Advection

Introduction to flow problems
PDE for advection
Finite difference method

Continuous Models

- Recall from the course introduction there are two major types of computational models
- Example: population dynamics
  - Continuous model: Lotka-Volterra equations
    - system of differential equations
    - numeric solvers evaluate the equations, predict population sizes at future times
  - Discrete model: individual-based simulation
    - cellular-automaton representation of space, individuals
    - state transitions define interactions
    - count objects at each time step to determine population sizes

Continuous Models (cont’d)

- Next unit this term: overview of continuous models
- Start with simple flow problem
  - advection: 2D ("horizontal") flow
  - applied to water (e.g. currents in oceanography), air (e.g. air pollution models), other problems that can be described in terms of flows
  - 3D systems model convection
- Show how system described by differential equations
- Introduction to techniques for integrating the equations

Reading

- Advection: An Introduction
  - chapter from Fosdick, et al (PDF on-line)
- Wilkinson and Allen, Ch 6
  - 6.3.2 describes the heat distribution problem (topic for project 3)
Advection

- From Britannica Online:
  
  advection (n.) ... in climatology, change in a property of a moving mass of air because the mass moves to a region where the property has a different value (e.g., the change in temperature when a warm air mass moves into a cool region). Attention can be centred on either the horizontal or vertical components of the motion, but the term advection commonly signifies only the horizontal transport of air.

- The Advection chapter describes a model for air pollution
- Property of interest: density of particulates at a specified time and location
- Given initial distribution and rules for air flow (direction, barriers, etc) we want to know particulate concentration at future times and locations

Example: L.A. Basin

- Below is a colorized satellite image of the Los Angeles region
- Ocean breezes typically move air from west to east
  - particulates generated during the day build up against the San Gabriel mountains

L.A. Basin (cont’d)

- Data from the South Coast Air Quality Management District (http://www.aqmd.gov)

L.A. Basin (cont’d)

- Measured particulate concentrations
Modeling Air Pollution

- A complete advection model of the build-up of particulates during the day would include:
  - initial conditions
    - distribution of particles at 12:00 A.M.
    - weather conditions (temp, humidity, ...)
  - terrain
    - elevations (mountains, ...)
    - surface (fields, lakes, concrete)
  - sources of particulates
    - factories
    - freeways
  - Complex 2D model of generation, movement of particulates

1D Model

- To get started we'll consider a very simple one-dimensional model
  - position: $x$
  - density of particles at location $x$ at time $t$: $\phi(x,t)$
  - velocity: $u$
  - $u > 0$ means particles are moving to the right

1D Model (cont’d)

- An equation that says the “air mass” moves to the right at a constant rate $u$:
  \[
  \frac{\partial \phi(x,t)}{\partial t} + u \frac{\partial \phi(x,t)}{\partial x} = 0
  \]

1D Model (cont’d)

- To see why this description works, think of time as a 3rd dimension
  \[
  \frac{\partial \phi(x,t)}{\partial t} + u \frac{\partial \phi(x,t)}{\partial x} = 0
  \]
Notation

- A common notation (used by Fosdick):

$$\frac{\partial \phi(x,t)}{\partial t} = \phi_t(x,t)$$
$$\frac{\partial \phi(x,t)}{\partial x} = \phi_x(x,t)$$

- The 1D model can then be written

$$\phi_t(x,t) + u \phi_x(x,t) = 0$$

Solution

- What does it mean to “solve” this equation?
- We want a method that will tell us $\phi(x,t)$ at any point $x$ and time $t$.
- Ideally we could integrate the differential equation in the model
- paper-and-pencil methods learned in math / physics classes
- symbolic math packages, e.g. Mathematica or Maple
- But for complicated equations or realistic systems we need a method for doing numeric integration
  - break the problem into small pieces, sum over all pieces

Aside: Numeric Integration

- There are several methods for estimating integrals of equations of one variable
- The goal is to calculate $I = \int_a^b f(x) \, dx$
  - area of curve below $f$
- A simple approximation (the “midpoint rule”) is to divide the interval from $a$ to $b$ into small rectangles and add the areas of the rectangles

$$I \approx \sum \Delta x$$

- Assumes that even if we can’t integrate $f(x)$ we can still evaluate $f(x)$ for any $x$...

