An Introduction to Finite Difference Methods

A Tour Through Partial Differential Equations

Manuel A. Diaz

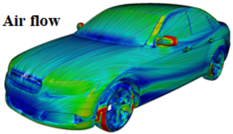
InspiraSTEM Internship

May 19, 2026

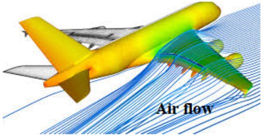
Outline

The Motivation

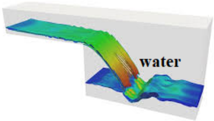
Applications of CFD



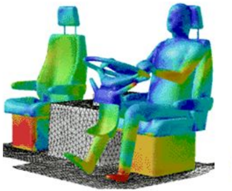
Automobile



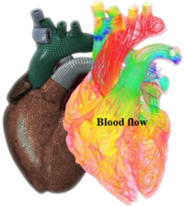
Aerospace



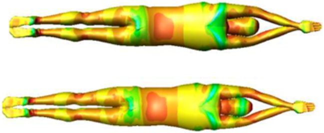
Water fall



Ventillation



Biomedical

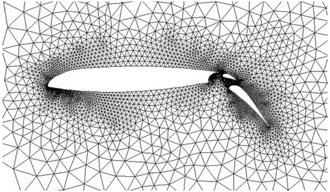


Swinning

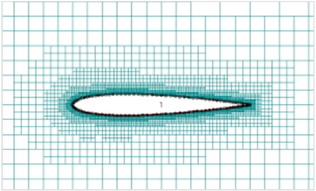
NOTES:

- Modeling compressible flows is a challenging task.
- The gap between theory and practice is still large.
- Much of what we "know" in compressible flow is valid only under limiting assumptions – inviscid flow, perfect gas, steady state, simple geometry. Real engineering systems violate all of these simultaneously. Bridging that gap is still ongoing work.
- So rather than knowing everything, we have a solid classical framework (Rankine-Hugoniot, method of characteristics, boundary layer theory) built over the past century.
- To date the most challenging problem is the modeling of turbulence. This is because turbulence is a complex phenomenon that requires larga-scale numerical simulations. This is because of the mesh resolution required to resolve the small scales of turbulence.

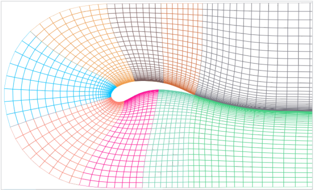
The Motivation



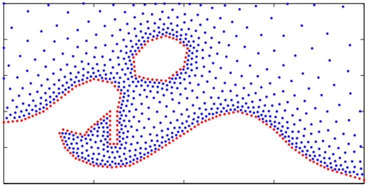
(a) Unstructured grid.



(b) Mesh refinement.

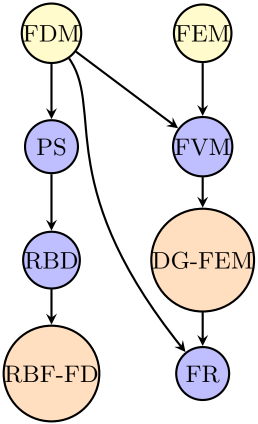


(c) Blocked mesh.



(d) Mesh free.

Figure: Different philosophies of domain discretization



PDE approaches
(Randall J. LeVeque 2002; Morton and Mayers 2005; Moin 2010)

NOTES:

- We are interested in solving partial differential equations (PDEs) numerically.
- However, not all discretization methods are suitable for all PDEs. For example,
  - All methods struggle with strong gradients like shocks,
  - Finite volume methods are not suitable for diffusive problems,
  - Finite element methods are not suitable for convection-dominated problems,
  - Discontinuous Galerkin methods are very expensive for diffusion problems,
  - and so on...
- However, finite difference methods (FDM) are suitable for a wide range of PDEs, including hyperbolic, parabolic, and elliptic equations.
- Moreover, explicit FDMs of moderate order are very easy to implement and very efficient.
- Because of this, in this project we will use finite difference methods as our numerical tool of choice.
- But what PDEs are we going to solve?

Outline

1. Motivation
2. Problem Formulation
3. The Method
4. Putting things together
5. How do we know it is right?
6. Main Takeaways

The First Problem

On a bounded domain $\Omega \in]0, L]$ and a time interval $t \in]0, T]$, we wish to solve:

The one-dimensional problem

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0,$$

where $u(x, t)$ is the solution, $c > 0$ is the speed of the wave, and x and t are the spatial and time variables, respectively. The initial condition is given by:

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

and the boundary condition $(\partial\Omega)$ is given by:

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

where $f(x)$ is the initial condition and $g(t)$ is the boundary condition.

NOTES:

For the sake of simplicity, we will consider a one-dimensional problem without naming it or describing what it represents physically. Instead, I want to focus on the mathematical aspects of the problem. For that let me re-write the problem as

The one-dimensional problem

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \quad \Longrightarrow \quad \frac{\partial u}{\partial t} = -c\frac{\partial u}{\partial x},$$

which means that any rate of change of u is proportional to the negative of the spatial derivative of u. This condition is so strict that it is trivially satisfied when u is constant. It also means that any change in u over time must be offset by the spatial derivative of u in order to avoid introducing or removing information from the domain. For this reason, this problem is called a **conservative** problem.

As long as $c > 0$ is a constant, the problem is linear and can be solved analytically.

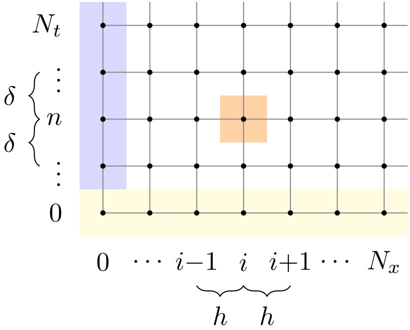
- Can it be solved for a general $f(x)$ and $g(t)$?
  The answer is no, because we cannot guarantee that for any $f(x)$ and $g(t)$, the problem is well-posed. But if we choose a specific $f(x)$ and $g(t)$ suitable for the problem, we can solve it analytically so we can study our numerical methods on it.

Space–time grid

- Many time-dependent PDE problems are solved by introducing a **structured space–time grid**: spatial locations indexed by i and time levels by n.
- Let $x_i = ih$ for $i = 0, 1, \ldots, N_x$ and
- let $t_n = n\delta$ for $n = 0, 1, \ldots, N_t$.
- So that $u_i^n \approx u(x_i, t_n)$.
- Is there a relation between h and δ?

Classical theory for stability and convergence connects h and δ (for example the CFL condition (Courant, Friedrichs, and Lewy 1928)).

For $\Omega \in]0, L]$ and $t \in]0, T]$:



spatial step size is $h = \frac{L}{N_x - 1}$,

time step size is $\delta = \frac{T}{N_t}$.

NOTES:

What is the finite difference method?

The finite difference method (FDM) is a numerical approach to solving Ordinary Differential Equations (ODEs) and Partial Differential Equations (PDEs) in which spatial derivatives are replaced by algebraic approximations. This is the main idea behind any finite difference scheme.

Here is a short video on using FDM to solve 1-d ODEs:
https://youtu.be/to82dv2SX28?si=y_o8FvL_9k5KvzyN
Here is another video on using FDM to solve 2-d Poisson's equation:
https://youtu.be/4IL8n8yYNjw?si=yALsTGaX9zE2TUmK
(I love these videos, because they use the Manim library to create beautiful animations!)

