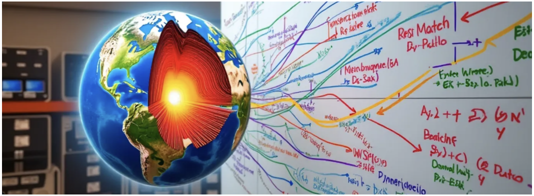
Principles of Numerical Modelling in Geosciences

Lecture 4: Linear Advection and Numerical Schemes

Mattia de' Michieli Vitturi

PhD Course in Earth Sciences



Outline

Lesson 4 Goals: Modeling Advective Transport

Today, we shift our focus to another fundamental transport process: advection. We aim to:

- Understand the physical meaning of advection and its importance in Earth systems.
- Derive the linear advection equation, a key hyperbolic PDE.
- Explore how to set appropriate boundary conditions, introducing periodic boundaries.
- Develop and compare finite difference schemes for solving the advection equation.
- Analyze the crucial concepts of numerical stability and the CFL condition.
- Implement these schemes in Python to simulate advective transport.

What is Advection? The "Carrying Along" Process

Advection is the transport of a scalar quantity (like temperature, chemical concentration, or even momentum) due to the bulk motion of a fluid.

- Imagine volcanic ash being carried hundreds of kilometers by prevailing winds, or dissolved salts being transported downstream by a river.
- The ash or salt doesn't actively move on its own through the air/water; it's *passively carried along* by the flow.

- This is distinct from diffusion, where transport occurs due to random molecular motion or small-scale turbulence, typically from high to low concentration regions.

Advection is a primary mechanism for distributing mass, energy, and chemical constituents throughout many Earth systems.

Advection in Geosciences: Shaping Our Planet

The advective transport of materials and properties is central to countless geological and environmental processes:

- Atmospheric Transport:
  - Dispersion of volcanic ash and gases (SO_2, CO_2) affecting climate and air quality.
  - Movement of air masses, moisture, and pollutants.
- Hydrological Transport:
  - River transport of sediments, nutrients, and contaminants.
- Oceanographic Transport:
  - Ocean currents advecting heat (e.g., Gulf Stream), salinity, and dissolved substances.
- Magmatic and Mantle Processes:
  - Advection of crystals or chemical heterogeneities within flowing magma in conduits or chambers.
  - Large-scale advection of heat and material by mantle convection.

Understanding these requires modeling the advection component of the system.

Building on Previous Concepts

We will uses tools and ideas from earlier lectures:

- Conservation Laws (L3): The equations describing advection are derived from fundamental conservation principles.
- General Scalar Transport Equation (L3): We saw this equation describes changes due to accumulation, convection (advection), diffusion, and sources:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho \mathbf{u} \phi) = \nabla \cdot (\Gamma \nabla \phi) + S$$

  Today, we'll isolate the advection term $\nabla \cdot (\rho \mathbf{u} \phi)$.
- PDEs and Discretization (L2, L3): We've learned that PDEs describe quantities varying in space and time, and we've started discretizing derivatives (e.g., for the heat equation).
- Boundary Conditions (L3): We'll see how Dirichlet, Neumann, and new *periodic* boundary conditions are applied to advection problems.

Isolating Advection: The Linear Advection Equation

To understand the core advective process, we simplify the General Scalar Transport Equation by making several assumptions. Let consider the transport of a scalar quantity ϕ.

Simplifying Assumptions for 1D Linear Advection:

1. Constant Density ρ: The carrying fluid is incompressible.
2. No Diffusion: The spreading of ϕ by molecular or turbulent diffusion is negligible ($\Gamma = 0$).
3. No Sources or Sinks: The quantity ϕ is not created or destroyed within the domain ($S = 0$).
4. One-Dimensional, Constant Velocity Flow: The fluid moves in one direction (say, x) with a constant velocity u. So, $\mathbf{u} = (u, 0, 0)$ with $u =$ constant.

Under these conditions, how does the general equation simplify?

Derivation of the 1D Linear Advection Equation

Start with the general transport equation, assuming constant density ρ:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = \nabla \cdot (\frac{\Gamma}{\rho}\nabla\phi) + \frac{S}{\rho}$$

Apply our simplifying assumptions:

1. No diffusion ($\Gamma = 0$) and no sources/sinks ($S = 0$):

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}\phi) = 0$$

2. For 1D flow $\mathbf{u} = (u, 0, 0)$ with $u =$ constant: The divergence term becomes $\nabla \cdot (\mathbf{u}\phi) = \frac{\partial(u\phi)}{\partial x}$. Since u is constant, $\frac{\partial(u\phi)}{\partial x} = u\frac{\partial \phi}{\partial x}$.

This gives us the 1D Linear Advection Equation:

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

Derivation of the 1D Linear Advection Equation

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

Key Properties:

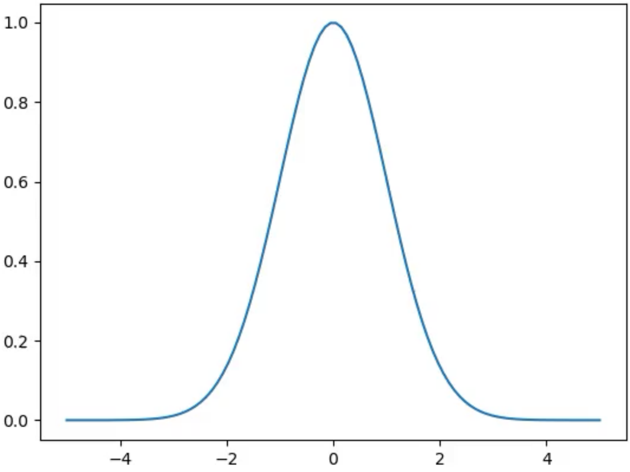
- First-order in both time and space.
- Linear (if u is constant or only a function of x, t, not ϕ).
- Hyperbolic PDE: This classification relates to how information (the value of ϕ) propagates.

