Introduction to Finite Difference Approximations

1. Introduction

Finite difference methods are numerical techniques used to approximate derivatives of functions. They are widely used in solving differential equations numerically, especially in engineering and physics applications.

2. The Grid and Notation

Assume we are given a smooth function $f(x)$ defined on an interval $[a, b]$. We discretize the interval into grid points:

$$x_i = a + ih, \quad i = 0, 1, 2, \ldots, N,$$

where $h = \frac{b-a}{N}$ is the uniform grid spacing.

We denote the function values at the grid points as $f_i = f(x_i)$.

3. Finite Difference Approximations

3.1 Forward Difference

The forward difference approximation of the first derivative at x_i is:

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

This formula has a local truncation error of $\mathcal{O}(h)$.

3.2 Backward Difference

The backward difference approximation of the first derivative is:

$$f'(x_i) \approx \frac{f(x_i) - f(x_{i-1})}{h},$$

also with an error of $\mathcal{O}(h)$.

3.3 Central Difference

The central difference approximation is more accurate:

$$f'(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1})}{2h},$$

with a truncation error of $\mathcal{O}(h^2)$.

3.4 Second Derivative Approximation

The second derivative can be approximated using:

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

which also has an error of $\mathcal{O}(h^2)$.

4. Derivation via Taylor Series

Taylor expanding $f(x_{i+1})$ and $f(x_{i-1})$ around x_i gives:

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \frac{h^3}{6}f^{(3)}(x_i) + \dots,$$

$$f(x_{i-1}) = f(x_i) - hf'(x_i) + \frac{h^2}{2}f''(x_i) - \frac{h^3}{6}f^{(3)}(x_i) + \dots.$$

Subtracting these gives the central difference formula, and combining them gives the second derivative approximation.

5. Example

Let $f(x) = \sin(x)$ on $[0, \pi]$ with $h = \frac{\pi}{4}$. Compute $f'(x)$ at $x = \frac{\pi}{4}$ using central differences:

$$f'(x_i) \approx \frac{\sin\left(\frac{\pi}{2}\right) - \sin(0)}{2h} = \frac{1 - 0}{2 \cdot \frac{\pi}{4}} = \frac{1}{\frac{\pi}{2}} = \frac{2}{\pi} \approx 0.6366.$$

The true derivative is $\cos\left(\frac{\pi}{4}\right) \approx 0.7071$, so the approximation is reasonable.

6. Error Analysis

The error in finite difference approximations comes from truncating the Taylor series. For the central difference of the first derivative:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{h^2}{6}f^{(3)}(x_i) + \mathcal{O}(h^4).$$

Hence, the error is proportional to h^2.

8. Summation By Parts (SBP) Operators

8.1 Motivation

In the continuous setting, the integration by parts identity plays a crucial role in energy estimates and stability analysis:

$$\int_a^b u(x)\frac{dv(x)}{dx}\,dx = [u(x)v(x)]_a^b - \int_a^b \frac{du(x)}{dx}v(x)\,dx.$$

To mimic such stability-preserving structures in the discrete setting, we introduce **Summation By Parts** (SBP) finite difference operators.

8.2 Discrete Setting

Let $u = (u_0, u_1, \ldots, u_N)^T$ and similarly for v, representing function values at equally spaced grid points $x_i = a + ih$ for $i = 0, \ldots, N$. An SBP operator consists of:

- A diagonal, symmetric, positive-definite norm matrix H approximating the L^2 inner product:
$$\langle u, v\rangle_H = u^T H v \approx \int_a^b u(x)v(x)\,dx.$$
- A derivative matrix D such that the following discrete summation by parts identity holds:
$$u^T HDv + v^T HDu = u^T Bv,$$
where B is a boundary matrix representing the discrete analog of evaluating uv at the boundary:
$$B = \mathrm{diag}(-1, 0, 0, \ldots, 0, 1).$$

10. Explicit SBP Operator: First-Order at Boundaries, Second-Order in Interior

We now give an explicit example of a finite difference Summation By Parts (SBP) operator for the first derivative on a uniform grid of $N+1$ points $x_0, x_1, \ldots, x_N$ with spacing h.