Numeric Integration (cont’d)

- A better approximation (the “trapezoid rule”) makes small trapezoids instead of rectangles

$$I \approx \sum \frac{f(x_i) + f(x_i + \Delta x)}{2} \times \Delta x$$

Methods for Partial Differential Equations

- A similar idea is used for systems of equations of more than one variable
  - discretize the domains
  - estimate the value of the function at a point in terms of values at neighboring points
- Finite difference methods
  - divide domains (time, space) into fixed-size cells
- Finite element methods
  - finer mesh at critical points
  - may use triangles or other polygon shapes
  - "wire frame" outlines of model system

Finite Difference Models

- Commonly used in oceanography, meteorology, hydrology, and other areas with a uniform domain
- Advantages:
  - uniform calculations
  - easily vectorized
  - easily constructed
- Disadvantage:
  - may overlook fine details in some cells

Finite Element Models

- Finite element modeling uses mesh construction software to build models that conform to the shapes of objects

Grid for the Advection Problem

- We'll use a simple finite difference method for the advection problem
  - also for Laplace's equation (project 3)
- The function we’re integrating has two variables (space and time) so we have a 2D grid
- Goal: compute value at each cell in the grid
**Notation**

- In the discussion of methods for solving PDEs:
  - $x$ is a spatial dimension, indexed by $j$
  - $t$ is the time dimension, indexed by $n$
  - $\phi$ density (the function being integrated)
  - $\phi_x(j,n)$ partial derivative wrt $x$, evaluated at $j,n$
  - $\phi_{j,n}$ computed (approximate) value of density at $j,n$
  - $\phi_{0:j,n}$ density at all points at $t = n$ (a row of the grid)
  - $\phi_{j:0:N}$ density at all times at $x = j$ (a column of the grid)

Note order of subscripts:
- convention for geometry $(x, y)$
- but not matrices (row, col)

**Taylor Series Expansion**

- From analytic geometry:
  - if you can evaluate a function $f$ at a point $x$, and
  - the function is continuous and differentiable, then
  - you can estimate the value of $f$ at a nearby point $x + dx$

$$f(x + \Delta x) \approx f(x) + f'(x)\Delta x + \frac{f''(x)}{2!}\Delta x^2 + \frac{f'''(x)}{3!}\Delta x^3 + \cdots$$

**Difference Equations**

- The first-order expansion of the equation for density of particulates:
  - $\phi(x + \Delta x, t) = \phi(x, t) + \phi_x(x, t)\Delta x + O((\Delta x)^2)$
  - $\phi(x, t + \Delta t) = \phi(x, t) + \phi_t(x, t)\Delta t + O((\Delta t)^2)$

- expand each variable individually -- treat the other as a constant
- the "big-O" terms describe the size of the error
- a second-order expansion would have a smaller error term

- Rearrange a bit to get formulas for derivatives wrt $x$, $t$:
  - $\phi_x(x, t) = \frac{\phi(x + \Delta x, t) - \phi(x, t)}{\Delta x} + O(\Delta x)$
  - $\phi_t(x, t) = \frac{\phi(x, t + \Delta t) - \phi(x, t)}{\Delta t} + O(\Delta t)$
Forward Difference Equation

Plug these definitions of the derivatives into the PDE that describes advection:

\[ \phi_t(x,t) + u \phi_x(x,t) = 0 \]

\[ \frac{\phi_{j,n+1} - \phi_{j,n}}{\Delta t} + u \frac{\phi_{j+1,n} - \phi_{j,n}}{\Delta x} = 0 \]

- note use of notation for value of \( \phi \) at grid points
- the error terms have been left out

This equation is a **forward difference equation**

- \( \phi(j,n) \) combined with grid points at higher values of \( x, t \)

Forward Difference Equation (cont’d)

Rearrange the terms from this new, discretized, version of the PDE:

\[ \frac{\phi_{j,n+1} - \phi_{j,n}}{\Delta t} + u \frac{\phi_{j+1,n} - \phi_{j,n}}{\Delta x} = 0 \]

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

- We now have a formula we can use to compute values in grid cells
  - the LHS specifies values for row \( n + 1 \) of the grid
  - the RHS has terms from row \( n \)
  - algorithm: fill in row 0 with initial conditions, fill in remaining rows from bottom to top

Forward Difference Equation, C Style

\[
a = \frac{u \Delta t}{\Delta x}
\]

for \( j = 0..J-1 \)

\( \phi[0,j] = f(j) \)

for \( t = 0..N-1 \)

\( \phi[t,J] = \phi[t,0] \)

for \( j = 0..J-1 \)

\( \phi[t+1,j] = \phi[t,j] - a \cdot (\phi[t,j+1] - \phi[t,j]) \)

initial conditions boundary

Forward Difference Equation, Vector Style

\[
a = \frac{u \Delta t}{\Delta x}
\]

\( \phi[0,0:J-1] = f(0:J-1) \)

for \( t = 0..N-1 \)

\( \phi[t,J] = \phi[t,0] \)

\( \phi[t+1,0:J-1] = \phi[t,0:J-1] - a \cdot (\phi[t,1:J] - \phi[t,0:J-1]) \)

initial conditions boundary

(note periodic boundaries -- first step of main loop)
Back to Los Angeles

- For a 2D model of air flow in the LA Basin, use a 3D grid
  - $x_j$, $y_i$ are spatial coordinates
  - $t_n$ is the time coordinate
  - add terms for generation of particulates, barriers to flow, ...

To Be Continued...

- Next time:
  - evaluating the accuracy of the forward difference method
  - backward and centered difference equations
  - stencils and characteristics
  - explicit vs implicit solvers