Understanding Hyperbolic PDEs: Propagation of Information

The "hyperbolic" classification has a profound physical meaning for how solutions behave. Information (the value of ϕ, or disturbances in ϕ) travels through the domain at a finite speed, determined by the velocity u. This propagation is directional. In parabolic (diffusive) systems, a disturbance at any point in space is, in principle, felt instantly (though weakly) everywhere else in the domain.

First order linear equation

Advection



First order linear equation

Advection equation

The 1D advection equation requires an initial condition of the form

$$\phi(x, t = 0) = \phi_0(x),$$

where $\phi_0(x)$ is given.

The solution is trivial: any initial configuration simply shifts to the right (for $u > 0$) or to the left (for $u < 0$):

- $\phi(x + ut, t) = \phi_0(x)$ is a solution

or, equivlently:

- $\phi(x, t) = \phi_0(x - ut)$ is a solution

This demonstrates that the solution is constant on lines $x - ut = c$. These are called the characteristic curves.

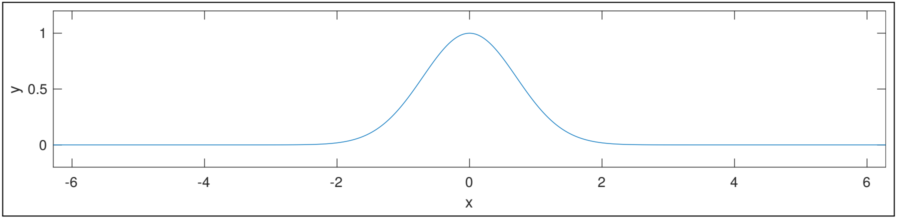
First order linear equation

Advection equation

With this equation:

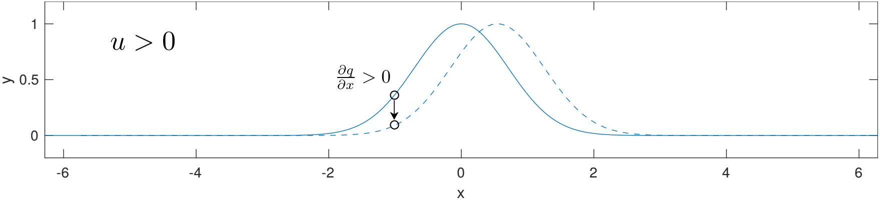
$$\frac{\partial \phi}{\partial t} = -u \frac{\partial \phi}{\partial x}$$

it is easy to understand how variations in space and time are related. Let consider an initial solution given by the function $\phi_0(x) = e^{-x^2}$. We want to see how the solution changes in time accordingly with the terms on the right-hand side of the previous equation.



First order linear equation

Advection equation. Case 1: $\frac{\partial \phi}{\partial x} > 0$ and $u > 0$



In $x = -1$ we have $\frac{\partial \phi}{\partial x} > 0$. Thus, if $u > 0$:

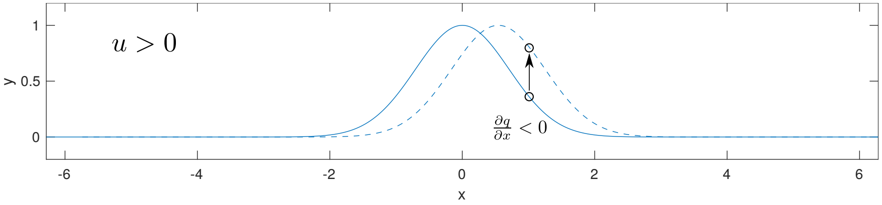
$$\frac{\partial \phi}{\partial t} = -u \frac{\partial \phi}{\partial x} < 0$$

and the value of ϕ in $x = -1$ will decrease with time.

The derivative expresses a local property of the solution, so the we can only say the the value of ϕ in $x = -1$ will decrease with time "for a short time".

First order linear equation

Advection equation. Case 2: $\frac{\partial \phi}{\partial x} < 0$ and $u > 0$



In $x = 1$ we have $\frac{\partial \phi}{\partial x} < 0$. Thus, if $u > 0$:

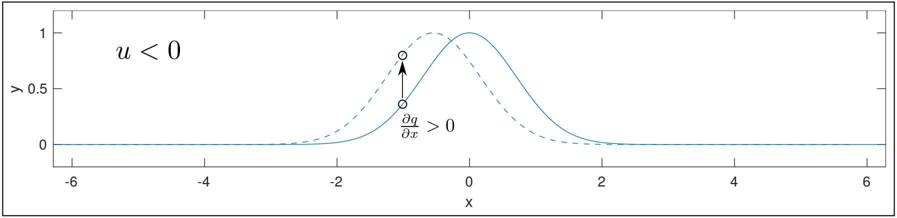
$$\frac{\partial \phi}{\partial t} = -u\frac{\partial \phi}{\partial x} > 0$$

and the value of ϕ in $x = -1$ will decrease with time.

The derivative expresses a local property of the solution, so the we can only say the the value of ϕ in $x = -1$ will increase with time "for a short time".

First order linear equation

Advection equation. Case 3: $\frac{\partial \phi}{\partial x} > 0$ and $u < 0$



In $x = -1$ we have $\frac{\partial \phi}{\partial x} > 0$. Thus, if $u < 0$:

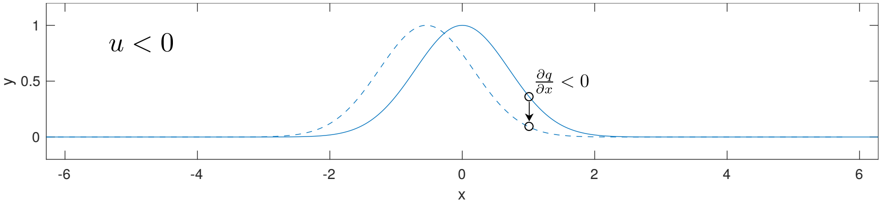
$$\frac{\partial \phi}{\partial t} = -u \frac{\partial \phi}{\partial x} > 0$$

and the value of ϕ in $x = 1$ will increase with time.