Outline

Taylor expansions

- Let us assume that $u(x)$ is C^4 on $x \in \mathbb{R}$, and:

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h} \tag{1}$$

- In the neighborhood of point x, we can use a Taylor series expansion to approximate $u(x+h)$ as:

$$u(x+h) = u(x) + \frac{h}{1!}u'(x) + \frac{h^2}{2!}u''(x) + \frac{h^3}{3!}u'''(x) + \frac{h^4}{4!}u^{(4)}(x),$$

- Let us assume we are interested in an approximation based only on the first two terms of the previous expression:

$$u(x+h) \approx u(x) + hu'(x) + O(h^2),$$

- The term $O(h^2)$ indicates the error of the approximation is proportional to h^2.
- The approximation is said to be consistent at first order.

NOTES:

- $u(x)$ being C^4 means that the function has four continuous derivatives on $x \in \mathbb{R}$.
- When h is small (but nonzero), the difference quotient on the right-hand side (rhs) is a good approximation of the derivative $u'(x)$.
- If (1) holds, we may assume that u is smooth and its derivatives exist at x; then we can use a Taylor series expansion to approximate u at $x \pm h$.

Perhaps, using grid-index notation, the main idea behind the Taylor series expansion reads more clearly:

$$u_{i\pm1} = u_i \pm hu_i' + \frac{h^2}{2}u_i'' \pm \frac{h^3}{6}u_i''' + \frac{h^4}{24}u^{(4)}(x),$$

Taylor expansions

- Let us assume that $u(x)$ is C^4 on $x \in \mathbb{R}$, and:

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h} \tag{1}$$

- In the neighborhood of point x, we can use a Taylor series expansion to approximate $u(x+h)$ as:

$$u(x+h) = u(x) + \frac{h}{1!}u'(x) + \frac{h^2}{2!}u''(x) + \frac{h^3}{3!}u'''(x) + \frac{h^4}{4!}u^{(4)}(x),$$

- Let us assume we are interested in an approximation based only on the first two terms of the previous expression:

$$u(x+h) \approx u(x) + hu'(x) + O(h^2),$$

- The term $O(h^2)$ indicates the error of the approximation is proportional to h^2.
- The approximation is said to be consistent at first order.

NOTES:

- $u(x)$ being C^4 means that the function has four continuous derivatives on $x \in \mathbb{R}$.
- When h is small (but nonzero), the difference quotient on the right-hand side (rhs) is a good approximation of the derivative $u'(x)$.
- If (1) holds, we may assume that u is smooth and its derivatives exist at x; then we can use a Taylor series expansion to approximate u at $x \pm h$.

Perhaps, using grid-index notation, the main idea behind the Taylor series expansion reads more clearly:

$$u_{i\pm1} = u_i \pm hu_i' + \frac{h^2}{2}u_i'' \pm \frac{h^3}{6}u_i''' + \frac{h^4}{24}u^{(4)}(x),$$

Taylor expansions

- Let us assume that $u(x)$ is C^4 on $x \in \mathbb{R}$, and:

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h} \tag{1}$$

- In the neighborhood of point x, we can use a Taylor series expansion to approximate $u(x+h)$ as:

$$u(x+h) = u(x) + \frac{h}{1!}u'(x) + \frac{h^2}{2!}u''(x) + \frac{h^3}{3!}u'''(x) + \frac{h^4}{4!}u^{(4)}(x),$$

- Let us assume we are interested in an approximation based only on the first two terms of the previous expression:

$$u(x+h) \approx u(x) + hu'(x) + O(h^2),$$

- The term $O(h^2)$ indicates the error of the approximation is proportional to h^2.
- The approximation is said to be consistent at first order.

NOTES:

- $u(x)$ being C^4 means that the function has four continuous derivatives on $x \in \mathbb{R}$.
- When h is small (but nonzero), the difference quotient on the right-hand side (rhs) is a good approximation of the derivative $u'(x)$.
- If (1) holds, we may assume that u is smooth and its derivatives exist at x; then we can use a Taylor series expansion to approximate u at $x \pm h$.

Perhaps, using grid-index notation, the main idea behind the Taylor series expansion reads more clearly:

$$u_{i\pm1} = u_i \pm hu_i' + \frac{h^2}{2}u_i'' \pm \frac{h^3}{6}u_i''' + \frac{h^4}{24}u^{(4)}(x),$$

Taylor expansions

- Let us assume that $u(x)$ is C^4 on $x \in \mathbb{R}$, and:

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h} \tag{1}$$

- In the neighborhood of point x, we can use a Taylor series expansion to approximate $u(x+h)$ as:

$$u(x+h) = u(x) + \frac{h}{1!}u'(x) + \frac{h^2}{2!}u''(x) + \frac{h^3}{3!}u'''(x) + \frac{h^4}{4!}u^{(4)}(x),$$

- Let us assume we are interested in an approximation based only on the first two terms of the previous expression:

$$u(x+h) \approx u(x) + hu'(x) + O(h^2),$$

- The term $O(h^2)$ indicates the error of the approximation is proportional to h^2.
- The approximation is said to be consistent at first order.

NOTES:

- $u(x)$ being C^4 means that the function has four continuous derivatives on $x \in \mathbb{R}$.
- When h is small (but nonzero), the difference quotient on the right-hand side (rhs) is a good approximation of the derivative $u'(x)$.
- If (1) holds, we may assume that u is smooth and its derivatives exist at x; then we can use a Taylor series expansion to approximate u at $x \pm h$.

Perhaps, using grid-index notation, the main idea behind the Taylor series expansion reads more clearly:

$$u_{i\pm 1} = u_i \pm h u_i' + \frac{h^2}{2} u_i'' \pm \frac{h^3}{6} u_i''' + \frac{h^4}{24} u^{(4)}(x),$$

Taylor expansions

- Let us assume that $u(x)$ is C^4 on $x \in \mathbb{R}$, and:

$$u'(x) = \lim_{h \to 0} \frac{u(x+h) - u(x)}{h} \tag{1}$$

- In the neighborhood of point x, we can use a Taylor series expansion to approximate $u(x+h)$ as:

$$u(x+h) = u(x) + \frac{h}{1!}u'(x) + \frac{h^2}{2!}u''(x) + \frac{h^3}{3!}u'''(x) + \frac{h^4}{4!}u^{(4)}(x),$$

- Let us assume we are interested in an approximation based only on the first two terms of the previous expression:

$$u(x+h) \approx u(x) + hu'(x) + O(h^2),$$

- The term $O(h^2)$ indicates the error of the approximation is proportional to h^2.
- The approximation is said to be consistent at first order.

NOTES:

- $u(x)$ being C^4 means that the function has four continuous derivatives on $x \in \mathbb{R}$.
- When h is small (but nonzero), the difference quotient on the right-hand side (rhs) is a good approximation of the derivative $u'(x)$.
- If (1) holds, we may assume that u is smooth and its derivatives exist at x; then we can use a Taylor series expansion to approximate u at $x \pm h$.

