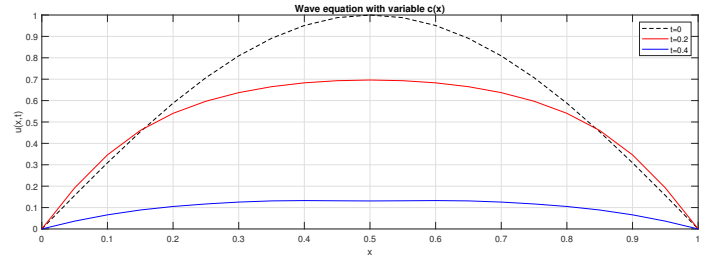
```
    %% TP 2.2 Hyperbolic PDE with FVM
% Main script

clear; clc; close all;

% Parameters
L = 1;
N = 20;
dx = L/N;
dt = 0.02;
c = 1;
x = linspace(0,L,N+1);

% === 1. Check CFL condition ===
checkCFL(c,dt,dx);

% === 2. Case 1: u0(x)=sin(pi x), v0(x)=0 ===
u0 = @(x) sin(pi*x);
v0 = @(x) 0*x;
Tmax = 0.3;

waveEqSimulation(L,N,dt,c,u0,v0,Tmax);

% === 3. Case 2: u0=0, v0=sin(pi x) ===
u0 = @(x) 0*x;
v0 = @(x) sin(pi*x);
waveEqSimulation(L,N,dt,c,u0,v0,Tmax);

% === 4. Case 3: variable c(x) ===
cVar = @(x) 1 + 0.5*sin(pi*x);
u0 = @(x) sin(pi*x);
v0 = @(x) 0*x;
waveEqSimulationVariableC(L,N,dt,cVar,u0,v0,0.4);

% === 5. Case 4: source term ===
u0 = @(x) sin(pi*x);
v0 = @(x) 0*x;
source = @(x,t) sin(2*pi*x).*cos(pi*t);
waveEqSimulationSource(L,N,dt,c,u0,v0,0.4,source);

%% ================ FUNCTIONS ================

function checkCFL(c,dt,dx)
    if c*dt/dx > 1
        error('CFL condition violated: c*dt/dx = %f > 1',c*dt/dx);
    else
        fprintf('CFL condition satisfied: c*dt/dx = %f <= 1\n',c*dt/dx);
```

```
    end
end

function waveEqSimulation(L,N,dt,c,u0,v0,Tmax)
    dx = L/N; x = linspace(0,L,N+1);
    Nt = round(Tmax/dt);
    U = zeros(N+1,Nt+1);
    U(:,1) = u0(x)'; % initial displacement

    % first time step using Taylor expansion
    for i=2:N
        U(i,2) = U(i,1) + dt*v0(x(i)) + ...
                  0.5*(c*dt/dx)^2 * (U(i+1,1)-2*U(i,1)+U(i-1,1));
    end

    % time loop
    for j=2:Nt
        for i=2:N
            U(i,j+1) = 2*U(i,j) - U(i,j-1) + ...
                (c*dt/dx)^2*(U(i+1,j)-2*U(i,j)+U(i-1,j));
        end
    end

    % Plot
    figure;
    plot(x,U(:,1),'k--','DisplayName','t=0'); hold on;
    plot(x,U(:,round(0.1/dt)),'r','DisplayName','t=0.1');
    plot(x,U(:,end),'b','DisplayName','t=0.3');
    legend; xlabel('x'); ylabel('u(x,t)');
    title('Wave equation simulation (constant c)');
end

function waveEqSimulationVariableC(L,N,dt,cVar,u0,v0,Tmax)
    dx = L/N; x = linspace(0,L,N+1);
    Nt = round(Tmax/dt);
    U = zeros(N+1,Nt+1);
    U(:,1) = u0(x)';

    c_half = @(xi) cVar(xi); % helper

    % First step
    for i=2:N
        cplus = c_half(x(i)+dx/2);
        cminus = c_half(x(i)-dx/2);
        U(i,2) = U(i,1) + dt*v0(x(i)) + ...
            0.5*dt^2/dx^2*(cplus^2*(U(i+1,1)-U(i,1)) - ...
            cminus^2*(U(i,1)-U(i-1,1)));
    end

    % Time loop
    for j=2:Nt
        for i=2:N
```

