PDEs and Diffusion

Carver J. Bierson

August 3, 2017

A partial differential equation is a equation that depends on the derivative with respect to two separate quantities. Just a few example PDEs are

$$\frac{\partial A}{\partial t} = u\frac{\partial A}{\partial x} \quad \text{Advection} \tag{1}$$

$$\frac{\partial A}{\partial t} = D\frac{\partial^2 A}{\partial x^2} \quad \text{Diffusion} \tag{2}$$

$$\frac{\partial^2 A}{\partial t^2} = c^2\frac{\partial^2 A}{\partial x^2} \quad \text{Wave equation} \tag{3}$$

$$\frac{\partial^2 A}{\partial x^2} + \frac{\partial^2 A}{\partial y^2} = f(x,y) \quad \text{Poisson's equation} \tag{4}$$

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u}\cdot\nabla)\vec{u} - \nu\nabla^2\vec{u} = \vec{f} \quad \text{Navier-Stokes equation} \tag{5}$$

Solving partial differential equations (PDEs) numerically is a huge topic. The form of the PDE (what order derivative, the number of different variables, etc) can have a huge effect on methods for solving the equation. To begin with we will solve the diffusion equation. The diffusion equation describes a wide variety of physical processes including heat transfer and the spread of chemical species.

1 Some Theory of Heat Transfer

Fourier's law of heat conduction states the energy flux through a material with a thermal gradient is

$$q = -k\frac{\partial T}{\partial x} \tag{6}$$

where T is the temperature, k is the thermal conductivity, and q is the heat flux. The heat flux has units of power per area ($\mathrm{W/m^2}$). The rate at which a slab will be warmed by incoming energy is given by

$$\rho c_p\frac{\partial T}{\partial t} = \frac{\partial q}{\partial x} \tag{7}$$

Here ρ is the density of the material and C_p is the specific heat at constant pressure. The reason there is another dx coming out here is a thicker slab will take longer to warm. By combining these and adding a heat production term we get

$$\rho c_p\frac{\partial T}{\partial t} = -\frac{\partial}{\partial x}\left(k\frac{\partial T}{\partial x}\right) + H \tag{8}$$

Here H has units of W/m^3. In the case where where $\partial k/\partial x$=0, we can define a diffusivity

$$\kappa = \frac{k}{\rho c_p} \tag{9}$$

κ has units of m^2/s and is an extremely useful quantity for getting a physical intuition for your system.

2 Discretizing Diffusion

For PDEs we need to discretize each part of the equation. We will simpfly our system slightly by assuming $\partial k/\partial x$=0. We can start with time derivative in the same way we did for ODEs.

$$\frac{\partial T}{\partial t} \approx \frac{\Delta_t T}{\Delta t} \tag{10}$$

The underscore t here indicates it is the change in the time dimension. Because T varies in both space and time I will use the i subscript to indicate space and j superscript to indicate time. Using this notation we can write

$$\frac{\Delta_t T}{\Delta t} = \frac{T_i^{j+1} - T_i^j}{\Delta t} \tag{11}$$

The space derivative is a second order derivative which we will discretize as

$$\frac{\partial}{\partial x}\left(\frac{\partial T}{\partial x}\right) \approx \frac{\Delta_x}{\Delta x}\left(\frac{\Delta_x T}{\Delta x}\right) \tag{12}$$

$$= \frac{1}{\Delta x}\left(\frac{T_{i+1}^j - T_i^j}{\Delta x} - \frac{T_i^j - T_{i-1}^j}{\Delta x}\right) \tag{13}$$

$$= \frac{T_{i+1}^j + T_{i-1}^j - 2T_i^j}{(\Delta x)^2} \tag{14}$$

We can combine these into the discrete equation

$$\frac{T_i^{j+1} - T_i^j}{\Delta t} = -\kappa\left[\frac{T_{i+1}^j + T_{i-1}^j - 2T_i^j}{(\Delta x)^2}\right] \tag{15}$$

This discretization is known as forward in time, centered in space (FTCS). Also note that this is basically equivalent to how we used the Euler discretization for ODEs. We have seen with ODEs that this solver can blow up if our timestep is too large. The key question then is how large can we make our timesteps (Δt) while keeping our solution stable. This condition can be found by what is known as *Von Neumann stability analysis*. This essentially asks the question what is the condition that ensures that the discretization error will not grow larger each timestep. This gives us what is known as the Courant–Friedrichs–Lewy (CFL) condition.

$$\Delta t \leq \frac{1}{2}\frac{(\Delta x)^2}{\kappa} \tag{16}$$

This is the timestep below which your error will not grow with each timestep. Keep in mind it does not tell you anything about how accurate will be. It is

common practice to use a timestep of

$$\Delta t = \frac{1}{3}\frac{(\Delta x)^2}{\kappa} \tag{17}$$

As this places you safely below that limit.

