Notes on Diffusion

These summarize methods for solving the diffusion equation.

1 Parabolic equations

The diffusion equation is

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial x}\left(k\frac{\partial \phi}{\partial x}\right) \tag{1}$$

This can describe thermal diffusion (for example, as part of the energy equation in compressible flow), species/mass diffusion for multispecies flows, or the viscous terms in incompressible flows. In this form, the diffusion coefficient (or conductivity), k, can be a function of x, or even ϕ. We will consider a constant diffusion coefficient:

$$\frac{\partial \phi}{\partial t} = k\frac{\partial^2 \phi}{\partial x^2} \tag{2}$$

The diffusion equation is the prototypical parabolic PDE. The basic behavior of the diffusion equation is to take strongly peaked concentrations of ϕ and smooth them out with time.

2 Explicit differencing

The simplest way to difference this equation is *explicit* in time (i.e. the righthand side depends only on the old state):

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = k\frac{\phi_{i+1}^n - 2\phi_i^n + \phi_{i-1}^n}{\Delta x^2} \tag{3}$$

This is second-order accurate in space, but only first order accurate in time (since the righthand side is not centered in time).

As with the advection equation, when differenced explicitly, there is a constraint on the timestep required for stability. Looking at the growth of a single Fourier mode, $\phi = A^n e^{ij\theta}$ with $j = \sqrt{-1}$, we find:

$$\frac{A^{n+1}}{A^n} = 1 + 2\frac{k\Delta t}{\Delta x^2}(\cos\theta - 1) \tag{4}$$

Stability requires that $|A^{n+1}/A^n| \leq 1$, which can only be true if $2k\Delta t/\Delta x^2 \leq 1$. Therefore, our timestep constraint in this case is

$$\Delta t < \frac{1}{2}\frac{\Delta x^2}{k} \tag{5}$$

Note the Δx^2 dependence—this constraint can become really restrictive.

To complete the solution, we need boundary conditions at the left (x_l) and right (x_r) boundaries. These are typically either Dirichlet:

$$\phi|_{x=x_l} = \phi_l \tag{6}$$
$$\phi|_{x=x_r} = \phi_r \tag{7}$$

or Neumann:

$$\phi_x|_{x=x_l} = \phi_x|_l \tag{8}$$
$$\phi_x|_{x=x_r} = \phi_x|_r \tag{9}$$

3 Implicit with direct solve

A backward-Euler implicit discretization would be:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = k \frac{\phi_{i+1}^{n+1} - 2\phi_i^{n+1} + \phi_{i-1}^{n+1}}{\Delta x^2} \tag{10}$$

This is still first-order in time, but is not restricted by the timestep constraint (although the timestep will still determine the accuracy). Defining:

$$\alpha \equiv k \frac{\Delta t}{\Delta x^2} \tag{11}$$

we can write this as:

$$-\alpha \phi_{i+1}^{n+1} + (1 + 2\alpha)\phi_i^{n+1} - \alpha \phi_{i-1}^{n+1} = \phi_i^n \tag{12}$$

This is a set of coupled algebraic equations. We can write this in matrix form. Using a cell-centered grid, spanning $[\mathrm{lo}, \mathrm{hi}]$, and Neumann BCs on the left:

$$\phi_{\mathrm{lo}-1} = \phi_{\mathrm{lo}} \tag{13}$$

the update for the leftmost cell is:

$$(1 + \alpha)\phi_{\mathrm{lo}}^{n+1} - \alpha \phi_{\mathrm{lo}+1}^{n+1} = \phi_{\mathrm{lo}}^n \tag{14}$$

If we choose Dirichlet BCs on the right ($\phi|_{x=x_l} = A$), then:

$$\phi_{\mathrm{hi}+1} = 2A - \phi_{\mathrm{hi}} \tag{15}$$

and the update for the rightmost cell is:

$$-\alpha \phi_{\mathrm{hi}-1}^{n+1} + (1 + 3\alpha)\phi_{\mathrm{hi}}^{n+1} = \phi_{\mathrm{hi}}^n + \alpha 2A \tag{16}$$

For all other interior cells, the stencil is unchanged. The resulting system can be written in matrix form and appears as a *tridiagonal* matrix.

$$\begin{pmatrix} 1+\alpha & -\alpha & & & & & \\ -\alpha & 1+2\alpha & -\alpha & & & & \\ & -\alpha & 1+2\alpha & -\alpha & & & \\ & & \ddots & \ddots & \ddots & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & -\alpha & 1+2\alpha & -\alpha \\ & & & & & -\alpha & 1+3\alpha \end{pmatrix} \begin{pmatrix} \phi_{\mathrm{lo}}^{n+1} \\ \phi_{\mathrm{lo}+1}^{n+1} \\ \phi_{\mathrm{lo}+2}^{n+1} \\ \vdots \\ \vdots \\ \phi_{\mathrm{hi}-1}^{n+1} \\ \phi_{\mathrm{hi}}^{n+1} \end{pmatrix} = \begin{pmatrix} \phi_{\mathrm{lo}}^n \\ \phi_{\mathrm{lo}+1}^n \\ \phi_{\mathrm{lo}+2}^n \\ \vdots \\ \vdots \\ \phi_{\mathrm{hi}-1}^n \\ \phi_{\mathrm{hi}}^n + \alpha 2A \end{pmatrix} \tag{17}$$

This can be solved by standard matrix operations, using a tridiagonal solvers (for example).

Exercise 1: Write a one-dimensional implicit diffusion solver for the domain $[0, 1]$ with Neumann boundary conditions at each end and $k = 1$. Your solver should use a tridiagonal solver and initialize a matrix like that above. Use a timestep close to the explicit step, a grid with N = 128 zones.