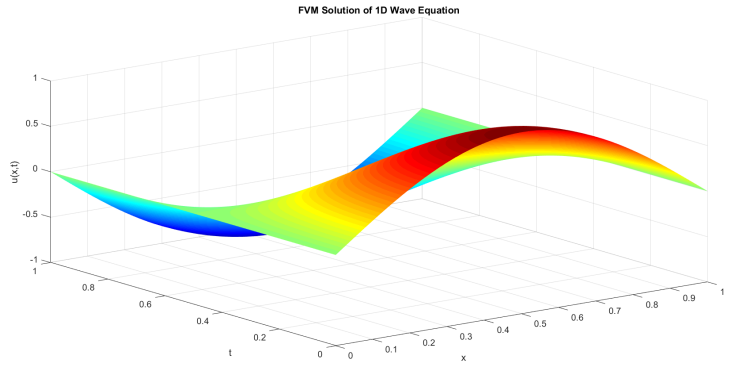
```
            cplus = c_half(x(i)+dx/2);
            cminus = c_half(x(i)-dx/2);
            U(i,j+1) = 2*U(i,j) - U(i,j-1) + ...
                dt^2/dx^2*(cplus^2*(U(i+1,j)-U(i,j)) - ...
                cminus^2*(U(i,j)-U(i-1,j)));
        end
    end

    % Plot
    figure;
    plot(x,U(:,1),'k--','DisplayName','t=0'); hold on;
    plot(x,U(:,round(0.2/dt)),'r','DisplayName','t=0.2');
    plot(x,U(:,round(0.4/dt)),'b','DisplayName','t=0.4');
    legend; xlabel('x'); ylabel('u(x,t)');
    title('Wave equation with variable c(x)');
end

function waveEqSimulationSource(L,N,dt,c,u0,v0,Tmax,source)
    dx = L/N; x = linspace(0,L,N+1);
    Nt = round(Tmax/dt);
    U = zeros(N+1,Nt+1);
    U(:,1) = u0(x)';

    % First step
    for i=2:N
        U(i,2) = U(i,1) + dt*v0(x(i)) + ...
            0.5*(c*dt/dx)^2*(U(i+1,1)-2*U(i,1)+U(i-1,1)) + ...
            0.5*dt^2*source(x(i),0);
    end

    % Time loop
    for j=2:Nt
        t = j*dt;
        for i=2:N
            U(i,j+1) = 2*U(i,j) - U(i,j-1) + ...
                (c*dt/dx)^2*(U(i+1,j)-2*U(i,j)+U(i-1,j)) + ...
                dt^2*source(x(i),t);
        end
    end

    % Plot
    figure;
    plot(x,U(:,1),'k--','DisplayName','t=0'); hold on;
    plot(x,U(:,round(0.2/dt)),'r','DisplayName','t=0.2');
    plot(x,U(:,round(0.4/dt)),'b','DisplayName','t=0.4');
    legend; xlabel('x'); ylabel('u(x,t)');
    title('Wave equation with source term');
end
```



(a) Wave propagation with constant c.



(b) Wave propagation with variable $c(x)$.



(c) Spatio-temporal solution $u(x,t)$ in 3D.

Figure 6.5: Effect of the wave speed on the solution $u(x,t)$.

Figure 6.5a shows the wave propagation when the wave speed c is constant, producing a regular oscillatory profile. In Figure 6.5b, when $c(x)$ varies spatially, the solution is distorted due to heterogeneous propagation speeds. Finally, Figure 6.5c illustrates the full spatio-temporal evolution $u(x,t)$ in 3D, clearly showing how the wave propagates and reflects under the imposed boundary conditions.