The derivative expresses a local property of the solution, so the we can only say the the value of ϕ in $x = -1$ will increase with time "for a short time".

First order linear equation

Advection equation. Case 4: $\frac{\partial\phi}{\partial x} < 0$ and $u < 0$



In $x = -1$ we have $\frac{\partial\phi}{\partial x} < 0$. Thus, if $u < 0$:

$$\frac{\partial\phi}{\partial t} = -u\frac{\partial\phi}{\partial x} < 0$$

and the value of ϕ in $x = 1$ will decrease with time.

The derivative expresses a local property of the solution, so the we can only say the the value of $q\phi$ in $x = -1$ will decrease with time "for a short time".

First order linear equation

Advection equation

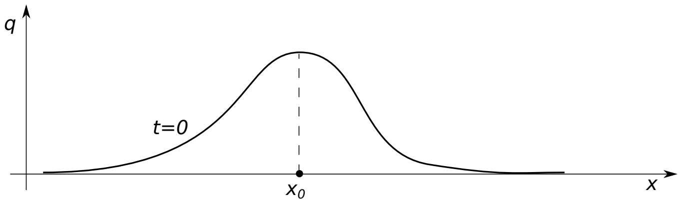
We consider again the advection equation with constant velocity u:

$$\frac{\partial \phi}{\partial t} + \frac{\partial}{\partial x}(u\phi) = 0$$

or

$$\frac{\partial \phi}{\partial t} = -u\frac{\partial \phi}{\partial x}$$

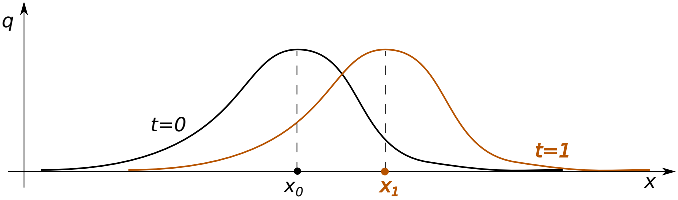
We plot the value of the ϕ versus x at the time $t = 0$, and we denote with x_0 the location of the maximum of the concentration.



First order linear equation

Advection equation

At the time $t = 1$, if the velocity u is positive, the concentration will be shifted to the right (red plot).



The maximum value of the concentration will be at $x_1 = x_0 + u \cdot t$.
The initial profile $\phi_0(x)$ simply translates (shifts) without changing shape along the x-axis with velocity u.

- If $u > 0$, the profile moves to the right.
- If $u < 0$, the profile moves to the left.

Exact Solution and Characteristic Curves

Given an initial condition $\phi(x, t = 0) = \phi_0(x)$, the exact solution to the linear advection equation is:

$$\phi(x, t) = \phi_0(x - ut)$$

This means the initial profile $\phi_0(x)$ simply translates (shifts) without changing shape along the x-axis with velocity u.

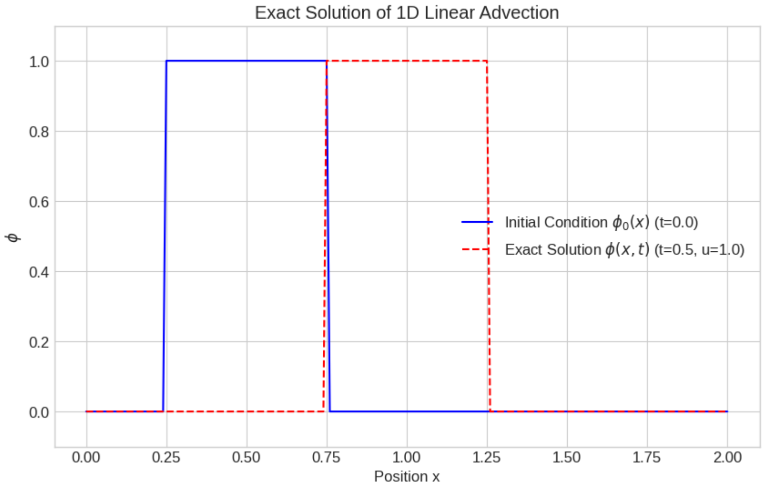
- If $u > 0$, the profile moves to the right.
- If $u < 0$, the profile moves to the left.

Let's visualize this exact solution for a sample initial condition:

$$\phi_0(x) = \begin{cases} \phi_{high} & \text{if } x_{left} \leq x \leq x_{right} \\ \phi_{low} & \text{otherwise} \end{cases}$$

This represents a square wave.

Exact Solution and Characteristic Curves



Outline

Boundary Conditions for Advection?

The linear advection equation, $\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0$, describes how a quantity ϕ evolves due to transport by a constant velocity u *within* a defined spatial domain.

However, for a problem defined on a finite spatial domain (e.g., $x \in [0, L]$):

- The PDE alone does not provide a unique solution. We need to specify what happens at the domain's edges ($x = 0$ and $x = L$).
- Boundary Conditions (BCs) provide this crucial information, representing:
  - How the system interacts with its surroundings.
  - How "information" (the value of ϕ) enters or leaves the computational domain.
- The choice of BCs depends heavily on the physical nature of the problem being modeled.