Perhaps, using grid-index notation, the main idea behind the Taylor series expansion reads more clearly:

$$u_{i\pm1} = u_i \pm hu_i' + \frac{h^2}{2}u_i'' \pm \frac{h^3}{6}u_i''' + \frac{h^4}{24}u^{(4)}(x),$$

Taylor expansions notation

Taylor expansions around x give either a **forward** or a **backward** approximation of the derivative, namely:

$$u(x+h) = u(x) + h\,u'(x) + \frac{h^2}{2}u''(x) + \frac{h^3}{6}u'''(x) + O(h^4),$$
$$u(x-h) = u(x) - h\,u'(x) + \frac{h^2}{2}u''(x) - \frac{h^3}{6}u'''(x) + O(h^4).$$

For the sake of readability, the following notation is used:

$$u_i = u(x_i),$$
$$u_i' = u'(x_i),$$
$$u_i'' = u''(x_i),$$

Then the forward and backward formulas become:

$$u_{i+1} = u_i + h\,u_i' + \frac{h^2}{2}u_i'' + \frac{h^3}{6}u_i''' + O(h^4),$$
$$u_{i-1} = u_i - h\,u_i' + \frac{h^2}{2}u_i'' - \frac{h^3}{6}u_i''' + O(h^4).$$

Taylor expansions notation

Taylor expansions around x give either a **forward** or a **backward** approximation of the derivative, namely:

$$u(x+h) = u(x) + h\,u'(x) + \tfrac{h^2}{2}u''(x) + \tfrac{h^3}{6}u'''(x) + O(h^4),$$
$$u(x-h) = u(x) - h\,u'(x) + \tfrac{h^2}{2}u''(x) - \tfrac{h^3}{6}u'''(x) + O(h^4).$$

For the sake of readability, the following notation is used:

$$u_i = u(x_i),$$

$$u_i' = u'(x_i),$$

$$u_i'' = u''(x_i),$$

Then the forward and backward formulas become:

$$u_{i+1} = u_i + h\,u_i' + \tfrac{h^2}{2}u_i'' + \tfrac{h^3}{6}u_i''' + O(h^4),$$
$$u_{i-1} = u_i - h\,u_i' + \tfrac{h^2}{2}u_i'' - \tfrac{h^3}{6}u_i''' + O(h^4).$$

Taylor expansions notation

Taylor expansions around x give either a **forward** or a **backward** approximation of the derivative, namely:

$$u(x+h) = u(x) + h\,u'(x) + \tfrac{h^2}{2}u''(x) + \tfrac{h^3}{6}u'''(x) + O(h^4),$$
$$u(x-h) = u(x) - h\,u'(x) + \tfrac{h^2}{2}u''(x) - \tfrac{h^3}{6}u'''(x) + O(h^4).$$

For the sake of readability, the following notation is used:

$$u_i = u(x_i),$$

$$u_i' = u'(x_i),$$
$$u_i'' = u''(x_i),$$

Then the forward and backward formulas become:

$$u_{i+1} = u_i + h\,u_i' + \tfrac{h^2}{2}u_i'' + \tfrac{h^3}{6}u_i''' + O(h^4),$$
$$u_{i-1} = u_i - h\,u_i' + \tfrac{h^2}{2}u_i'' - \tfrac{h^3}{6}u_i''' + O(h^4).$$

Taylor expansions and first-order spatial differences

Let us consider the first-order **forward** and **backward** Taylor expansions around x:

$$u_{i+1} = u_i + h\,u_i' + O(h^2),$$
$$u_{i-1} = u_i - h\,u_i' + O(h^2).$$

From them, the following difference approximations to u_i' and u_i'' can be derived:

- **Forward 1st-derivative:** $|u_i' - (u_{i+1} - u_i)/h| \leq Ch$
- **Backward 1st-derivative:** $|u_i' - (u_i - u_{i-1})/h| \leq Ch$
- **Central 1st-derivative:** $|u_i' - (u_{i+1} - u_{i-1})/2h| \leq Ch^2$
- **Central 2nd-derivative:** $|u_i'' - (u_{i+1} - 2u_i + u_{i-1})/h^2| \leq Ch^2$

Observe that the finite difference approximations share a common algebraic structure:

$$u_i^{(k)} \approx \frac{1}{h^k}(\alpha_{i-1}\,u_{i-1} + \alpha_i\,u_i + \alpha_{i+1}\,u_{i+1}) \equiv \frac{1}{h^k}\sum_{j=i-1}^{i+1} \alpha_j\,u_j,$$

where α_j are the weights and k is the order of the approximation.

Taylor expansions and first-order spatial differences

Let us consider the first-order **forward** and **backward** Taylor expansions around x:

$$u_{i+1} = u_i + h\,u_i' + O(h^2),$$
$$u_{i-1} = u_i - h\,u_i' + O(h^2).$$

From them, the following difference approximations to u_i' and u_i'' can be derived:

- **Forward 1st-derivative:** $u_i' = (u_{i+1} - u_i)/h + O(h^2)$
- **Backward 1st-derivative:** $u_i' = (u_i - u_{i-1})/h + O(h^2)$
- **Central 1st-derivative:** $u_i' = (u_{i+1} - u_{i-1})/2h + O(h^3)$
- **Central 2nd-derivative:** $u_i'' = (u_{i+1} - 2u_i + u_{i-1})/h^2 + O(h^3)$
- **Forward 1st-derivative:** $|u_i' - (u_{i+1} - u_i)/h| \le Ch$
- **Backward 1st-derivative:** $|u_i' - (u_i - u_{i-1})/h| \le Ch$
- **Central 1st-derivative:** $|u_i' - (u_{i+1} - u_{i-1})/2h| \le Ch^2$
- **Central 2nd-derivative:** $|u_i'' - (u_{i+1} - 2u_i + u_{i-1})/h^2| \le Ch^2$

Observe that the finite difference approximations share a common algebraic structure:

Taylor expansions and first-order spatial differences

Let us consider the first-order **forward** and **backward** Taylor expansions around x:

$$u_{i+1} = u_i + h\, u_i' + O(h^2),$$
$$u_{i-1} = u_i - h\, u_i' + O(h^2).$$

From them, the following difference approximations to u_i' and u_i'' can be derived:

- **Forward 1st-derivative:** $|u_i' - (u_{i+1} - u_i)/h| \leq Ch$
- **Backward 1st-derivative:** $|u_i' - (u_i - u_{i-1})/h| \leq Ch$
- **Central 1st-derivative:** $|u_i' - (u_{i+1} - u_{i-1})/2h| \leq Ch^2$
- **Central 2nd-derivative:** $|u_i'' - (u_{i+1} - 2u_i + u_{i-1})/h^2| \leq Ch^2$

Observe that the finite difference approximations share a common algebraic structure:

$$u_i^{(k)} \approx \frac{1}{h^k}(\alpha_{i-1}\, u_{i-1} + \alpha_i\, u_i + \alpha_{i+1}\, u_{i+1}) \equiv \frac{1}{h^k} \sum_{j=i-1}^{i+1} \alpha_j\, u_j,$$

where α_j are the weights and k is the order of the approximation.

