Forward Difference Equation

- From last time:
  - a mathematical model for simple advection (1D motion of fluid):
    \[
    \frac{\partial \Phi(x,t)}{\partial t} + u \frac{\partial \Phi(x,t)}{\partial x} = 0
    \]
  - use Taylor series expansion to derive a discrete form of the PDE:
    \[
    \frac{\Phi_{j,n+1} - \Phi_{j,n}}{\Delta t} + u \frac{\Phi_{j+1,n} - \Phi_{j,n}}{\Delta x} = 0
    \]
  - the discrete form defines a 2D grid
    - rows are time steps, columns are spatial positions
    - \( \Phi_{j,n} \) is density of particles at position \( x = j \) at time \( t = n \)

Stencils

- The pattern of relationships between terms used to compute the value at a grid point is called a **stencil**

- For the forward difference equation:

\[
\Phi_{j,n+1} = \Phi_{j,n} - u \frac{\Delta t}{\Delta x} (\Phi_{j+1,n} - \Phi_{j,n})
\]

Convenient graphical notation to compare different discrete forms

Illustrates why this is “forward difference”
**Characteristics**

- In the exact solution of the advection equation it would be possible to plot the progress of any point in the initial density distribution.
  - A curve that follows the path of a single value from the initial function is a **characteristic**.
  - “Curve of information propagation”
  - In this model the characteristic is a line, with a slope that depends on \( u \).

\[
x = u t + x_i
\]

**Solution Region**

- Plotting the characteristics of each point in the initial distribution defines a region where we have information for a solution.
  - A large region to the left of the characteristic for \( x_0 \) is undefined.

**Periodic Boundary Conditions**

- To provide information for this undefined region use periodic boundary conditions.
  - The stencil for the forward difference equation uses values from the right.
  - Copy the corresponding information from the right edge to the left.
- May not be realistic.
  - Particles from Pasadena don’t just reappear in Santa Monica.
- For abstract models (e.g., general groundwater flow) it may be OK.
  - Useful for evaluating models and solutions.

**Evaluating the Forward Difference Equation**

- This plot shows an initial distribution and expected vs computed values after one time step.
  - Not too good....
  - After a few time steps the distribution is unrecognizable (see Fosdick chapter).
Backward Differences

- In deriving the forward difference equation we started with an estimate for a value of $\phi(x + \Delta x, t)$:
  $$\phi(x + \Delta x, t) = \phi(x, t) + \phi_t(x, t)\Delta x + O((\Delta x)^2)$$
- We could have asked for a prediction of $\phi(x - \Delta x, t)$:
  $$\phi(x - \Delta x, t) = \phi(x, t) - \phi_t(x, t)\Delta x + O((\Delta x)^2)$$
- Solve for $\phi_t(x, t)$, and combine with the old equation for $\phi_t(x, t)$

Backward Differences (cont’d)

- The new equation is slightly different, but has a big effect on accuracy
  $$\phi_{j,n+1} = \phi_{j,n} - \mu \frac{\Delta t}{\Delta x}(\phi_{j,n} - \phi_{j-1,n})$$

Accuracy of Backward Difference Equation

- Much better after one time step
- But why?

Backward Difference and Characteristic

- Short answer to why backward is better:
  - when $u > 0$ the distribution is moving right
  - getting information from the left (direction of flow) is more accurate
- (Slightly) more technical answer:
  - the characteristic that goes through a point also passes over paths used to compute the value for the point
\[ t = N \]

- With periodic boundary conditions the initial distribution will wrap around and overlay the original
- Comparing the two can provide a measure of accuracy
  - backward difference looked OK after a few time steps
  - not as good after a full cycle

**Frequency Components**

- Why does the triangular distribution flatten out?
- Ans: *dispersion*
  - Fourier transform of shapes with sharp edges (this triangle, square wave, ...) show lots of high frequencies
  - Numeric waves with higher frequencies disperse more slowly
- See Fosdick for details

**Aside: Testing Models**

- One of the difficulties in building computational models is testing
- For PDE solvers:
  - verify the PDE is a correct model of the system
  - evaluate the discretization
  - debug the software implementing the discretization
- This process is hard enough, but now we see a new complication:
  - pick a test distribution that has similar characteristics to system being modeled
  - smog won't likely be distributed as a square wave or other distribution with sharp edges
  - test with a Gaussian or other smooth pattern

**Other Difference Equations**

- Different applications of Taylor series expansion lead to other methods for numerically solving the equations
- Centered difference equation (aka "forward time centered space", or FTCS):

\[
\phi_{j,n+1} = \phi_{j,n} - \frac{\Delta t}{2\Delta x} (\phi_{j+1,n} - \phi_{j-1,n})
\]
Other Difference Equations (cont’d)

- Leapfrog
  - evaluation of a row requires two previous rows
  - use one of previous methods for $n = 1$
  - much better accuracy for triangle distribution (see Fosdick)

\[ \phi_{j,n+1} = \phi_{j,n-1} + u \frac{\Delta t}{\Delta x} (\phi_{j+1,n} - \phi_{j-1,n}) \]

Implicit Methods

- A difference equation that is useful in diffusion problems (next topic) compares values from the same time step
  - $u$ is the quantity being modeled
  - $D$ is a diffusion coefficient
  - this is a second-order equation, which is why there is a $(\Delta x)^2$ term

\[ u_{j,n+1} = u_{j,n} + \frac{D\Delta t}{(\Delta x)^2} (u_{j-1,n+1} - 2u_{j,n+1} + u_{j+1,n+1}) \]

Implicit Methods (cont’d)

- It may seem there isn’t any way to compute $u_{j,n+1}$
  - values in one column depend on values from both neighboring columns
  - The equation for an interior grid point is a linear equation
  - For the entire grid there are $m \times n$ equations but fewer than that many unknowns
  - we have initial values for boundary cells
  - Gaussian elimination or other methods to solve systems of linear equations can find values of grid points

Diffusion Problems

- Mathematicians classify PDEs according to the shape of the characteristics

\[ \frac{\partial^2 u}{\partial t^2} = v^2 \frac{\partial^4 u}{\partial x^2} \] hyperbolic (“wave equation”)

\[ \frac{\partial u}{\partial t} = - \frac{\partial}{\partial x} \left( D \frac{\partial u}{\partial x} \right) \] parabolic (“diffusion equation”)

\[ \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x,y) \] elliptic (Poisson, Laplace)
**Initial Value Problems**

- Our advection equation is a special case of a parabolic equation
  - no diffusion, i.e. wave keeps its shape as it moves
- From a computational point of view, hyperbolic and parabolic equations are very similar
  - both are derivatives with respect to time
  - used to describe systems that evolve over time
- Computational science term: *initial value problem*
- An important consideration for numeric solution:
  - presence or absence of diffusion
  - FTCS usually a poor choice for advection problems, but can be stable for diffusion problems

---

**Boundary Value Problems**

- Elliptic PDEs describe a different type of problem
  \[
  \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x,y)
  \]
  - Note time is not represented in this equation
- These equations describe the spatial distribution of some quantity, e.g. heat
- The goal is to describe the steady-state distribution of the quantity
- In computational science, these are *boundary value problems*
  - values at boundaries are known
  - goal is to compute values at internal points

---

**To Learn More**

- Numeric solution of PDEs is a huge area
  - an art as much as a science (e.g. using “upstream” vs “downstream” differences)
  - many techniques work well on some types of equations, poorly on others
  - could devote an entire second volume of NR to PDEs alone
- For more information:
  - many other books on numeric methods