Recap: Dirichlet and Neumann Conditions

From Lecture 3, we recall two common types of boundary conditions:

1. Dirichlet (Fixed Value) BC: The value of ϕ is specified at the boundary.

$$\phi(x_{\text{boundary}}, t) = \phi_{\text{specified}}(t)$$

Example for Advection (Inflow): If fluid enters the domain at $x = 0$ with a known concentration of a tracer ϕ_{in}:

$$\phi(0, t) = \phi_{in}(t)$$

2. Neumann (Fixed Gradient/Flux) BC: The spatial gradient of ϕ is specified at the boundary.

$$\left.\frac{\partial \phi}{\partial x}\right|_{x_{\text{boundary}}} = g_{\text{specified}}(t)$$

Example for Advection (Outflow): A common "open" or "outflow" condition is zero gradient, suggesting the profile exits smoothly.

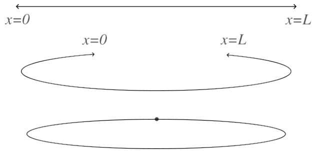
For purely advective problems, outflow conditions need careful consideration to avoid artificial reflections.

New Type: Periodic Boundary Conditions

For certain problems different type of BC is very useful: Periodic BC.
Definition: The value of ϕ and its spatial derivatives at one end of the domain are set equal to the values at the other end. For a 1D domain $[0, L]$:

- $\phi(0, t) = \phi(L, t)$

Intuitive Concept: Imagine the domain is "wrapped around" to form a circle or a loop. What flows out of one end ($x = L$) immediately re-enters at the other end ($x = 0$).

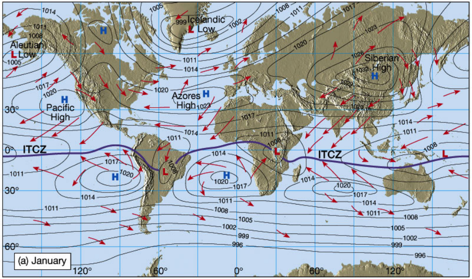


There is no "external" influence at the boundaries; the system is self-contained in a loop.

Geoscience Examples of Periodic Systems

While truly periodic systems are idealized, periodic BCs are useful approximations or tools for:

- Global Atmospheric/Oceanic Models (Zonal Averages): Modeling flow or tracer transport along a latitude band. After circling the Earth, the conditions should "reconnect".



- Numerical Method Testing: To study how a scheme advects a feature over long distances without boundary interference.

Implementing Periodic BCs Numerically (Finite Differences)

Consider a 1D grid with N_x points, indexed $i = 0, 1, \dots, N_x - 1$. The physical domain is from x_0 to x_{N_x-1}. Δx is the grid spacing.
To apply periodic BCs, we often use "ghost cells" (or direct indexing):

- At the left boundary (node $i = 0$): When a scheme needs a value from "left" of x_0 (e.g., x_{-1}), we use the value from the *other end* of the physical domain.

$$\phi^n_{-1} \equiv \phi^n_{N_x-1}$$

  (If N_x is the number of cells and nodes are cell-centers, it might be $\phi^n_{N_x-1}$. If nodes are $0..N_x$, and $i = 0$ is the first physical node, $i - 1$ could be N_x). Let's assume nodes $0, \dots, N_x - 1$. So, value at "virtual" node x_{-1} is taken from x_{N_x-1}.
- At the right boundary (node $i = N_x - 1$): When a scheme needs a value from "right" of x_{N_x-1} (e.g., x_{N_x}), we use the value from the *beginning* of the physical domain.

Advantages of Periodic BCs for Testing Advection Schemes

Periodic boundary conditions are particularly valuable when analyzing and testing numerical schemes for advection:

- Eliminate Boundary-Induced Errors: They prevent artificial reflections or damping of waves/profiles that can occur with fixed Dirichlet or Neumann BCs if not carefully chosen.
- Focus on Intrinsic Scheme Properties: By removing "real" boundary effects, we can better observe the inherent numerical diffusion (smearing) and dispersion (oscillations) produced by the scheme itself.
- Long-Term Simulations: Ideal for simulating phenomena over multiple "passes" through the domain, allowing accumulation of errors to become more apparent.
- Conservation Studies: It's easier to check for global conservation of quantities like $\int \phi \, dx$ because, with periodic BCs, whatever mass/energy flows out one side perfectly re-enters the other, so the total amount in the domain should remain constant (if the scheme is conservative).

Outline

Recap: Approximating First Spatial Derivatives

For the advection equation, $\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0$, we need to approximate the spatial derivative $\frac{\partial \phi}{\partial x}$. From previous lectures, we recall:

- Central Difference: (Uses ϕ_{i-1}, ϕ_{i+1})

$$\left.\frac{\partial \phi}{\partial x}\right|_i \approx \frac{\phi_{i+1} - \phi_{i-1}}{2\Delta x} \quad (\text{Order } \mathcal{O}(\Delta x^2))$$

- Forward Difference: (Uses ϕ_i, ϕ_{i+1})

$$\left.\frac{\partial \phi}{\partial x}\right|_i \approx \frac{\phi_{i+1} - \phi_i}{\Delta x} \quad (\text{Order } \mathcal{O}(\Delta x))$$

- Backward Difference: (Uses ϕ_i, ϕ_{i-1})

$$\left.\frac{\partial \phi}{\partial x}\right|_i \approx \frac{\phi_i - \phi_{i-1}}{\Delta x} \quad (\text{Order } \mathcal{O}(\Delta x))$$

The choice of spatial approximation, combined with time discretization, defines our numerical scheme.