If we begin with a Gaussian, the resulting solution is also a Gaussian. Initialize using the following with $t = 0$:

$$\phi(x, t) = (\phi_2 - \phi_1)\sqrt{\frac{t_0}{t + t_0}} e^{-\frac{1}{4}(x - x_c)^2 / k(t + t_0)} + \phi_1 \tag{18}$$

with $t_0 = 0.001$, $\phi_1 = 1$, and $\phi_2 = 2$, and x_c is the coordinate of the center of the domain. Run until $t = 0.01$ and compare to the analytic solution above.

(Note: the solution for two-dimensions differs slightly)

4 Implicit multi-dimensional diffusion via multigrid

Consider a second-order accurate time discretization (this means that the RHS is centered in time), for the multi-dimensional diffusion equation:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{1}{2}\left(k\nabla^2\phi_i^n + k\nabla^2\phi_i^{n+1}\right) \tag{19}$$

This time-discretization is sometimes called *Crank-Nicolson*. Grouping all the $n+1$ terms on the left, we find:

$$\phi_i^{n+1} - \frac{\Delta t}{2}k\nabla^2\phi^{n+1} = \phi^n + \frac{\Delta t}{2}k\nabla^2\phi^n \tag{20}$$

This is in the form of a constant-coefficient Helmholtz equation,

$$(\alpha - \beta\nabla^2)\phi = f \tag{21}$$

with

$$\alpha = 1 \tag{22}$$

$$\beta = \frac{\Delta t}{2}k \tag{23}$$

$$f = \phi_i^n + \frac{\Delta t}{2}k\nabla^2\phi_i^n \tag{24}$$

This can be solved using multigrid techniques with a Helmholtz operator. The same boundary conditions described above apply here.

Note: when using multigrid, you do not need to actually construct the matrix. This is usually the most efficient way to implement diffusion in a multi-dimensional simulation code, especially when distributing the grid across parallel processors.

5 Going further

- *Non-constant conductivity*: for the case where $k = k(x)$, we discretize as:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{\{k\nabla\phi\}_{i+1/2} - \{k\nabla\phi\}_{i-1/2}}{\Delta x} \tag{25}$$

  Here we need the values of k at the interfaces, $k_{i-1/2}$ and $k_{i+1/2}$. We can get these from the cell-centered values in a variety of ways including straight-averaging:

$$k_{i+1/2} = \frac{1}{2}(k_i + k_{i+1}) \tag{26}$$

  or averaging the inverses:

$$\frac{1}{k_{i+1/2}} = \frac{1}{2}\left(\frac{1}{k_i} + \frac{1}{k_{i+1}}\right) \tag{27}$$

  The latter may be the right method for conduction, since it is behaves like resistance.

- *State-dependent transport coefficients*: many times the transport coefficents themselves depend on the quantity being diffused:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{1}{2}\left\{\nabla\cdot[k(\phi^n)\nabla\phi^n] + \nabla\cdot[k(\phi^{n+1})\nabla\phi^{n+1}]\right\} \tag{28}$$

(for example, with thermal diffusion, the conductivity can be temperature dependent). In this case, we can achieve second-order accuracy by doing a predictor-corrector. First we diffuse with the transport coefficients evaluated at the old time, giving a provisional state, $\phi^\star$:

$$\frac{\phi_i^\star - \phi_i^n}{\Delta t} = \frac{1}{2}\left\{\nabla\cdot[k(\phi^n)\nabla\phi^n] + \nabla\cdot[k(\phi^n)\nabla\phi^\star]\right\} \tag{29}$$

Then we redo the diffusion, evaluating k with $\phi^\star$ to center the righthand side in time, giving the new state, ϕ^{n+1}:

$$\frac{\phi_i^{n+1} - \phi_i^n}{\Delta t} = \frac{1}{2}\left\{\nabla\cdot[k(\phi^n)\nabla\phi^n] + \nabla\cdot[k(\phi^\star)\nabla\phi^{n+1}]\right\} \tag{30}$$

This is the approach used, for example, in [1].

- *Temperature diffusion in energy equation*: Often we find diffusion represented as one of many physical processes in a single equation. For example, consider the internal energy equation with both reactions and diffusion:

$$\rho\frac{\partial e}{\partial t} + \rho U\cdot\nabla e + p\nabla\cdot U = \nabla\cdot k\nabla T + \rho S \tag{31}$$

This can be solved via an explicit-implicit discretization. First the advection terms are computed as:

$$A = \rho U\cdot\nabla e + p\nabla\cdot U \tag{32}$$

Then the advective-diffusive part is solved implicitly. Expressing $e = e(\rho, T)$, and rewriting

$$\nabla T = (\nabla e - e_\rho\nabla\rho)/e_T \tag{33}$$

and then

$$\rho\frac{\partial e}{\partial t} = \nabla\cdot(k/e_T)\nabla e - \nabla\cdot(ke_\rho/e_T)\nabla\rho - A + \rho S \tag{34}$$

This is now a diffusion equation for e, which can be solved by the techniques described above. This is discussed, for example, in [1, 2].

References

[1] J. B. Bell, M. S. Day, C. A. Rendleman, S. E. Woosley, and M. A. Zingale. Adaptive low Mach number simulations of nuclear flame microphysics. *Journal of Computational Physics*, 195(2):677–694, 2004.

[2] C. M. Malone, A. Nonaka, A. S. Almgren, J. B. Bell, and M. Zingale. Multidimensional Modeling of Type I X-ray Bursts. I. Two-dimensional Convection Prior to the Outburst of a Pure ^{4}He Accretor. *ApJ*, 728:118, February 2011.