10.1 Norm Matrix H

The norm matrix H is diagonal and corresponds to the trapezoidal rule:

$$H = h \cdot \mathrm{diag}\left(\tfrac{1}{2}, 1, 1, \ldots, 1, \tfrac{1}{2}\right).$$

In matrix form for a small grid (e.g. $N+1 = 6$):

$$H = h \cdot \begin{bmatrix} \frac{1}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{2} \end{bmatrix}.$$

10.2 Derivative Matrix D

The derivative matrix D is defined as follows:

- At the left boundary (forward difference):

$$D_0 = \begin{bmatrix} -\frac{1}{h}, & \frac{1}{h}, & 0, \ldots, 0 \end{bmatrix}.$$

- At the right boundary (backward difference):

$$D_N = \begin{bmatrix} 0, \ldots, 0, & -\frac{1}{h}, & \frac{1}{h} \end{bmatrix}.$$

- In the interior (central difference):

$$D_i = \begin{bmatrix} \ldots, -\frac{1}{2h}, & 0, & \frac{1}{2h}, \ldots \end{bmatrix}, \quad \text{for } 1 \leq i \leq N-1.$$

Explicitly, for $N+1 = 6$, the derivative matrix is:

$$D = \frac{1}{h} \cdot \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & 0 \\ -1/2 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & -1/2 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & -1/2 & 0 & 1/2 & 0 \\ 0 & 0 & 0 & -1/2 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

10.3 SBP Identity

With the above matrices, the following discrete summation by parts identity holds:

$$u^T H D v + v^T H D u = u_0 v_0 - u_N v_N.$$

This mimics the continuous integration by parts formula and ensures discrete stability for PDE solvers.

8.5 Importance in PDE Solvers

SBP operators allow for stable finite difference discretizations of PDEs, particularly in conjunction with the *Simultaneous Approximation Term (SAT)* method for weakly enforcing boundary conditions. The SBP-SAT framework is widely used in high-order numerical solvers.

8.6 Higher-Order SBP Operators

Higher-order SBP operators exist and are characterized by:

- High-order accuracy in the interior of the domain.
- Lower-order accuracy near boundaries.
- Structured H and D matrices ensuring stability via the SBP identity.

9. Conclusion

SBP operators are essential tools for stable high-order finite difference methods. By mimicking the integration by parts property discretely, they enable accurate and stable solutions of PDEs, especially in domains with boundaries.

11. The Method of Lines (MoL)

11.1 Overview

The **Method of Lines (MoL)** is a powerful technique for solving time-dependent partial differential equations (PDEs). The idea is to discretize the spatial variables while keeping time continuous, leading to a system of ordinary differential equations (ODEs) in time:

$$\frac{d\mathbf{u}(t)}{dt} = \mathbf{F}(\mathbf{u}(t)),$$

where $\mathbf{u}(t)$ is the vector of the solution at the spatial grid points and $\mathbf{F}$ is the spatial discretization (e.g., via finite differences, spectral methods, or finite elements).

Once the spatial discretization is fixed, the resulting ODE system can be integrated in time using standard ODE solvers such as Runge–Kutta or multistep methods.

11.2 Spatial Discretization Example

Consider the 1D linear advection equation:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \quad a > 0.$$

Discretizing space using a finite difference method (e.g., SBP operators) yields a semi-discrete system:

$$\frac{d\mathbf{u}}{dt} = -aD\mathbf{u},$$

where D is a finite difference operator approximating ∂_x.

11.3 Energy and Stability

For linear hyperbolic problems, energy conservation (or dissipation control) is critical for stability. Suppose we define a discrete energy norm using a symmetric positive-definite matrix H, such that the discrete energy is:

$$E(t) = \mathbf{u}(t)^T H \mathbf{u}(t).$$

Taking the derivative in time:

$$\frac{dE}{dt} = 2\mathbf{u}^T H \frac{d\mathbf{u}}{dt} = -2a\mathbf{u}^T H D\mathbf{u}.$$

If D and H satisfy the Summation By Parts (SBP) property:

$$\mathbf{u}^T H D\mathbf{u} + \mathbf{u}^T H D^T \mathbf{u} = \mathbf{u}^T B\mathbf{u},$$

then we get a discrete analog of energy conservation (up to boundary terms). This shows that the spatial discretization preserves energy *semidiscretely*.

11.4 Why Use a 3rd-Order Runge–Kutta or Higher

While the spatial discretization may preserve energy, careless time discretization can introduce numerical dissipation or even instability. To maintain energy conservation at the discrete level, we must match the time integrator's accuracy to the energy dynamics.

- Low-order (e.g., Euler or 2nd-order RK) time integrators can introduce significant artificial energy growth or decay, even for stable spatial discretizations.
- It has been shown (see e.g., [Gottlieb et al. 2001]) that **at least a 3rd-order explicit Runge–Kutta method** is required to preserve energy (or boundedness of the discrete norm) when applied to semidiscrete hyperbolic systems.