Time Discretization: Forward Euler

For the time derivative $\frac{\partial \phi}{\partial t}$, we will consistently use the Forward Euler approximation for the explicit schemes discussed in this section:

$$\left.\frac{\partial \phi}{\partial t}\right|_i^n \approx \frac{\phi_i^{n+1} - \phi_i^n}{\Delta t}$$

This approximation is first-order accurate in time, with a truncation error of $\mathcal{O}(\Delta t)$.
Our goal is to derive explicit update formulas of the form:

$$\phi_i^{n+1} = \text{function}(\text{values of } \phi^n, u, \Delta t, \Delta x)$$

where ϕ_i^n represents the value of ϕ at spatial node x_i and time level t^n.
This formula represent an explicit scheme, because ϕ_i^{n+1} is expressed explicitly in terms on known quantities.

Scheme 1: Forward Time, Centered Space (FTCS)

We combine Forward Euler in time with a Central Difference in space for the advection equation $\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0$:

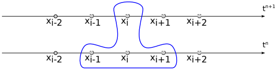
$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} + u\frac{\phi_{i+1}^n - \phi_{i-1}^n}{2\Delta x} = 0$$

Solving for ϕ_i^{n+1}, we get the explicit update formula:

FTCS Update Formula

$$\phi_i^{n+1} = \phi_i^n - \frac{u\Delta t}{2\Delta x}(\phi_{i+1}^n - \phi_{i-1}^n)$$

- Stencil: Uses $\phi_{i-1}^n, \phi_i^n, \phi_{i+1}^n$ to find ϕ_i^{n+1}.
- Truncation Error: $\mathcal{O}(\Delta t, \Delta x^2)$. The 2nd oreder spatial accuracy might seem advantageous.



FTCS: Python Implementation Snippet

The core of the FTCS scheme involves updating interior points based on their neighbors at the previous time step.

Python Code (Core Loop for FTCS with Periodic BCs) - PART 1

```
# nx: number of grid points (0 to nx-1)
# phi_old: array with values at time n
# phi_new: array for values at time n+1

for i in range(nx): # Loop over ALL physical points
    # Determine periodic indices for i-1 and i+1
    if i == 0:
        im1 = nx - 1  # Left neighbor of phi_old[0]
    else:
        im1 = i - 1
```

FTCS: Python Implementation Snippet (Periodic BCs)

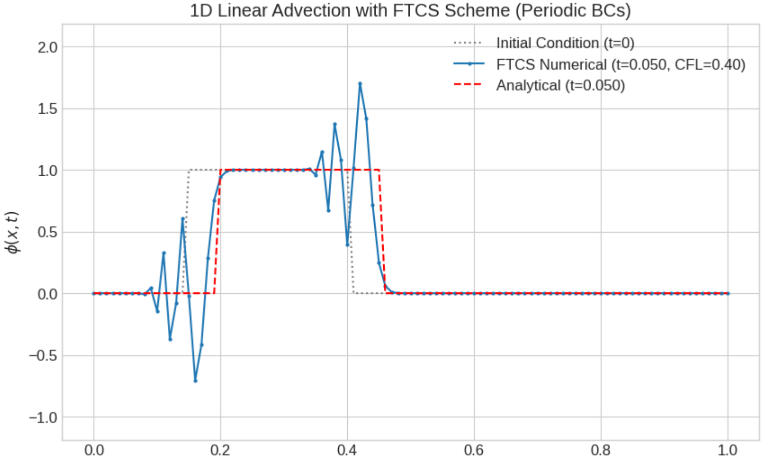
Python Code (Core Loop for FTCS with Periodic BCs) - PART 2

```
    if i == nx - 1:
        ip1 = 0       # Right neighbor of phi_old[nx-1]
    else:
        ip1 = i + 1

    phi_new[i] = phi_old[i] - \
                 u * dt / (2 * dx) \
                 * (phi_old[ip1] - phi_old[im1])

# After loop: phi = phi_new.copy()
```

This ensures that information leaving one boundary re-enters from the opposite side.

FTCS: Numerical Result - Unstable Behavior!

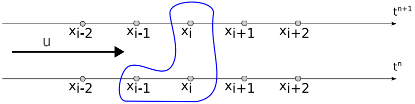
Let's attempt to advect an initial Gaussian pulse using FTCS. We set $u = 1$, $\Delta x = 0.01$, and Δt such that the Courant number $C = u\Delta t/\Delta x = 0.5$. Boundary conditions are periodic.

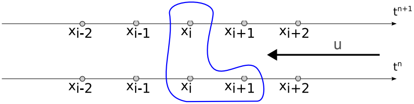


Upwind Schemes: Following the Flow of Information

The advection equation $\frac{\partial \phi}{\partial t} + u\frac{\partial \phi}{\partial x} = 0$ describes information (the value of ϕ) being carried by the flow at velocity u.

- Physical Insight: The value of ϕ at point x_i at the new time t^{n+1} should be determined by the value of ϕ "upwind" (or "upstream") at the previous time t^n.
- If $u > 0$ (flow is to the right): Information for x_i arrives from its left ($x < x_i$). We should use a backward difference for $\frac{\partial \phi}{\partial x}$.
- If $u < 0$ (flow is to the left): Information for x_i arrives from its right ($x > x_i$). We should use a forward difference for $\frac{\partial \phi}{\partial x}$.





Upwind schemes use one-sided spatial differences that respect this direction of information flow.

Upwind Scheme for $u > 0$: FTBS

When the advection velocity $u > 0$ (flow to the right):

- Time derivative: Forward Euler $\approx \frac{\phi_i^{n+1} - \phi_i^n}{\Delta t}$
- Spatial derivative (upwind): Backward Difference $\approx \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x}$

Substituting into $\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} = 0$:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} + u \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x} = 0$$

Solving for ϕ_i^{n+1} gives the FTBS (or 1st Order Upwind for $u > 0$) scheme:

FTBS Update Formula ($u > 0$)

$$\phi_i^{n+1} = \phi_i^n - \frac{u \Delta t}{\Delta x} (\phi_i^n - \phi_{i-1}^n)$$

- Stencil: Uses ϕ_{i-1}^n, ϕ_i^n to find ϕ_i^{n+1}.
- Truncation Error: $\mathcal{O}(\Delta t, \Delta x)$ (first-order in both time and space).