Taylor expansions and first-order spatial differences

Let us consider the first-order **forward** and **backward** Taylor expansions around x:

$$u_{i+1} = u_i + h\,u_i' + O(h^2),$$
$$u_{i-1} = u_i - h\,u_i' + O(h^2).$$

From them, the following difference approximations to u_i' and u_i'' can be derived:

- **Forward 1st-derivative:** $|u_i' - (u_{i+1} - u_i)/h| \leq Ch$
- **Backward 1st-derivative:** $|u_i' - (u_i - u_{i-1})/h| \leq Ch$
- **Central 1st-derivative:** $|u_i' - (u_{i+1} - u_{i-1})/2h| \leq Ch^2$
- **Central 2nd-derivative:** $|u_i'' - (u_{i+1} - 2u_i + u_{i-1})/h^2| \leq Ch^2$

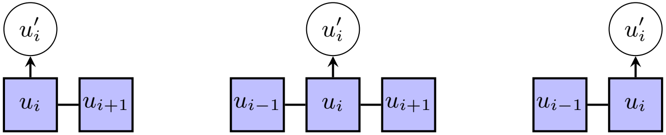
Observe that the finite difference approximations share a common algebraic structure:

$$u_i^{(k)} \approx \frac{1}{h^k}(\alpha_{i-1}\,u_{i-1} + \alpha_i\,u_i + \alpha_{i+1}\,u_{i+1}) \equiv \frac{1}{h^k}\sum_{j=i-1}^{i+1} \alpha_j\,u_j\,,$$

where α_j are the weights and k is the order of the approximation.

Taylor expansions and first-order spatial differences

Schemes of the form $u_i^{(k)} \approx \sum \alpha_j \, u_j$ are called **explicit schemes** (Fornberg 1988).
Following the previous examples we can recognize three patterns, graphically:



forward central backward

Schemes of the form $\sum \beta_i \, u_i^{(k)} \approx \sum \alpha_j \, u_j$ are called **implicit schemes** (Fornberg 1998). If the field values u and u' are collocated they are called **compact schemes**, graphically:

compact Padé-type

Padé-type schemes are commonly built using a Taylor Table technique (Moin 2010).

NOTES:

What is a Padé scheme?

- In CFD and numerical PDEs, a Padé (compact finite-difference) scheme approximates derivatives on a grid where the derivative values themselves appear on both sides of a small linear system (often tridiagonal), instead of each derivative being an explicit combination of neighboring function values only. For example, the classical Padé fourth-order compact scheme reads:

$$\frac{1}{4}\,u'_{i-1} + u'_i + \frac{1}{4}\,u'_{i+1} = \frac{3}{4h}\,\big(u_{i+1} - u_{i-1}\big) + O(h^4)$$

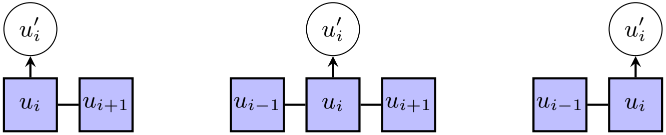
What is the Taylor Table technique?

- It is a technique to derive the coefficients of the difference approximations. It is based on the Taylor series expansion of the function to be approximated.

An example using a Taylor table for finding the coefficients of a classical Padé fourth-order compact scheme is found in Chapter 2.4 of Moin (2010).

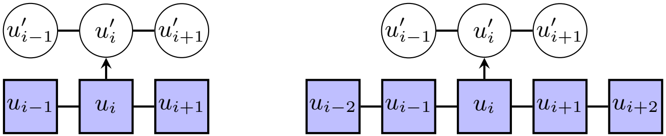
Taylor expansions and first-order spatial differences

Schemes of the form $u_i^{(k)} \approx \sum \alpha_j \, u_j$ are called **explicit schemes** (Fornberg 1988).
Following the previous examples we can recognize three patterns, graphically:



forward central backward

Schemes of the form $\sum \beta_i \, u_i^{(k)} \approx \sum \alpha_j \, u_j$ are called **implicit schemes** (Fornberg 1998). If the field values u and u' are collocated they are called **compact schemes**, graphically:



compact Padé-type

Padé-type schemes are commonly built using a Taylor Table technique (Moin 2010).

NOTES:

What is a Padé scheme?

- In CFD and numerical PDEs, a Padé (compact finite-difference) scheme approximates derivatives on a grid where the derivative values themselves appear on both sides of a small linear system (often tridiagonal), instead of each derivative being an explicit combination of neighboring function values only. For example, the classical Padé fourth-order compact scheme reads:

$$\frac{1}{4}\,u'_{i-1} + u'_i + \frac{1}{4}\,u'_{i+1} = \frac{3}{4h}\,\big(u_{i+1} - u_{i-1}\big) + O(h^4)$$

What is the Taylor Table technique?

- It is a technique to derive the coefficients of the difference approximations. It is based on the Taylor series expansion of the function to be approximated.

An example using a Taylor table for finding the coefficients of a classical Padé fourth-order compact scheme is found in Chapter 2.4 of Moin (2010).

Outline

The one-dimensional problem

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0,$$

Consider $\Omega \in]0, L]$ and $t \in]0, T]$, where $L = 2.0$ and $T = 2.0$, along with suitable initial condition $f(x)$ and boundary condition $g(t)$.

- For simplicity, let us consider:

$$c = 1.0, \qquad f(x) = 0 \quad \text{and} \quad g(t) = \exp\left(-\frac{(t-1.0)^2}{0.05}\right).$$

- We discretize the spatial domain into N_x=7 nodes so that $h = L/(N_x - 1)$ and

$$x_i = ih \quad \text{for} \quad i = 0, 1, \ldots, N_x - 1.$$

- The discrete solution vector is $\mathbf{u} = [u_0, u_1, u_2, u_3, u_4, u_5, u_6]^T$.

- Lastly, let us consider $N_t = 7$ time steps, so that $\delta = T/N_t$ and

$$t^n = n\delta \quad \text{for} \quad n = 1, \ldots, N_t.$$

Using difference approximations

The one-dimensional problem (discrete form)

$$\frac{u^{n+1}{}_i - u^n{}_i}{\delta t} + c\frac{u^n{}_i - u^n{}_{i-1}}{h} = 0, \qquad \text{(FTBS)}$$

Consider $\Omega \in]0, L]$ and $t \in]0, T]$, where $L = 2.0$ and $T = 2.0$, along with suitable initial condition $f(x)$ and boundary condition $g(t)$.

- For simplicity, let us consider:

$$c = 1.0, \qquad f(x) = 0 \quad \text{and} \quad g(t) = \exp\left(-\frac{(t-1.0)^2}{0.05}\right).$$

- We discretize the spatial domain into N_x=7 nodes so that $h = L/(N_x - 1)$ and

$$x_i = ih \quad \text{for} \quad i = 0, 1, \ldots, N_x - 1.$$

- The discrete solution vector is $\mathbf{u} = [u_0, u_1, u_2, u_3, u_4, u_5, u_6]^T$.
- Lastly, let us consider $N_t = 7$ time steps, so that $\delta = T/N_t$ and

$$t^n = n\delta \quad \text{for} \quad n = 1, \ldots, N_t.$$

Using difference approximations

The one-dimensional problem (discrete form)

$$\frac{u^{n+1}{}_i - u^n{}_i}{\delta t} + c\frac{u^n{}_i - u^n{}_{i-1}}{h} = 0, \qquad \text{(FTBS)}$$

Consider $\Omega \in]0, L]$ and $t \in]0, T]$, where $L = 2.0$ and $T = 2.0$, along with suitable initial condition $f(x)$ and boundary condition $g(t)$.