11.5 Example: Energy Growth with Euler

Using forward Euler:

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \cdot \mathbf{F}(\mathbf{u}^n),$$

we get:

$$E^{n+1} = \mathbf{u}^{n+1,T} H \mathbf{u}^{n+1} = \mathbf{u}^{n,T} H \mathbf{u}^n + \mathcal{O}(\Delta t),$$

but this approximation can grow or decay depending on the structure of $\mathbf{F}$. Thus, low-order time methods do not preserve the semidiscrete energy structure.

11.6 Summary

To preserve the stability and energy properties of spatially discretized hyperbolic PDEs:

- Use energy-stable spatial discretizations (e.g., SBP operators),
- Employ time integrators that do not destroy the energy structure,
- Use at least 3rd-order Runge–Kutta methods for accurate and stable time integration.

12. Example: Energy Behavior with Third-Order Runge–Kutta (RK3)

12.1 Setup: Linear Advection Equation

Consider the 1D linear advection equation on a periodic domain:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0, \quad x \in [0,1], \quad a > 0.$$

We discretize the spatial derivative using a finite difference operator D, e.g., an SBP operator, to obtain a semi-discrete system:

$$\frac{d\mathbf{u}}{dt} = -aD\mathbf{u}.$$

Let H be a diagonal norm matrix associated with the SBP operator, and define the discrete energy:

$$E(t) = \mathbf{u}(t)^T H \mathbf{u}(t).$$

12.2 Time Integration Using RK3

We now integrate in time using the classical third-order Runge–Kutta method:

$$\begin{aligned}
\mathbf{u}^{(1)} &= \mathbf{u}^n + \Delta t \cdot \mathbf{F}(\mathbf{u}^n), \\
\mathbf{u}^{(2)} &= \frac{3}{4}\mathbf{u}^n + \frac{1}{4}\left(\mathbf{u}^{(1)} + \Delta t \cdot \mathbf{F}(\mathbf{u}^{(1)})\right), \\
\mathbf{u}^{n+1} &= \frac{1}{3}\mathbf{u}^n + \frac{2}{3}\left(\mathbf{u}^{(2)} + \Delta t \cdot \mathbf{F}(\mathbf{u}^{(2)})\right),
\end{aligned}$$

where $\mathbf{F}(\mathbf{u}) = -aD\mathbf{u}$.

12.3 Energy Update

Define $E^n = \mathbf{u}^{nT} H \mathbf{u}^n$ and $E^{n+1} = \mathbf{u}^{n+1T} H \mathbf{u}^{n+1}$.

For the linear advection case, one can show that:

$$E^{n+1} = E^n + \mathcal{O}(\Delta t^4),$$

meaning that RK3 preserves the discrete energy up to fourth-order accuracy in time. This contrasts with Forward Euler, for which:

$$E^{n+1} = E^n + \mathcal{O}(\Delta t),$$

which leads to significant artificial energy growth or decay over time.

12.4 Numerical Observation

Numerical experiments confirm that:

- Using Forward Euler results in a monotonic increase in energy (numerical instability).
- RK2 improves stability but still introduces noticeable energy error.
- RK3 preserves the energy almost exactly for many time steps, especially when paired with an SBP operator.

12.5 Conclusion

The third-order Runge–Kutta method provides sufficient accuracy in time to maintain the energy stability granted by SBP discretizations in space. For hyperbolic problems and long-time simulations, RK3 or higher-order time integrators are essential to avoid artificial energy drift and ensure numerical stability.

13. Hyperbolicity in $1+1$ Dimensions

13.1 General Form of a First-Order System

Consider a general linear first-order system of PDEs in $1+1$ dimensions (one time and one space dimension):

$$\partial_t \mathbf{u} + A\partial_x \mathbf{u} = 0,$$

where $\mathbf{u}(t,x) \in \mathbb{R}^n$ is the vector of unknowns and $A \in \mathbb{R}^{n\times n}$ is a constant coefficient matrix. This system arises, for example, after linearization and/or first-order reduction of a more general PDE.

13.2 Definition: Hyperbolicity

The nature of the solutions and the well-posedness of the initial value problem (IVP) depend critically on the properties of the matrix A. The classification is as follows:

- **Strongly Hyperbolic**: The matrix A is diagonalizable with real eigenvalues. That is, there exists an invertible matrix S such that:

$$A = S\Lambda S^{-1}, \quad \Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \quad \lambda_i \in \mathbb{R}.$$

Strong hyperbolicity ensures well-posedness of the IVP in the sense of Hadamard, and allows energy estimates and stable numerical schemes.

- **Weakly Hyperbolic**: All eigenvalues of A are real, but A is not diagonalizable. This typically means there exists at least one eigenvalue with fewer linearly independent eigenvectors than its algebraic multiplicity. In this case, the system lacks a complete set of eigenvectors.