FTBS (Upwind $u > 0$): Python Implementation Snippet

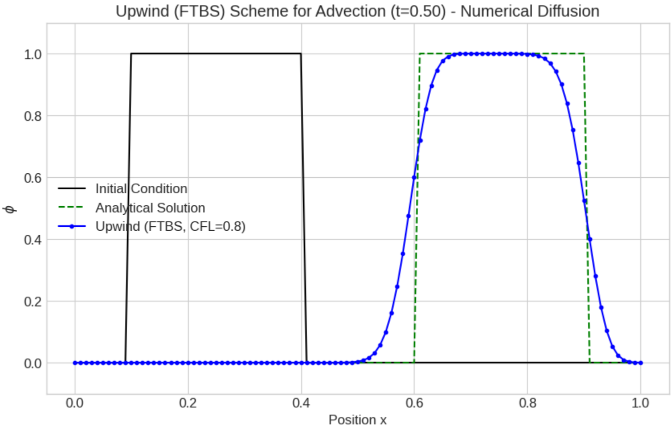
Core loop for the FTBS scheme, assuming $u > 0$.

Python Code (Core Loop for FTBS, $u > 0$)

```
phi_new_upwind = np.zeros_like(phi_current_upwind)
for n in range(nt_upwind):
    phi_old_upwind = phi_current_upwind.copy()

    # Periodic Boundary Conditions
    for i in range(nx_upwind): # Loop through all points
        # Determine index for phi[i-1] with periodicity
        im1 = (i - 1 + nx_upwind) % nx_upwind # Modulo operator

        phi_new_upwind[i] = phi_old_upwind[i] - \
                            CFL_upwind * (phi_old_upwind[i] - \
                            phi_old_upwind[im1])

    phi_current_upwind = phi_new_upwind.copy()
```

FTBS (Upwind $u > 0$): Numerical Result - Numerical Diffusion

Advecting a Gaussian pulse with FTBS ($u = 1$, periodic BCs, Courant number $C = u\Delta t/\Delta x = 0.8$).



Upwind Scheme for $u < 0$: FTFS

If the advection velocity $u < 0$ (flow is to the left):

- Information for x_i comes from its right ($x > x_i$).
- Spatial derivative (upwind): Forward Difference $\approx \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x}$.

The update formula becomes the FTFS (or 1st Order Upwind for $u < 0$) scheme:

FTFS Update Formula ($u < 0$)

$$\phi_i^{n+1} = \phi_i^n - \frac{u\Delta t}{\Delta x}(\phi_{i+1}^n - \phi_i^n)$$

- Stencil: Uses ϕ_i^n, ϕ_{i+1}^n to find ϕ_i^{n+1}.
- Truncation Error: $\mathcal{O}(\Delta t, \Delta x)$.
- Also stable under its CFL condition ($0 \leq -C \leq 1$, i.e. $|C| \leq 1$) and numerically diffusive.

Numerical Artifacts: A Closer Look

Numerical schemes for advection can introduce errors that alter the solution's shape: Numerical Diffusion (Dissipation):

- Causes sharp gradients to smear out and peaks to diminish in amplitude.
- Acts like an artificial (unphysical) diffusion process.
- Prominent in first-order schemes like Upwind.

Numerical Dispersion:

- Different wavelength components of the solution travel at incorrect speeds.
- Can lead to spurious oscillations (wiggles), especially near sharp fronts.
- Can be an issue with centered schemes (like FTCS if it were stable) or some higher-order schemes if not carefully designed.

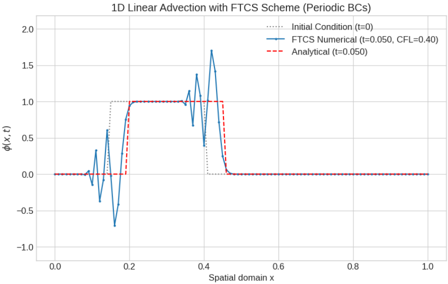
The goal is to choose/design schemes that minimize these unwanted artifacts while maintaining stability.

Outline

What is Numerical Stability? A Critical Property

We've seen numerical schemes approximate PDEs. But how do we ensure these approximations are reliable?

- Numerical Stability means that small errors introduced at one step (from truncation, round-off, or initial data) do not grow uncontrollably and overwhelm the true solution as the computation proceeds.
- An unstable scheme will produce solutions that diverge, often with wild oscillations, rendering the results meaningless.
- We witnessed this with the FTCS scheme for advection:



The Courant Number: A Key Parameter for Advection

For explicit schemes solving hyperbolic PDEs like the advection equation, stability is often governed by a dimensionless parameter called the Courant-Friedrichs-Lewy (CFL) number, or simply the Courant number, C:

Courant Number Definition

$$C = \frac{|u|\Delta t}{\Delta x}$$

Physical Interpretation:

- C represents the distance $|u|\Delta t$ traveled by a signal in one time step, measured in units of grid cells Δx.
- If $C = 0.5$: The signal travels half a grid cell in one Δt.
- If $C = 1.0$: The signal travels exactly one grid cell in one Δt.
- If $C > 1.0$: The signal travels more than one grid cell in one Δt.

The CFL Condition

The Courant-Friedrichs-Lewy (CFL) condition is a necessary (but not always sufficient) condition for the stability and convergence of many explicit numerical schemes for hyperbolic PDEs.