- For simplicity, let us consider:

$$c = 1.0, \qquad f(x) = 0 \quad \text{and} \quad g(t) = \exp\left(-\frac{(t-1.0)^2}{0.05}\right).$$

- We discretize the spatial domain into N_x=7 nodes so that $h = L/(N_x - 1)$ and

$$x_i = ih \quad \text{for} \quad i = 0, 1, \ldots, N_x - 1.$$

- The discrete solution vector is $\mathbf{u} = [u_0, u_1, u_2, u_3, u_4, u_5, u_6]^T$.
- Lastly, let us consider $N_t = 7$ time steps, so that $\delta = T/N_t$ and

$$t^n = n\delta \quad \text{for} \quad n = 1, \ldots, N_t.$$

Using difference approximations

The one-dimensional problem (discrete form)

$$\frac{u^{n+1}{}_i - u^n{}_i}{\delta t} + c\frac{u^n{}_i - u^n{}_{i-1}}{h} = 0, \qquad \text{(FTBS)}$$

Consider $\Omega \in]0, L]$ and $t \in]0, T]$, where $L = 2.0$ and $T = 2.0$, along with suitable initial condition $f(x)$ and boundary condition $g(t)$.

- For simplicity, let us consider:

$$c = 1.0, \qquad f(x) = 0 \quad \text{and} \quad g(t) = \exp\left(-\frac{(t-1.0)^2}{0.05}\right).$$

- We discretize the spatial domain into N_x=7 nodes so that $h = L/(N_x - 1)$ and

$$x_i = ih \quad \text{for} \quad i = 0, 1, \ldots, N_x - 1.$$

- The discrete solution vector is $\mathbf{u} = [u_0, u_1, u_2, u_3, u_4, u_5, u_6]^T$.
- Lastly, let us consider $N_t = 7$ time steps, so that $\delta = T/N_t$ and

$$t^n = n\delta \quad \text{for} \quad n = 1, \ldots, N_t.$$

Using difference approximations

The one-dimensional problem (discrete form)

$$\frac{u^{n+1}{}_{i} - u^{n}{}_{i}}{\delta t} + c\frac{u^{n}{}_{i} - u^{n}{}_{i-1}}{h} = 0, \qquad \text{(FTBS)}$$

Consider $\Omega \in]0, L]$ and $t \in]0, T]$, where $L = 2.0$ and $T = 2.0$, along with suitable initial condition $f(x)$ and boundary condition $g(t)$.

- For simplicity, let us consider:

$$c = 1.0, \qquad f(x) = 0 \quad \text{and} \quad g(t) = \exp\left(-\frac{(t-1.0)^2}{0.05}\right).$$

- We discretize the spatial domain into N_x=7 nodes so that $h = L/(N_x - 1)$ and

$$x_i = ih \quad \text{for} \quad i = 0, 1, \ldots, N_x - 1.$$

- The discrete solution vector is $\mathbf{u} = [u_0, u_1, u_2, u_3, u_4, u_5, u_6]^T$.

- Lastly, let us consider $N_t = 7$ time steps, so that $\delta = T/N_t$ and

$$t^n = n\delta \quad \text{for} \quad n = 1, \ldots, N_t.$$

Using difference approximations

The one-dimensional problem (discrete form)

$$\frac{u^{n+1}{}_i - u^n{}_i}{\delta t} + c\frac{u^n{}_i - u^n{}_{i-1}}{h} = 0, \qquad \text{(FTBS)}$$

Consider $\Omega \in]0, L]$ and $t \in]0, T]$, where $L = 2.0$ and $T = 2.0$, along with suitable initial condition $f(x)$ and boundary condition $g(t)$.

- For simplicity, let us consider:

$$c = 1.0, \qquad f(x) = 0 \quad \text{and} \quad g(t) = \exp\left(-\frac{(t-1.0)^2}{0.05}\right).$$

- We discretize the spatial domain into N_x=7 nodes so that $h = L/(N_x - 1)$ and

$$x_i = ih \quad \text{for} \quad i = 0, 1, \ldots, N_x - 1.$$

- The discrete solution vector is $\mathbf{u} = [u_0, u_1, u_2, u_3, u_4, u_5, u_6]^T$.

- Lastly, let us consider $N_t = 7$ time steps, so that $\delta = T/N_t$ and

$$t^n = n\delta \quad \text{for} \quad n = 1, \ldots, N_t.$$

Implementing the scheme explicitly

The 1-d problem can be written as:

$$u_i^{n+1} = u_i^n - c\frac{\delta t}{h}(u_i^n - u_{i-1}^n)$$

Explicitly:

$$\begin{aligned}
u_0^{n+1} =& g(t^{n+1}), \\
u_1^{n+1} =& u_1^n - c\frac{\delta t}{h}(u_1^n - u_0^n), \\
u_2^{n+1} =& u_2^n - c\frac{\delta t}{h}(u_2^n - u_1^n), \\
u_3^{n+1} =& u_3^n - c\frac{\delta t}{h}(u_3^n - u_2^n), \\
u_4^{n+1} =& u_4^n - c\frac{\delta t}{h}(u_4^n - u_3^n), \\
u_5^{n+1} =& u_5^n - c\frac{\delta t}{h}(u_5^n - u_4^n), \\
u_6^{n+1} =& u_6^n - c\frac{\delta t}{h}(u_6^n - u_5^n),
\end{aligned}$$

Implementing the scheme explicitly

The 1-d problem can be written as:

$$u_i^{n+1} = u_i^n - c\frac{\delta t}{h}(u_i^n - u_{i-1}^n)$$

Explicitly:

$$\begin{aligned}
u_0^{n+1} =& g(t^{n+1}),\\
u_1^{n+1} =& u_1^n - c\frac{\delta t}{h}(u_1^n - u_0^n),\\
u_2^{n+1} =& u_2^n - c\frac{\delta t}{h}(u_2^n - u_1^n),\\
u_3^{n+1} =& u_3^n - c\frac{\delta t}{h}(u_3^n - u_2^n),\\
u_4^{n+1} =& u_4^n - c\frac{\delta t}{h}(u_4^n - u_3^n),\\
u_5^{n+1} =& u_5^n - c\frac{\delta t}{h}(u_5^n - u_4^n),\\
u_6^{n+1} =& u_6^n - c\frac{\delta t}{h}(u_6^n - u_5^n),
\end{aligned}$$

A minimal working implementation reads as follows:

```python
import numpy as np
# parameters
c, L, T_final = 1.0, 2.0, 2.0
N_x, N_t = 7, 7
h, dt = L / (N_x - 1), T_final / N_t
# boundary condition function
g = lambda t: np.exp(-(t - 1.0)**2 / 0.05)
# initial condition and boundary condition
u = np.zeros(N_x); u[0] = g(0)
# scheme
un = np.zeros_like(u)
for n in range(1, N_t+1):
  # Update boundary condition
  t = n * dt; un[0] = g(t)
  for i in range(1, N_x):
    un[i] = u[i] - c * dt / h * (u[i] - u[i-1])
  # Update solution
  u = un.copy()
```

Implementing the scheme explicitly (Matrix-based approach)