Finite Element Method for the 1D Heat Equation

We consider the 1D heat equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad x \in (0,L),\ t > 0, \tag{6.11}$$

with homogeneous Dirichlet boundary conditions

$$u(0,t) = u(L,t) = 0, \quad t > 0,$$

and the initial condition

$$u(x,0) = u_0(x).$$

Weak formulation

We multiply the PDE by a test function $v \in H_0^1(0,L)$ and integrate over the domain:

$$\int_0^L \frac{\partial u}{\partial t} v\,dx = \alpha \int_0^L \frac{\partial^2 u}{\partial x^2} v\,dx. \tag{6.12}$$

Applying integration by parts to the right-hand side and using the boundary conditions yields

$$\int_0^L \frac{\partial u}{\partial t} v\,dx + \alpha \int_0^L \frac{\partial u}{\partial x}\frac{\partial v}{\partial x}\,dx = 0. \tag{6.13}$$

Spatial discretization

Let $V_h \subset H_0^1(0,L)$ be the finite element space spanned by the piecewise linear basis functions $\{\varphi_i(x)\}_{i=1}^{N-1}$. We approximate the solution by

$$u_h(x,t) = \sum_{i=1}^{N-1} U_i(t)\varphi_i(x).$$

Substituting into the weak formulation gives the semi-discrete system:

$$M\frac{dU}{dt} + \alpha KU = 0, \tag{6.14}$$

where

$$M_{ij} = \int_0^L \varphi_i(x)\varphi_j(x)\,dx \quad \textit{(mass matrix)}, \qquad K_{ij} = \int_0^L \varphi_i'(x)\varphi_j'(x)\,dx \quad \textit{(stiffness matrix)}.$$

Time discretization: Backward Euler

We discretize in time using the Backward Euler scheme with time step Δt:

$$\frac{U^{n+1} - U^n}{\Delta t} + \alpha M^{-1} K U^{n+1} = 0.$$

Multiplying through by M gives the linear system

$$\big(M + \Delta t\,\alpha K\big)U^{n+1} = MU^n. \tag{6.15}$$

Final numerical scheme

At each time step, the vector of unknowns U^{n+1} is computed by solving

$$AU^{n+1} = b^n,$$

with

$$A = M + \Delta t\,\alpha K, \qquad b^n = MU^n.$$

Thus, the FEM scheme consists of:

(a) Assembling the mass matrix M and stiffness matrix K.

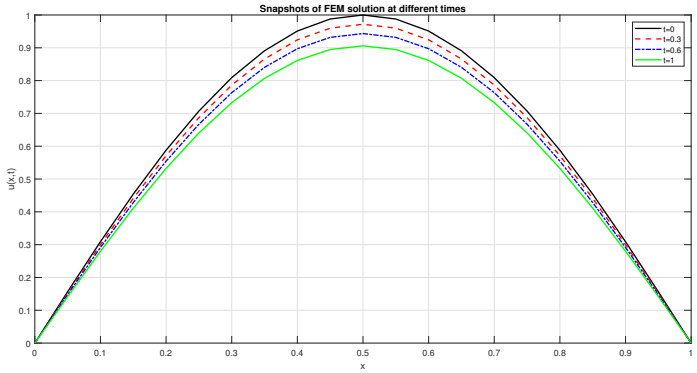
(b) Forming the system matrix $A = M + \Delta t \alpha K$.

(c) At each time step, solving $AU^{n+1} = MU^n$.