  Weakly hyperbolic systems may have solutions that grow polynomially in time, and numerical schemes are often unstable, even when consistent.

- **Not Hyperbolic (Ill-posed)**: The matrix A has complex eigenvalues. This leads to exponential growth of high-frequency modes, and the IVP is ill-posed.

13.3 Example: Testing Hyperbolicity

Let us illustrate with a 2×2 system:

$$\partial_t \begin{bmatrix} u \\ v \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \partial_x \begin{bmatrix} u \\ v \end{bmatrix} = 0.$$

We compute the eigenvalues and eigenvectors of A:

- Characteristic polynomial:

$$\det(A - \lambda I) = (1 - \lambda)^2 = 0 \quad \Rightarrow \lambda = 1 \text{ (double eigenvalue)}.$$

- Eigenvectors: Solve $(A - I)\mathbf{v} = 0$:

$$A - I = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \Rightarrow \quad \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = 0 \quad \Rightarrow v_2 = 0.$$

  So the only eigenvector is $\begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Hence, there is only *one* linearly independent eigenvector for a repeated eigenvalue.

- Conclusion: The system is **weakly hyperbolic**.

13.4 Consequences of Weak Hyperbolicity

Although the system has real eigenvalues (suggesting finite-speed signal propagation), the lack of diagonalizability leads to:

- No well-posedness in the energy norm,
- Polynomial growth of solutions over time,
- Instabilities in numerical schemes, even with SBP operators or dissipation,
- No well-defined characteristic variables.

13.5 Summary

To assess hyperbolicity of a linear $1+1$ PDE system:

1. Compute the eigenvalues of the principal matrix A.
2. Check if all eigenvalues are real.
3. Check if there is a complete basis of linearly independent eigenvectors.
4. Classify:
   - Strongly hyperbolic: real, complete set of eigenvectors.
   - Weakly hyperbolic: real, but incomplete set of eigenvectors.
   - Ill-posed: complex eigenvalues.

Strong hyperbolicity is essential for constructing well-posed PDE models and designing stable finite difference or spectral methods.

14. Boundary Treatment in $1+1$ Dimensions for Strongly Hyperbolic Systems

14.1 Setup: First-Order Hyperbolic System

Consider a linear, constant-coefficient, first-order system in $1+1$ dimensions:

$$\partial_t \mathbf{u} + A\partial_x \mathbf{u} = 0, \quad x \in [0, L],$$

where $\mathbf{u}(t, x) \in \mathbb{R}^n$, and $A \in \mathbb{R}^{n\times n}$ is a diagonalizable matrix with real eigenvalues. This ensures the system is **strongly hyperbolic**.

14.2 Characteristic Variables

Let $A = S\Lambda S^{-1}$, where $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_n)$, with $\lambda_i \in \mathbb{R}$, and S is the matrix of right eigenvectors. Define the **characteristic variables**:

$$\mathbf{w} = S^{-1}\mathbf{u}.$$

The system becomes:

$$\partial_t \mathbf{w} + \Lambda\partial_x \mathbf{w} = 0,$$

which decouples into n scalar advection equations:

$$\partial_t w_i + \lambda_i \partial_x w_i = 0, \quad i = 1, \ldots, n.$$

14.3 Boundary Conditions Using Characteristics

At a boundary (e.g., $x = 0$), the sign of λ_i determines whether the characteristic variable w_i is incoming or outgoing:

- If $\lambda_i > 0$: w_i moves from left to right — **incoming at $x = 0$**.
- If $\lambda_i < 0$: w_i moves from right to left — **outgoing at $x = 0$**.

To ensure a well-posed boundary value problem, we must impose **exactly as many boundary conditions** as there are incoming characteristics.

Example: Suppose $\lambda_1 > 0, \lambda_2 < 0$. At $x = 0$, only w_1 is incoming. Then we impose:

$$w_1(t, x = 0) = g(t),$$

for some prescribed boundary data $g(t)$. No condition is imposed on w_2, since it is outgoing.

14.4 Returning to Physical Variables

To impose the boundary condition in physical variables $\mathbf{u}$, express:

$$\mathbf{u}(t, 0) = S\mathbf{w}(t, 0).$$

So, knowing $w_1(t, 0) = g(t)$, and letting $w_2(t, 0)$ evolve freely, we reconstruct:

$$\mathbf{u}(t, 0) = S \begin{bmatrix} g(t) \\ w_2(t, 0) \end{bmatrix}.$$

14.5 Numerical Boundary Implementation

In numerical schemes (e.g., finite differences), it is common to implement boundary conditions weakly using penalty terms or SATs (Simultaneous Approximation Terms). These enforce the incoming characteristics without over-constraining the system.