The 1-d problem can be written as:

$$u_i^{n+1} = u_i^n - c\frac{\delta t}{h}(u_i^n - u_{i-1}^n)$$

Explicitly:

$$\begin{aligned}
u_0^{n+1} =& g(t^{n+1}),\\
u_1^{n+1} =& u_1^n - c\frac{\delta t}{h}(u_1^n - u_0^n),\\
u_2^{n+1} =& u_2^n - c\frac{\delta t}{h}(u_2^n - u_1^n),\\
u_3^{n+1} =& u_3^n - c\frac{\delta t}{h}(u_3^n - u_2^n),\\
u_4^{n+1} =& u_4^n - c\frac{\delta t}{h}(u_4^n - u_3^n),\\
u_5^{n+1} =& u_5^n - c\frac{\delta t}{h}(u_5^n - u_4^n),\\
u_6^{n+1} =& u_6^n - c\frac{\delta t}{h}(u_6^n - u_5^n),
\end{aligned}$$

Notice the recursive nature of the equations. We can write them in a more compact form as follows:

$$\mathbf{u}^{n+1} = \mathbf{u}^n - c\,\delta t \boldsymbol{A}\mathbf{u}^n,$$

where $\boldsymbol{A}$ is a matrix with the following structure:

$$\boldsymbol{A} = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0\\
-\frac{1}{h} & \frac{1}{h} & 0 & 0 & 0 & 0 & 0\\
0 & -\frac{1}{h} & \frac{1}{h} & 0 & 0 & 0 & 0\\
0 & 0 & -\frac{1}{h} & \frac{1}{h} & 0 & 0 & 0\\
0 & 0 & 0 & -\frac{1}{h} & \frac{1}{h} & 0 & 0\\
0 & 0 & 0 & 0 & -\frac{1}{h} & \frac{1}{h} & 0\\
0 & 0 & 0 & 0 & 0 & -\frac{1}{h} & \frac{1}{h}
\end{bmatrix},$$

where the first row is the boundary condition and the rest are the interior points.

Outline

How do we know it is right?

Good computer scientists usually check 4 things (well, at least 3):

- The consistency of the scheme
  We use **Taylor expansions** of the operators to check if the scheme is consistent with the original PDE. (For FD approximations, this is straightforward.)
- The stability of the scheme
  The **Von Neumann stability analysis** is used to examine whether the growth is bounded or unbounded.
- The convergence of the scheme
  For well-posed PDEs, using linear numerical methods, we can rely on **Lax-Richtmyer Equivalence** theorem:
  $$\boxed{\text{consistency} + \text{stability} \implies \text{convergence}}$$
- The accuracy of the scheme
  If the scheme converges, we can compute the **order of accuracy (OoA)** by comparing the numerical solution with the exact solution.

How do we know it is right?

Good computer scientists usually check 4 things (well, at least 3):

- The consistency of the scheme
  We use **Taylor expansions** of the operators to check if the scheme is consistent with the original PDE. (For FD approximations, this is straightforward.)
- The stability of the scheme
  The **Von Neumann stability analysis** is used to examine whether the growth is bounded or unbounded.
- The convergence of the scheme
  For well-posed PDEs, using linear numerical methods, we can rely on **Lax-Richtmyer Equivalence** theorem:

$$\boxed{\text{consistency} + \text{stability} \implies \text{convergence}}$$

- The accuracy of the scheme
  If the scheme converges, we can compute the **order of accuracy (OoA)** by comparing the numerical solution with the exact solution.

How do we know it is right?

Good computer scientists usually check 4 things (well, at least 3):

- The consistency of the scheme
  We use **Taylor expansions** of the operators to check if the scheme is consistent with the original PDE. (For FD approximations, this is straightforward.)
- The stability of the scheme
  The **Von Neumann stability analysis** is used to examine whether the growth is bounded or unbounded.
- The convergence of the scheme
  For well-posed PDEs, using linear numerical methods, we can rely on **Lax-Richtmyer Equivalence** theorem:
  $$\boxed{\text{consistency} + \text{stability} \Longrightarrow \text{convergence}}$$
- The accuracy of the scheme
  If the scheme converges, we can compute the **order of accuracy (OoA)** by comparing the numerical solution with the exact solution.

How do we know it is right?

Good computer scientists usually check 4 things (well, at least 3):

- The consistency of the scheme
  We use **Taylor expansions** of the operators to check if the scheme is consistent with the original PDE. (For FD approximations, this is straightforward.)

- The stability of the scheme
  The **Von Neumann stability analysis** is used to examine whether the growth is bounded or unbounded.

- The convergence of the scheme
  For well-posed PDEs, using linear numerical methods, we can rely on **Lax-Richtmyer Equivalence** theorem:

  $$\boxed{\text{consistency} + \text{stability} \Longrightarrow \text{convergence}}$$

- The accuracy of the scheme
  If the scheme converges, we can compute the **order of accuracy (OoA)** by comparing the numerical solution with the exact solution.

Consistency

Consider the operator $\mathcal{L} = \partial/\partial t + \alpha \partial/\partial x$ with $\alpha > 0$ applied to a field ϕ so that:

Exact operator:

$$\mathcal{L}\phi = \phi_t + \alpha\phi_x.$$

Discrete operator:

$$\mathcal{L}_{\delta t, \delta x}\phi = \frac{{\phi^{n+1}}_i - {\phi^n}_i}{\delta t} + \alpha \frac{{\phi^n}_{i+1} - {\phi^n}_i}{\delta x}$$

(forward-time forward-space scheme)

where ${\phi^n}_i = \phi(n\delta t, i\delta x)$.

We begin by taking the Taylor expansion of ϕ in t and x around the point (t^n, x_i):

$$ {\phi^{n+1}}_i = {\phi^n}_i + \delta t {(\phi_t)^n}_i + \frac{1}{2}\delta t^2 {(\phi_{tt})^n}_i + O(\delta t^3), \qquad \phi^n_{i+1} = \phi^n_i + \delta x (\phi_x)^n_i + \frac{1}{2}\delta x^2 (\phi_{xx})^n_i + O(\delta x^3).$$

Substituting into the discrete operator and rearranging, we get:

$$\mathcal{L}_{\delta t, \delta x}\phi = \phi_t + \alpha\phi_x + \frac{1}{2}\delta t \phi_{tt} + \alpha\frac{1}{2}\delta x \phi_{xx} + O(\delta t^3) + O(\delta x^3).$$

Thus,

$$\mathcal{L}_{\delta t, \delta x}\phi - \mathcal{L}\phi = \frac{1}{2}\delta t \phi_{tt} + \alpha\frac{1}{2}\delta x \phi_{xx} + O(\delta t^3) + O(\delta x^3) \to 0 \quad \text{as} \quad \delta t, \delta x \to 0.$$

Thus this scheme is consistent $\square$.

Consistency

Consider the operator $\mathcal{L} = \partial/\partial t + \alpha \partial/\partial x$ with $\alpha > 0$ applied to a field ϕ so that:

Exact operator:

$$\mathcal{L}\phi = \phi_t + \alpha\phi_x.$$

Discrete operator:

$$\mathcal{L}_{\delta t,\delta x}\phi = \frac{\phi^{n+1}{}_i - \phi^n{}_i}{\delta t} + \alpha\frac{\phi^n{}_{i+1} - \phi^n{}_i}{\delta x}$$

(forward-time forward-space scheme)

where $\phi^n{}_i = \phi(n\delta t, i\delta x)$.