Based on formal stability analysis (like Von Neumann analysis), here are the stability conditions for the explicit schemes we have introduced:

- FTCS (Forward Time, Centered Space):
  - Condition: Unconditionally Unstable for $u \neq 0$.
  - It cannot be stabilized by adjusting C.
- Upwind (1st Order: FTBS if $u > 0$, FTFS if $u < 0$):
  - Condition: Stable if $0 \leq C \leq 1$.
  - Where $C = \frac{u\Delta t}{\Delta x}$ for $u > 0$, or $C = \frac{(-u)\Delta t}{\Delta x}$ for $u < 0$. More generally, $\frac{|u|\Delta t}{\Delta x} \leq 1$.

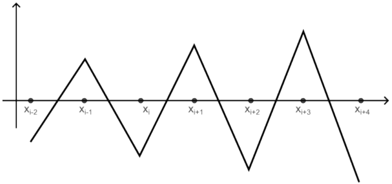
Why is FTCS Unstable for Advection? (Intuitive Idea)

A rigorous proof of FTCS instability requires Von Neumann analysis. However, we can get some intuition:

- Centered Stencil "Blindness": The centered difference for $\frac{\partial \phi}{\partial x}$ uses ϕ_{i-1}^n and ϕ_{i+1}^n. It averages information from both upwind and downwind directions.
- Advection is Directional: For pure advection, information strictly flows from upwind. The FTCS scheme doesn't inherently "know" or respect this directionality.
- Analogy to Heat Equation (Diffusion):
  - For the heat equation $\frac{\partial T}{\partial t} = \kappa \frac{\partial^2 T}{\partial x^2}$, the FTCS scheme IS conditionally stable ($\kappa \Delta t / \Delta x^2 \leq 1/2$).
  - Diffusion is a process where information spreads in all directions (smoothing). A centered stencil for $\frac{\partial^2 T}{\partial x^2}$ is appropriate here.
- The combination of a forward-in-time step with a centered-in-space step for a purely directional (hyperbolic) process like advection leads to this instability.

FTCS Instability: The "Sawtooth Wave" Argument (Heuristic)

Consider a "sawtooth" with increasing amplitude as an initial condition, e.g., $\phi_i^n = (-1)^i A_i$ (with $A_i > 0$ and $A_{i+1} > A_i$).



The FTCS update is $\phi_i^{n+1} = \phi_i^n - \frac{C}{2}(\phi_{i+1}^n - \phi_{i-1}^n)$.

- For i odd, $\phi_i^n = -A_i < 0$: $\phi_{i+1}^n = A_{i+1}$, $\phi_{i-1}^n = A_{i-1}$.
  $\phi_i^{n+1} = A_i - \frac{C}{2}(A_{i+1} - A_{i-1}) < A_i = \phi_i^n$.

  The negative peaks decrease (amplification of the peak).
- For i even, $\phi_i^n = A_i > 0$: $\phi_{i+1}^n = -A_{i+1}$, $\phi_{i-1}^n = -A_{i-1}$.
  $\phi_i^{n+1} = A_i - \frac{C}{2}(-A_{i+1} - (-A_{i-1})) = A_i - \frac{C}{2}(A_{i-1} - A_{i+1}) > A_i = \phi_i^n$.

  The positive peaks decrease (amplification of the peak).

Stability of the Upwind Scheme (FTBS, $u > 0$) (1/3)

Let's analyze the stability of the first-order Upwind scheme (FTBS for $u > 0$):

$$\phi_i^{n+1} = \phi_i^n - C(\phi_i^n - \phi_{i-1}^n)$$

where $C = \frac{u\Delta t}{\Delta x}$ is the Courant number (assume $u > 0$, so $C \geq 0$).
We can rewrite this equation by grouping terms involving ϕ^n:

$$\begin{aligned} \phi_i^{n+1} &= \phi_i^n - C\phi_i^n + C\phi_{i-1}^n \\ \phi_i^{n+1} &= (1 - C)\phi_i^n + C\phi_{i-1}^n \quad (*) \end{aligned}$$

This form shows that the new value ϕ_i^{n+1} is a linear combination of two values from the previous time step: ϕ_i^n and ϕ_{i-1}^n.

Intuition for Stability (Boundedness):

- If the solution ϕ^n is bounded at time n (i.e., $m \leq \phi_j^n \leq M$ for all j), we want ϕ^{n+1} to remain bounded by the same m and M.

Let's examine these coefficients.

Stability of the Upwind Scheme (FTBS, $u > 0$) (2/3)

From the rewritten Upwind (FTBS) scheme:

$$\phi_i^{n+1} = \underbrace{(1 - C)}_{\text{coeff of } \phi_i^n} \phi_i^n + \underbrace{C}_{\text{coeff of } \phi_{i-1}^n} \phi_{i-1}^n$$

For this to be a stable update (in the sense of not amplifying errors or creating new extrema), we require the coefficients to be non-negative:

1. Coefficient of ϕ_{i-1}^n is C: Since $u > 0$, $\Delta t > 0$, and $\Delta x > 0$, we have $C = \frac{u\Delta t}{\Delta x} \geq 0$. This condition is always met.
2. Coefficient of ϕ_i^n is $(1 - C)$: We require $1 - C \geq 0$, which implies $C \leq 1$.

Combining these, for the coefficients to be non-negative, we need:

$$0 \leq C \leq 1$$

This is precisely the CFL condition for the stability of the first-order Upwind scheme (when $u > 0$).

Stability of the Upwind Scheme (FTBS, $u > 0$) (3/3)

$$0 \leq \frac{u\Delta t}{\Delta x} \leq 1$$

Interpretation:

- If $0 \leq C \leq 1$, then ϕ_i^{n+1} is a weighted average of ϕ_i^n and ϕ_{i-1}^n with non-negative weights. This prevents the solution from growing beyond the maximum or falling below the minimum of the previous step's values (it satisfies a "maximum principle" and ensures monotonicity for certain initial data).
- This argument, while not a full Von Neumann analysis, demonstrates why the CFL condition $C \leq 1$ is crucial for the good behavior of this scheme. It ensures that errors are not amplified in magnitude.
- A similar argument holds for $u < 0$ with the FTFS scheme.