For example, using the penalty method:

$$\frac{d\mathbf{u}}{dt} = -AD\mathbf{u} + \tau S \left(\mathbf{w}_{\text{BC}} - \mathbf{w}_{\text{num}}\right) \delta_0,$$

where:

- D is the spatial derivative operator (e.g., SBP),
- $\mathbf{w}_{\text{BC}}$ contains prescribed values for incoming components,
- δ_0 is a vector selecting the boundary node,
- τ is a penalty strength, tuned for stability.

14.6 Summary

Boundary treatment for strongly hyperbolic systems should follow these principles:

- Diagonalize the system to find characteristic variables.
- Impose boundary conditions only on incoming characteristics.
- Avoid over-specifying (causes instability) or under-specifying (causes ill-posedness).
- Use numerical methods (e.g., penalty or SAT) that respect the characteristic structure for stable discretizations.

15. Boundary Treatment for the Scalar Advection Equation

15.1 The Scalar Advection Equation

Consider the scalar advection equation:

$$\partial_t u + a\,\partial_x u = 0, \quad x \in [0, L], \quad a \in \mathbb{R}.$$

This is a first-order linear hyperbolic PDE. The sign of the advection speed a determines the direction of information flow.

15.2 Characteristics and Direction of Propagation

The solution propagates along characteristic curves defined by:

$$\frac{dx}{dt} = a.$$

Hence:

- If $a > 0$, information flows from left to right. The boundary at $x = 0$ is **inflow**, and $x = L$ is **outflow**.
- If $a < 0$, information flows from right to left. The boundary at $x = L$ is **inflow**, and $x = 0$ is **outflow**.

15.3 Boundary Conditions

To ensure well-posedness, we impose a boundary condition only at the **inflow boundary**:

- For $a > 0$: impose
$$u(t, 0) = g(t),$$
where $g(t)$ is prescribed data.

- For $a < 0$: impose

$$u(t, L) = h(t).$$

No condition is imposed at the outflow boundary; the solution there is determined by interior values and the PDE.

15.4 Numerical Implementation (Penalty Method)

Discretize the domain using $N+1$ grid points $x_i \in [0, L]$, and let $\mathbf{u}(t) = [u_0(t), u_1(t), \ldots, u_N(t)]^T$ be the discrete solution vector. Suppose we use a Summation-By-Parts (SBP) derivative operator D with norm matrix H.

We implement the semi-discrete equation:

$$\frac{d\mathbf{u}}{dt} = -aD\mathbf{u} + \tau H^{-1} e_0 (g(t) - u_0),$$

where:

- $e_0 = [1, 0, \ldots, 0]^T$ selects the first grid point,
- u_0 is the solution at $x = 0$,
- $\tau > 0$ is a penalty parameter ensuring stability.

15.5 Energy Estimate

Define the discrete energy:

$$E(t) = \mathbf{u}^T H \mathbf{u}.$$

Differentiating:

$$\frac{dE}{dt} = -2a\mathbf{u}^T HD\mathbf{u} + 2\tau(g(t) - u_0)u_0.$$

Using SBP property:

$$\mathbf{u}^T HD\mathbf{u} = \frac{1}{2}u_0^2 - \frac{1}{2}u_N^2,$$

Thus:

$$\frac{dE}{dt} = -au_0^2 + au_N^2 + 2\tau(g(t) - u_0)u_0.$$

Choosing $\tau = a$ ensures cancellation of the u_0^2 terms:

$$\frac{dE}{dt} = au_N^2 + 2a(g(t) - u_0)u_0 - au_0^2 = au_N^2 + a(2g(t)u_0 - 2u_0^2 - u_0^2).$$

With a careful choice of τ, this yields a dissipative or non-increasing energy estimate, ensuring numerical stability.

15.6 Summary

- The scalar advection equation has a single characteristic with speed a.
- A boundary condition is imposed only at the inflow boundary.
- Using SBP operators with penalty methods allows stable enforcement of boundary conditions.
- Stability is verified through a discrete energy estimate.

16. High-Resolution Shock-Capturing (HRSC) Methods

16.1 Motivation

Nonlinear hyperbolic conservation laws often develop discontinuities (shocks) in finite time, even when starting from smooth initial data. Traditional high-order numerical methods suffer from spurious oscillations near these discontinuities due to the Gibbs phenomenon.

To address this, **High-Resolution Shock-Capturing (HRSC)** methods are designed to:

- Capture shocks and discontinuities without spurious oscillations,
- Maintain high-order accuracy in smooth regions,
- Preserve conservation form to ensure the correct shock speed (via the Rankine-Hugoniot conditions),
- Be Total Variation Diminishing (TVD) or at least non-increasing in total variation to suppress numerical instabilities.