We begin by taking the Taylor expansion of ϕ in t and x around the point (t^n, x_i):

$$\phi^{n+1}{}_i = \phi^n{}_i + \delta t(\phi_t)^n{}_i + \tfrac{1}{2}\delta t^2(\phi_{tt})^n{}_i + O(\delta t^3), \qquad \phi^n_{i+1} = \phi^n_i + \delta x(\phi_x)^n_i + \tfrac{1}{2}\delta x^2(\phi_{xx})^n_i + O(\delta x^3).$$

Substituting into the discrete operator and rearranging, we get:

$$\mathcal{L}_{\delta t,\delta x}\phi = \phi_t + \alpha\phi_x + \tfrac{1}{2}\delta t\phi_{tt} + \alpha\tfrac{1}{2}\delta x\phi_{xx} + O(\delta t^3) + O(\delta x^3).$$

Thus,

$$\mathcal{L}_{\delta t,\delta x}\phi - \mathcal{L}\phi = \tfrac{1}{2}\delta t\phi_{tt} + \alpha\tfrac{1}{2}\delta x\phi_{xx} + O(\delta t^3) + O(\delta x^3) \to 0 \quad \text{as} \quad \delta t, \delta x \to 0.$$

Thus this scheme is consistent $\square$.

Consistency

Consider the operator $\mathcal{L} = \partial/\partial t + \alpha\partial/\partial x$ with $\alpha > 0$ applied to a field ϕ so that:

Exact operator:

$$\mathcal{L}\phi = \phi_t + \alpha\phi_x.$$

Discrete operator:

$$\mathcal{L}_{\delta t,\delta x}\phi = \frac{\phi^{n+1}{}_i - \phi^n{}_i}{\delta t} + \alpha\frac{\phi^n{}_{i+1} - \phi^n{}_i}{\delta x}$$

(forward-time forward-space scheme)

where $\phi^n{}_i = \phi(n\delta t, i\delta x)$.

We begin by taking the Taylor expansion of ϕ in t and x around the point (t^n, x_i):

$$\phi^{n+1}{}_i = \phi^n{}_i + \delta t(\phi_t)^n{}_i + \tfrac{1}{2}\delta t^2(\phi_{tt})^n{}_i + O(\delta t^3), \qquad \phi^n_{i+1} = \phi^n_i + \delta x(\phi_x)^n_i + \tfrac{1}{2}\delta x^2(\phi_{xx})^n_i + O(\delta x^3).$$

Substituting into the discrete operator and rearranging, we get:

$$\mathcal{L}_{\delta t,\delta x}\phi = \phi_t + \alpha\phi_x + \tfrac{1}{2}\delta t\phi_{tt} + \alpha\tfrac{1}{2}\delta x\phi_{xx} + O(\delta t^3) + O(\delta x^3).$$

Thus,

$$\mathcal{L}_{\delta t,\delta x}\phi - \mathcal{L}\phi = \tfrac{1}{2}\delta t\phi_{tt} + \alpha\tfrac{1}{2}\delta x\phi_{xx} + O(\delta t^3) + O(\delta x^3) \to 0 \quad \text{as} \quad \delta t, \delta x \to 0.$$

Thus this scheme is consistent $\square$.

Von Neumann Stability Analysis

The Von Neumann Stability Analysis assumes that our solution field ϕ can be expressed as ${\phi_j}^n = G^n \exp(ikx_j)$, where G^n is an amplitude and k is the wave number. Consider again:

$$\mathcal{L}_{\delta t,\delta x}\phi : \frac{{\phi^{n+1}}_i - {\phi^n}_i}{\delta t} + \alpha \frac{{\phi^n}_{i+1} - {\phi^n}_i}{\delta x} = 0, \tag{FTFS}$$

Substituting into our FTFS scheme, we get:

$$\frac{G^{n+1}\exp(ikx_i) - G^n\exp(ikx_i)}{\delta t} + \alpha \frac{G^n\exp(ikx_{i+1}) - G^n\exp(ikx_i)}{\delta x} = 0,$$

$$\frac{G^n\exp(ikx_i)(G-1)}{\delta t} + \alpha \frac{G^n\exp(ikx_i)(\exp(ik\delta x)-1)}{\delta x} = 0,$$

$$G - 1 = -\alpha\frac{\delta t}{\delta x}(\exp(ik\delta x) - 1),$$

$$G = 1 - \nu(\exp(i\theta) - 1),$$

Decomposing the modulus,

$$G = (1 + \nu - \nu\cos(\theta)) + i(\nu\sin(\theta)),$$

$$|G|^2 = 1 + 2\nu(1+\nu)(1 - \cos(\theta)),$$

Since $(1 - \cos(\theta)) \geq 0$ and $\nu > 0$, this means that $|G| \geq 1$, meaning the scheme is unstable. In general, for a scheme to be stable, we need to have $|G| \leq 1$.

NOTES:

Long story short: for

$$u_t + cu_x = 0$$

- for $c > 0$:
  - FTFT is unstable,
  - FTCS is unstable,
  - FTBS is stable if $\nu \leq 1$ (CFL condition)
- for $c < 0$:
  - FTFT is stable if $\nu \leq 1$ (CFL condition)
  - FTCS is unstable,
  - FTBS is unstable.

Order of Accuracy (OoA)

The exact solution is given by:

$$u_{exact}(x,t) = \exp\left(-\frac{(x-c\,t+1.0)^2}{0.05}\right),$$

The global error is quantified as:

$$l_1\text{-norm} = \sum_{i=0}^{N_x} |u_i - u_i^{exact}|,$$

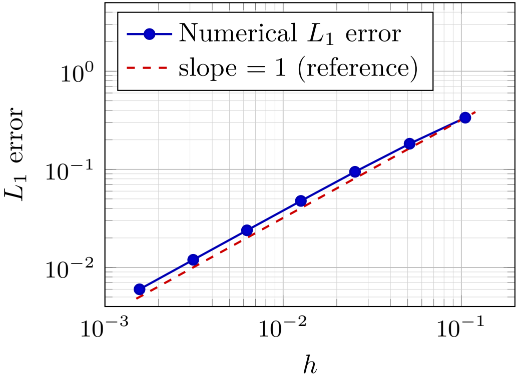
We collect the norm error for several grid sizes, as illustrated in the table below:

N	h	L1 Error	Order
20	1.053e-01	3.364e-01	-
40	5.128e-02	1.825e-01	0.85
80	2.532e-02	9.440e-02	0.93
160	1.258e-02	4.767e-02	0.98
320	6.270e-03	2.391e-02	0.99
640	3.130e-03	1.197e-02	1.00
1280	1.564e-03	5.990e-03	1.00

The *order of accuracy* (OoA) is computed as:

$$Order = \frac{\log(l_1 norm(h_1)) - \log(l_1 norm(h_2))}{\log(h_1) - \log(h_2)},$$

Graphically, on a log-log plot we observe:



Outline

Main Takeaways

1. On Finite Difference Methods (FDM):
   - FDM *is a simple and yet powerful tool* for solving ODEs and PDEs.
   - Although it is a legacy approach, it is still an *active field* of research and development (Fornberg 2021; Fornberg 2025).
   - Currently, explicit schemes are straightforward to build and compute. Nevertheless, boundary conditions are a crucial part of the numerical solution. They are typically handled by special treatment of the first and last grid points.
   - Explicit schemes can be implemented either in a *matrix-free* or *matrix-based* approach.
2. On our one-dimensional problem:
   - The model problem is a linear time-dependent equation: *the advection equation* (Randall J LeVeque 1998).
   - A time-dependent boundary condition is used to exemplify the treatment of boundary conditions in FDM.
   - The size of the time step, δt, is crucial for the stability of the numerical solution. This is a well-known concept called the CFL condition (Courant, Friedrichs, and Lewy 1928).