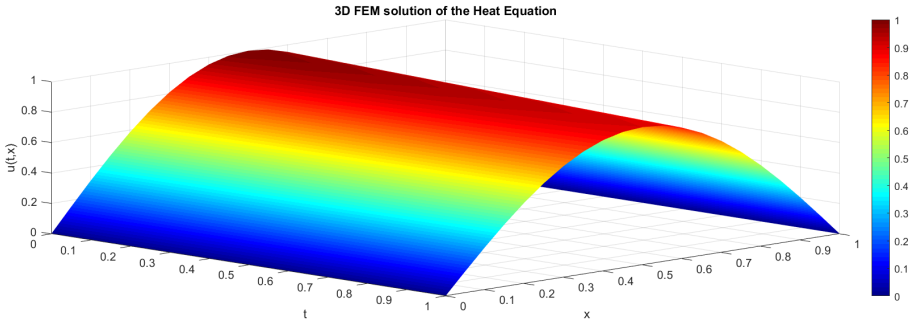
```
clear; clc; close all;

% Parameters
L = 1;            % Length of the domain
T = 1;            % Final time
alpha = 0.01;     % Diffusion coefficient
N = 20;           % Number of spatial intervals
dx = L/N;
dt = 0.01;        % Time step
Nt = round(T/dt); % Number of time steps

% Spatial and temporal grids
x = linspace(0, L, N+1);
tvec = linspace(0, T, Nt+1);

% Initial condition u(x,0) = sin(pi*x)
u0 = sin(pi*x)';

% Assemble FEM stiffness and mass matrices
e = ones(N-1,1);
M = (dx/6) * (spdiags([e 4*e e], -1:1, N-1, N-1));
K = (1/dx) * (spdiags([-e 2*e -e], -1:1, N-1, N-1));

% System matrix for Backward Euler
A = M + dt*alpha*K;

% Initialize solution matrix
U = zeros(N-1, Nt+1);
U(:,1) = u0(2:end-1);  % interior nodes only

% Time stepping
for n = 1:Nt
    rhs = M*U(:,n);
    U(:,n+1) = A\rhs;
end

% Build full solution including boundaries
Ugrid = zeros(N+1, Nt+1);
for j = 1:Nt+1
    Ugrid(:,j) = [0; U(:,j); 0]; % add boundaries
end

% --- 3D surface plot u(t,x) ---
[Tgrid, Xgrid] = meshgrid(tvec, x); % Time on x-axis, space on y-axis
figure('Units','normalized','Position',[0.1 0.1 0.65 0.65]);
surf(Tgrid, Xgrid, Ugrid, 'EdgeColor','none');
xlabel('t'); ylabel('x'); zlabel('u(t,x)');
title('3D FEM solution of the Heat Equation');
```

```
colormap parula; colorbar;
view(45,30); shading interp;

% --- 2D snapshots for better visualization ---
figure;
plot(x, Ugrid(:,1), 'k-', 'LineWidth',1.5); hold on;
plot(x, Ugrid(:,round(0.3/dt)), 'r--', 'LineWidth',1.5);
plot(x, Ugrid(:,round(0.6/dt)), 'b-.', 'LineWidth',1.5);
plot(x, Ugrid(:,end), 'g-', 'LineWidth',1.5);
xlabel('x'); ylabel('u(x,t)');
legend('t=0','t=0.3','t=0.6','t=1','Location','northeast');
title('Snapshots of FEM solution at different times');
grid on;
```

We obtain the following illustrations



(a) Temperature profiles at a distinct $t = -0, 0.3, 0.6, 1$



(b) 3D Evolution of $u(x,t)$ over time and space.

Figure 6.6: Numerical solution of the 1D heat equation using FEM with Backward Euler scheme.

Figure illustrates the finite element solution of the 1D heat equation. Subfigure 6.6a shows the temperature distribution at selected times. The profiles clearly decay with time, reflecting the diffusion of heat and the system's tendency toward equilibrium. Subfigure 6.6b presents the space–time evolution of the solution. The surface plot highlights the smooth decrease of $u(x,t)$ in both space and time, confirming the dissipative character of the heat equation. Overall, the results demonstrate the stability and effectiveness of the FEM combined with the Backward Euler scheme.

Conclusion

In this handout, we explored three fundamental numerical methods for solving partial differential equations: ***the finite difference method (FDM), the finite volume method (FVM)****, and the finite element method (FEM). Each approach was applied to a range of representative models, including elliptic equations, hyperbolic wave equations, parabolic heat equation, and transport problems.*

The finite difference method provided a direct and intuitive framework for discretizing derivatives on structured grids. The finite volume method emphasized conservation principles and was particularly effective for flux-based formulations, as seen in the elliptic and transport equations. The finite element method, built on variational formulations, offered greater flexibility and accuracy, especially for the heat equation.

Through these case studies, we highlighted not only the implementation details but also the strengths and limitations of each method. Together, they form a comprehensive toolbox for the numerical approximation of PDEs, adaptable to different classes of problems in applied mathematics and engineering.

Bibliography

[1] S. C. Brenner and L. R. Scott. The Mathematical Theory of Finite Element Methods. 3rd edition, Springer, 2008.

[2] P. G. Ciarlet. The Finite Element Method for Elliptic Problems. North Holland, Amsterdam, 1978.

[3] R. Eymard, T. Gallouët, and R. Herbin. The Finite Volume Method. In Handbook of Numerical Analysis, Vol. VII, Elsevier, 2000.

[4] Steven T. Karris. Numerical Analysis Using MATLAB and Excel. Orchard Publications, 2007.

[5] R. J. LeVeque. Finite Difference Methods for Ordinary and Partial Differential Equations. SIAM, 2007.