16.2 General Form of Conservation Laws

Consider a scalar conservation law:

$$\partial_t u + \partial_x f(u) = 0,$$

where $f(u)$ is a nonlinear flux function.

16.3 Finite Volume Framework

Divide the domain into cells $I_j = [x_{j-1/2}, x_{j+1/2}]$. The cell average is:

$$\bar{u}_j(t) = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(x,t)\, dx.$$

The semi-discrete finite volume scheme is:

$$\frac{d\bar{u}_j}{dt} = -\frac{1}{\Delta x}\left(F_{j+1/2} - F_{j-1/2}\right),$$

where $F_{j\pm1/2}$ are numerical fluxes approximating $f(u)$ at cell interfaces.

16.4 Numerical Fluxes and Limiters

To prevent oscillations and maintain high resolution:

- Use a **monotonicity-preserving flux** (e.g., Lax-Friedrichs, HLL, Roe).
- Reconstruct interface values using slope limiters (e.g., Minmod, MC, Superbee) or higher-order WENO schemes.

The numerical flux often uses a reconstruction:

$$F_{j+1/2} = F\left(u^L_{j+1/2}, u^R_{j+1/2}\right),$$

where $u^L_{j+1/2}$, $u^R_{j+1/2}$ are left and right reconstructed states at the interface.

16.5 Example: Burgers' Equation

The inviscid Burgers' equation:

$$\partial_t u + \partial_x\left(\frac{1}{2}u^2\right) = 0,$$

models nonlinear advection with flux $f(u) = \frac{1}{2}u^2$.

This equation develops shocks even from smooth initial data.

Numerical Scheme:

1. Use a finite volume discretization:
$$\frac{d\bar{u}_j}{dt} = -\frac{1}{\Delta x}\left(F_{j+1/2} - F_{j-1/2}\right).$$
2. Use the **Rusanov (local Lax-Friedrichs)** flux:
$$F_{j+1/2} = \frac{1}{2}\left[f(u^L_{j+1/2}) + f(u^R_{j+1/2})\right] - \frac{\alpha_{j+1/2}}{2}\left(u^R_{j+1/2} - u^L_{j+1/2}\right),$$
where $\alpha_{j+1/2} = \max\left(|u^L_{j+1/2}|, |u^R_{j+1/2}|\right)$ is the local maximum wavespeed.
3. Use a slope limiter for linear reconstruction:
$$u^L_{j+1/2} = \bar{u}_j + \frac{1}{2}\phi(r_j)(\bar{u}_j - \bar{u}_{j-1}), \quad u^R_{j+1/2} = \bar{u}_{j+1} - \frac{1}{2}\phi(r_{j+1})(\bar{u}_{j+1} - \bar{u}_j),$$
with
$$r_j = \frac{\bar{u}_j - \bar{u}_{j-1}}{\bar{u}_{j+1} - \bar{u}_j}, \quad \phi(r) = (1, r).$$

16.6 Key Properties

- **Conservation**: The scheme is conservative, ensuring correct shock speed.
- **Shock capturing**: No need to explicitly track shocks.
- **Non-oscillatory**: Limiters prevent spurious oscillations near discontinuities.
- **Accuracy**: High-order accuracy in smooth regions, low-order near discontinuities.

16.7 Summary

HRSC methods provide robust and accurate tools for solving nonlinear conservation laws. Using carefully designed fluxes and limiters, they handle discontinuities naturally while preserving accuracy in smooth regions.

1. Shock-Capturing with High-Resolution Methods

In high-resolution shock-capturing methods, shock discontinuities are resolved without introducing spurious oscillations. Below is an example illustrating the shock formation in Burgers' equation using a high-resolution scheme.