CFL Condition: Practical Implications in Geosciences

The stability condition $\Delta t \leq \frac{C_{\max}\Delta x}{|u|}$ has significant practical consequences:

- Computational Cost: To resolve fine spatial features (small Δx), Δt must be small, too. This means many more time steps are needed to simulate the same physical duration.
- High Velocities: In systems with high physical velocities $|u|$ (e.g., pyroclastic flows, fast rivers, strong winds carrying ash), Δt must be kept small to maintain stability.
- Variable Grids/Velocities: If Δx or u vary within the domain, Δt must be chosen based on the *most restrictive* combination (smallest Δx, largest $|u|$) to ensure stability everywhere.
- Choice of Scheme:
  - Explicit schemes (like those discussed) are often limited by CFL.
  - Implicit schemes (not covered) can often use larger Δt (less restrictive or no CFL limit for stability), but each time step is computationally more expensive as it requires solving a system of equations.

How to Choose Δt Based on CFL (for Explicit Schemes)

A practical workflow for selecting Δt:

1. Select your numerical scheme (e.g., Upwind). This determines $C_{\max}$ (e.g., $C_{\max} \approx 1$).
2. Determine your spatial resolution Δx. This is usually based on the smallest features you need to resolve in your geoscience problem.
3. Estimate the maximum velocity $|u|_{\max}$ expected in your simulation domain.
4. Calculate the stability-limited time step Δt_{stable}:

$$\Delta t_{\text{stable}} = \frac{C_{\max} \cdot \Delta x}{|u|_{\max}}$$

5. Choose your operational Δt: Select a Δt slightly smaller than Δt_{stable} to provide a safety margin (e.g., $\Delta t = 0.8 \cdot \Delta t_{\text{stable}}$ or $\Delta t = 0.9 \cdot \Delta t_{\text{stable}}$).
6. Always Test and Verify! Even if CFL is satisfied, check if results are physically reasonable and converge with grid refinement.

Outline

Summary: Stability and the CFL Condition

- Numerical stability is essential: errors must not grow uncontrollably.
- The Courant Number $C = \frac{|u|\Delta t}{\Delta x}$ relates time step, space step, and velocity.
- The CFL Condition ($C \leq C_{\max}$) provides an upper limit on Δt for explicit hyperbolic schemes to be stable.
- $C_{\max}$ depends on the specific scheme:
  - FTCS: Unconditionally unstable for pure advection.
  - Upwind (1st order): Stable for $C \leq 1$.
- Violating the CFL condition typically leads to unstable, non-physical solutions.
- Respecting CFL is necessary for stability, but accuracy (minimizing numerical diffusion/dispersion) often requires C to be somewhat less than $C_{\max}$ and small Δx.

Beyond Formal Accuracy: Choosing the Right Numerical Approach

Throughout this course, we've explored various numerical schemes and analyzed their properties, like order of accuracy and stability.

Is Higher Order Always Better?

- While higher-order schemes (e.g., $\mathcal{O}(\Delta x^2)$ vs. $\mathcal{O}(\Delta x)$) promise faster convergence to the exact solution as $\Delta x \to 0$, this is not the only factor.
- We saw that the FTCS scheme for advection, despite being second-order in space, is unconditionally unstable and thus unusable.
- First-order upwind schemes, though less accurate formally and more diffusive, are often robust and simple, making them a good starting point or suitable for problems where sharp fronts are not critical or are already smoothed by physical diffusion.

Beyond Formal Accuracy: Choosing the Right Numerical Approach

The "Best" Scheme is Problem-Dependent: There is no single "best" numerical scheme for all problems.
The choice depends on:

- The nature of the PDE itself (e.g., hyperbolic, parabolic, elliptic; linear, nonlinear).
- The characteristics of the expected solution (e.g., smooth, sharp fronts, shocks, oscillations).
- The required accuracy and available computational resources.
- Desired properties like conservation or monotonicity (non-oscillatory behavior).

The Numerical Modeller's Workflow: part 1

Effective numerical modelling in geosciences involves more than just picking a scheme:

1. Understand the Physics:
   - Deeply understand the dynamics of the Earth science process you are studying. What are the dominant forces and transport mechanisms? What are the key scales involved?
2. Formulate the Mathematical Model:
   - Translate the physical understanding into a set of governing equations (often PDEs), including appropriate initial and boundary conditions.
3. Verify the Mathematical Model (if possible):
   - Does the mathematical model (the PDEs + IC/BCs) adequately describe the behavior of the phenomenon of interest? Are there known analytical solutions for simplified cases that can be used for validation?
4. Choose and Implement a Numerical Method:
   - Select a suitable numerical scheme based on the PDE type, solution characteristics, and desired accuracy/stability (as discussed).
   - Implement it carefully, paying attention to details like boundary condition handling.

The Numerical Modeller's Workflow: part 2

Effective numerical modelling in geosciences involves more than just picking a scheme:

5. Verify and Validate the Numerical Solution:
    - Verification: "Are we solving the equations right?" Does the numerical solution converge to the known analytical solution of the *discretized equations* (or a manufactured solution) as $\Delta t, \Delta x \to 0$? This checks for bugs and consistency.
    - Validation: "Are we solving the right equations?" Does the numerical solution of the mathematical model compare well with experimental data, field observations, or analytical solutions of the *original physical problem*?
6. Analyze and Interpret Results:
    - Critically assess the numerical results in the context of the physical problem. Are they physically reasonable? What insights do they provide?

Numerical modelling is an iterative process of refinement and learning.