3 Implicit Methods

There are methods for solving the diffusion equation that are stable regardless of the timestep used (again stability is not accuracy). The best know of these is the backward time centered time (BTCS) implicit method. This discretization is

$$\frac{T_i^{j+1} - T_i^j}{\Delta t} = -\kappa \left[\frac{T_{i+1}^{j+1} + T_{i-1}^{j+1} - 2T_i^{j+1}}{(\Delta x)^2} \right] \tag{18}$$

Notice that in this form our space derivative depends on what the temperature will be after the timestep. At first glance this seems impossible to solve but it turns out to just by a set of simultaneous linear equations that we can solve with some linear algebra.

$$\left[-\frac{\kappa}{(\Delta x)^2} \right] T_{i+1}^{j+1} + \left[-\frac{\kappa}{(\Delta x)^2} \right] T_{i-1}^{j+1} + \left[\frac{1}{\Delta t} + 2\frac{\kappa}{(\Delta x)^2} \right] T_i^{j+1} = \frac{1}{\Delta t} T_i^j \tag{19}$$

$$\begin{bmatrix} a_1 & b_1 & 0 & 0 & 0 \\ b_2 & a_2 & b_2 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & b_{n-1} & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & b_n & a_n \end{bmatrix} \begin{bmatrix} T_1^{j+1} \\ T_2^{j+1} \\ \vdots \\ T_{n-1}^{j+1} \\ T_n^{j+1} \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_{n-1} \\ c_n \end{bmatrix} \tag{20}$$

where $a_i = 1/\Delta t + 2\kappa/(\Delta x)^2$, $b_i = -\kappa/(\Delta x)^2$, $c_i = 1/\Delta t T_i^j$. The coefficients at $i = 1$ and $i = n$ will often get modified to account for the boundary conditions. Coding up methods to invert the left most matrix can be a pain but we can use the numpy function *numpy.linalg.inv* to calculate the inverse matrix or *numpy.linalg.solve* to directly solve the system.

One popular method for solving diffusion problems is called *Crank-Nicholson*. In this method we essentialy take the average of FTCS and BTCS.

$$\frac{T_i^{j+1} - T_i^j}{\Delta t} = -\frac{\kappa}{2} \left[\frac{\left(T_{i+1}^{j+1} + T_{i-1}^{j+1} - 2T_i^{j+1} \right) + \left(T_{i+1}^{j} + T_{i-1}^{j} - 2T_i^{j} \right)}{(\Delta x)^2} \right] \tag{21}$$

This is solve the same way as BTCS only c_i now depends on the adjacent points as well.

4 Boundary conditions

For ODEs we needed to specify the initial conditions to solve our system. For PDEs we have boundary conditions. The information we need to provide is

- What is the initial state? i.e. the temperature at every point in our grid at $t = 0$.
- What happens at the grid edges? Is the temperature fixed at the edge, is there a heat flux coming from the edge?
- Source/sink terms. Heat production terms (radioactive decay) have to be explicitly added.

5 Problems

Lets start by solving the problem of a temperature profile that all starts at the same temperature, T_0, with one boundary ($x = 0$) fixed at a different temperature, T_1. This is a nice problem to start with because the exact solutioin is know to be

$$T(x,t) = T_0 + (T_1 - T_0)\,\mathrm{erfc}\left(\frac{x}{2\sqrt{\kappa t}}\right) \tag{22}$$

Here *erfc* is the complementary error function which can be found in *scipy.special.erfc*. **Write a diffusion solver and plot the error of your solution.** How does your error change with timestep?

Growing up the mountains of Colorado if you didn't bury your water pipes deep enough they would freeze. Lets build a code to answer the question how deep should you bury your pipes? Soil has a thermal diffusivity of $\kappa \approx 10^{-6}$ m^2/s. The Earth has a mean geothermal heat flux of 65 $\mathrm{mW/m}^2$. Assume the temperature at the surface is given by

$$T_s(t) = \bar{T} + T' \sin\left(\frac{2\pi}{365 * 24 * 3600} t\right) \tag{23}$$

Here t is in seconds, $\bar{T}$ is the yearly mean temperature and T' is the yearly temperature variation. For Colorado the mean temperature is about 10 °C (50 °F) with a variation of about 15 °C (30 °F).