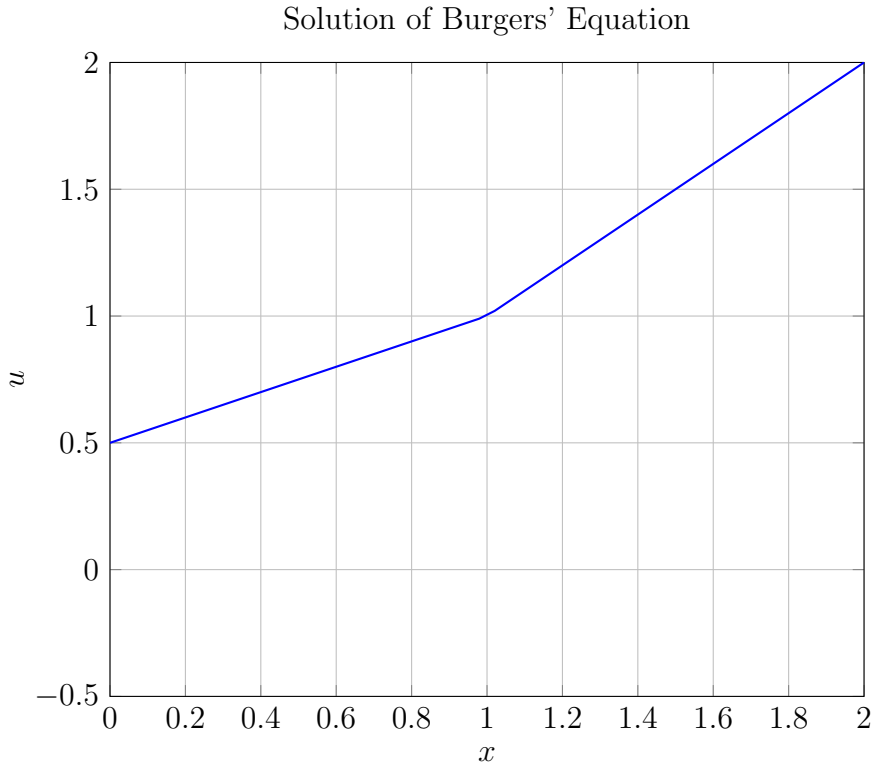

Figure 1: Shock formation in Burgers' equation with high-resolution shock-capturing.

2. Boundary Treatment Using Characteristics

Consider a hyperbolic system where we apply boundary conditions at inflow and outflow boundaries. The solution is described using characteristics, which depend on the eigenvalues of the system. For a first-order system, the characteristic lines define the directions of information propagation.

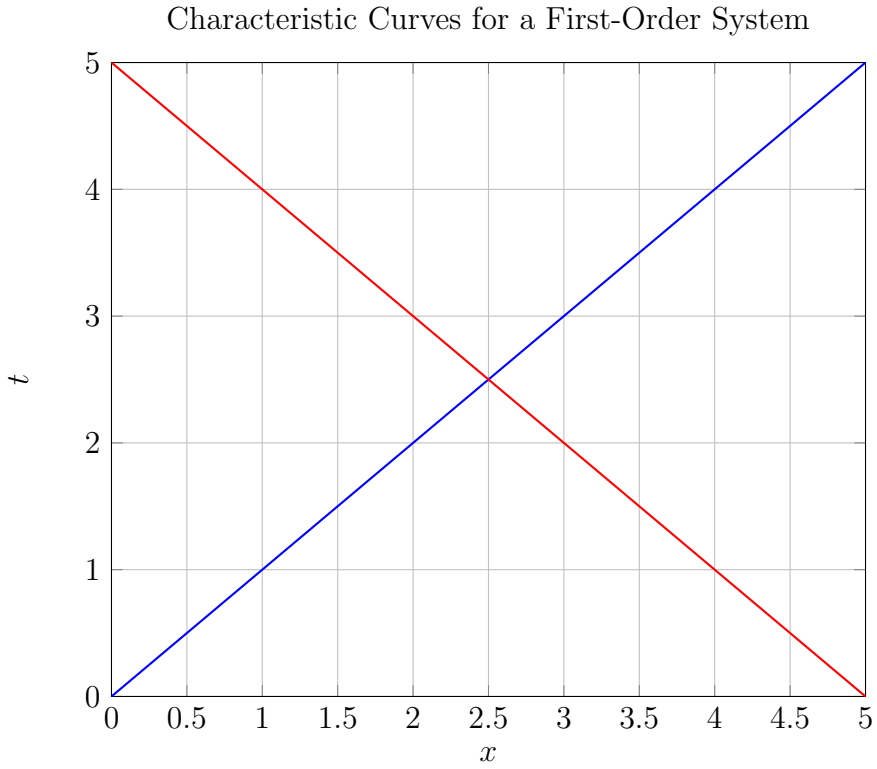

Figure 2: Characteristic curves for a first-order system with $a > 0$ (blue) and $a < 0$ (red).

3. Energy Estimates in Numerical Methods

Energy estimates are used to show the stability of numerical methods. Below is an illustration of how energy evolves in a numerical scheme for an advection equation with an imposed boundary condition.

4. Total Variation Diminishing (TVD)

The total variation diminishing property ensures that the numerical method doesn't introduce new oscillations. Below is an illustration of how a TVD method behaves on a simple test case.