Main Takeaways

1. On Finite Difference Methods (FDM):
   - FDM *is a simple and yet powerful tool* for solving ODEs and PDEs.
   - Although it is a legacy approach, it is still an *active field* of research and development (Fornberg 2021; Fornberg 2025).
   - Currently, explicit schemes are straightforward to build and compute. Nevertheless, boundary conditions are a crucial part of the numerical solution. They are typically handled by special treatment of the first and last grid points.
   - Explicit schemes can be implemented either in a *matrix-free* or *matrix-based* approach.
2. On our one-dimensional problem:
   - The model problem is a linear time-dependent equation: *the advection equation* (Randall J LeVeque 1998).
   - A time-dependent boundary condition is used to exemplify the treatment of boundary conditions in FDM.
   - The size of the time step, δt, is crucial for the stability of the numerical solution. This is a well-known concept called the CFL condition (Courant, Friedrichs, and Lewy 1928).

Main Takeaways

1. On Finite Difference Methods (FDM):
    - FDM *is a simple and yet powerful tool* for solving ODEs and PDEs.
    - Although it is a legacy approach, it is still an *active field* of research and development (Fornberg 2021; Fornberg 2025).
    - Currently, explicit schemes are straightforward to build and compute. Nevertheless, boundary conditions are a crucial part of the numerical solution. They are typically handled by special treatment of the first and last grid points.
    - Explicit schemes can be implemented either in a *matrix-free* or *matrix-based* approach.
2. On our one-dimensional problem:
    - The model problem is a linear time-dependent equation: *the advection equation* (Randall J LeVeque 1998).
    - A time-dependent boundary condition is used to exemplify the treatment of boundary conditions in FDM.
    - The size of the time step, δt, is crucial for the stability of the numerical solution. This is a well-known concept called the CFL condition (Courant, Friedrichs, and Lewy 1928).

Main Takeaways

1. On Finite Difference Methods (FDM):
   - FDM *is a simple and yet powerful tool* for solving ODEs and PDEs.
   - Although it is a legacy approach, it is still an *active field* of research and development (Fornberg 2021; Fornberg 2025).
   - Currently, explicit schemes are straightforward to build and compute. Nevertheless, boundary conditions are a crucial part of the numerical solution. They are typically handled by special treatment of the first and last grid points.
   - Explicit schemes can be implemented either in a *matrix-free* or *matrix-based* approach.
2. On our one-dimensional problem:
   - The model problem is a linear time-dependent equation: *the advection equation* (Randall J LeVeque 1998).
   - A time-dependent boundary condition is used to exemplify the treatment of boundary conditions in FDM.
   - The size of the time step, δt, is crucial for the stability of the numerical solution. This is a well-known concept called the CFL condition (Courant, Friedrichs, and Lewy 1928).

Main Takeaways

1. On Finite Difference Methods (FDM):
   - FDM *is a simple and yet powerful tool* for solving ODEs and PDEs.
   - Although it is a legacy approach, it is still an *active field* of research and development (Fornberg 2021; Fornberg 2025).
   - Currently, explicit schemes are straightforward to build and compute. Nevertheless, boundary conditions are a crucial part of the numerical solution. They are typically handled by special treatment of the first and last grid points.
   - Explicit schemes can be implemented either in a *matrix-free* or *matrix-based* approach.
2. On our one-dimensional problem:
   - The model problem is a linear time-dependent equation: *the advection equation* (Randall J LeVeque 1998).
   - A time-dependent boundary condition is used to exemplify the treatment of boundary conditions in FDM.
   - The size of the time step, δt, is crucial for the stability of the numerical solution. This is a well-known concept called the CFL condition (Courant, Friedrichs, and Lewy 1928).

Main Takeaways

1. On Finite Difference Methods (FDM):
   - FDM *is a simple and yet powerful tool* for solving ODEs and PDEs.
   - Although it is a legacy approach, it is still an *active field* of research and development (Fornberg 2021; Fornberg 2025).
   - Currently, explicit schemes are straightforward to build and compute. Nevertheless, boundary conditions are a crucial part of the numerical solution. They are typically handled by special treatment of the first and last grid points.
   - Explicit schemes can be implemented either in a *matrix-free* or *matrix-based* approach.

2. On our one-dimensional problem:
   - The model problem is a linear time-dependent equation: *the advection equation* (Randall J LeVeque 1998).
   - A time-dependent boundary condition is used to exemplify the treatment of boundary conditions in FDM.
   - The size of the time step, δt, is crucial for the stability of the numerical solution. This is a well-known concept called the CFL condition (Courant, Friedrichs, and Lewy 1928).

Main Takeaways

1. On Finite Difference Methods (FDM):
   - FDM *is a simple and yet powerful tool* for solving ODEs and PDEs.
   - Although it is a legacy approach, it is still an *active field* of research and development (Fornberg 2021; Fornberg 2025).
   - Currently, explicit schemes are straightforward to build and compute. Nevertheless, boundary conditions are a crucial part of the numerical solution. They are typically handled by special treatment of the first and last grid points.
   - Explicit schemes can be implemented either in a *matrix-free* or *matrix-based* approach.
2. On our one-dimensional problem:
   - The model problem is a linear time-dependent equation: *the advection equation* (Randall J LeVeque 1998).
   - A time-dependent boundary condition is used to exemplify the treatment of boundary conditions in FDM.
   - The size of the time step, δt, is crucial for the stability of the numerical solution. This is a well-known concept called the CFL condition (Courant, Friedrichs, and Lewy 1928).

References I

- Courant, R., K. Friedrichs, and H. Lewy (1928). "Über die partiellen Differenzengleichungen der mathematischen Physik". In: *Mathematische Annalen* 100.1, pp. 32–74.
- Fornberg, Bengt (1988). "Generation of finite difference formulas on arbitrarily spaced grids". In: *Mathematics of computation* 51.184, pp. 699–706.
- — (1998). "Classroom note: Calculation of weights in finite difference formulas". In: *SIAM review* 40.3, pp. 685–691.
- — (2021). "An algorithm for calculating Hermite-based finite difference weights". In: *IMA Journal of Numerical Analysis* 41.2, pp. 801–813.
- — (2025). *High-Accuracy Finite Difference Methods*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, pp. i–iv.
- LeVeque, Randall J (1998). "Finite difference methods for differential equations". In: *Draft version for use in AMath* 585.6, p. 112.

References II

- LeVeque, Randall J. (2002). *Finite Volume Methods for Hyperbolic Problems.* Cambridge: Cambridge University Press.
- Moin, Parviz (2010). *Fundamentals of engineering numerical analysis.* Cambridge University Press.
- Morton, K. W. and D. F. Mayers (2005). *Numerical Solution of Partial Differential Equations.* 3rd ed. Cambridge University Press.