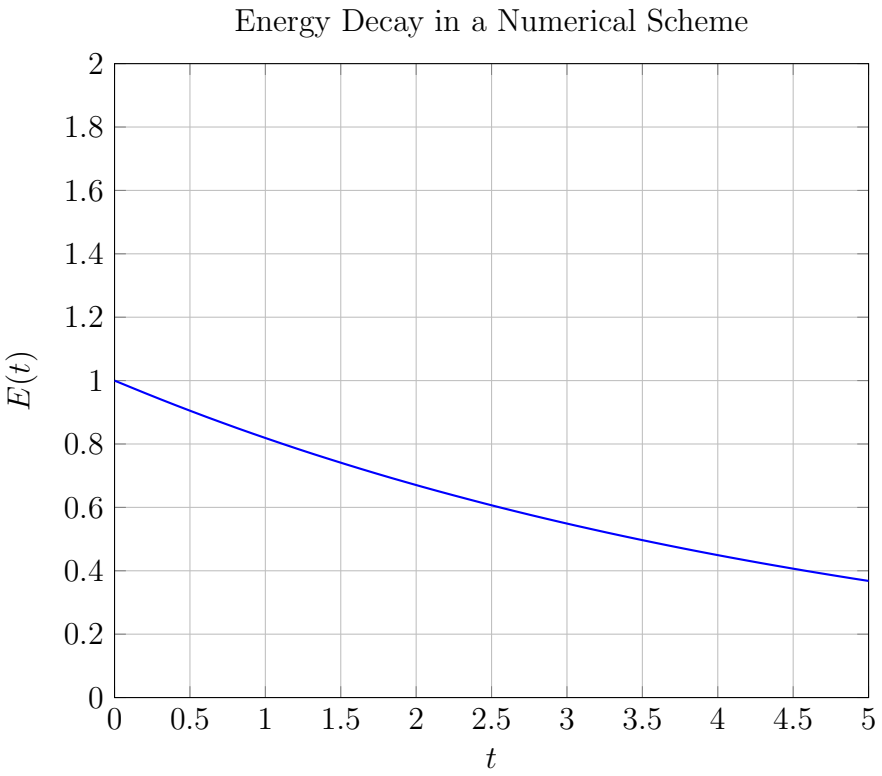

Figure 3: Energy decay over time, showing dissipation in a numerical scheme with boundary conditions.

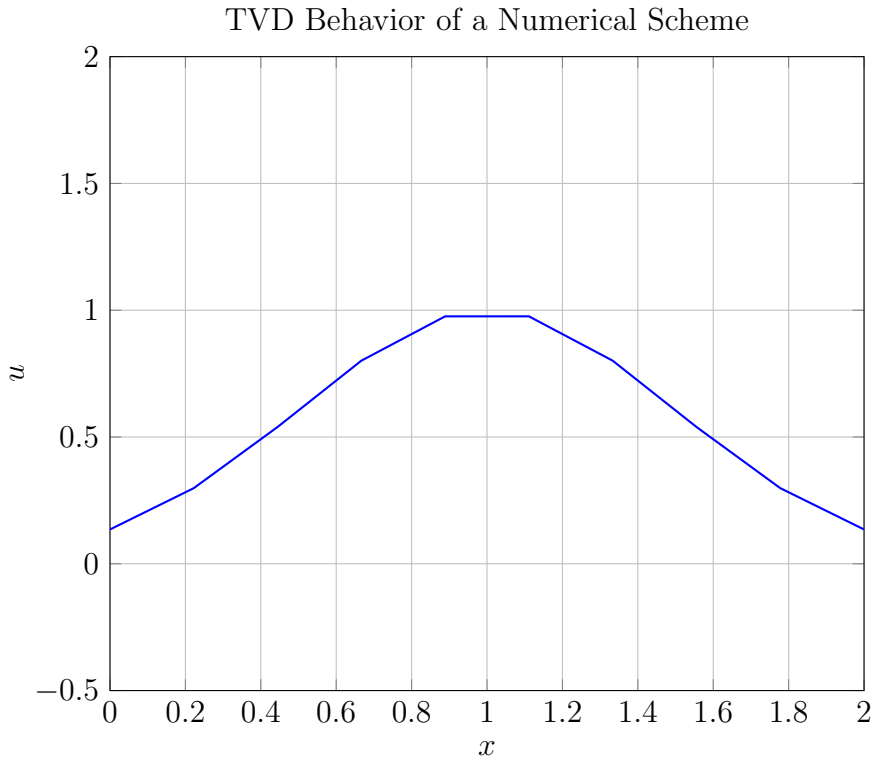

Figure 4: TVD behavior of a numerical scheme showing smooth solution without oscillations.