BBSI notes

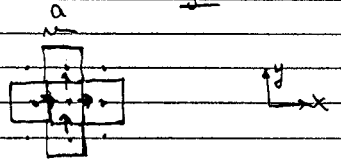
Partial Differential Equations (PDE's)

Real world applications: i) recovering oil from the ground
ii) modeling blood flow in the heart
iii) modeling flow (current) of ions through protein channels
⋮

The Continuity Equation

Fig 1

Illustrate in 2D



$\vec{j}$

arrows indicate flux through "faces"

For flow into (central) point (i,j): (x-flux = # particles / cm · sec)

$$a^2 \dot{\rho}_{ij} = -\left[j_x(i+\tfrac{1}{2}, j) - j_x(i-\tfrac{1}{2}, j) \right] a$$
$$-\left[j_y(i, j+\tfrac{1}{2}) - j_y(i, j-\tfrac{1}{2}) \right] a$$

thus:

$$\dot{\rho}_{ij} \underset{a\to 0}{=} -\left(\frac{\partial j_x}{\partial x}\right)_{ij} - \left(\frac{\partial j_y}{\partial y}\right)_{ij} \iff \boxed{\frac{\partial \rho(x,y,t)}{\partial t} = -\frac{\partial j_x}{\partial x}(x,y,t) - \frac{\partial j_y}{\partial y}(x,y,t)}$$

2D

In 3D:

$$\boxed{\frac{\partial \rho(x,y,z,t)}{\partial t}} = -\frac{\partial j_x}{\partial x}(x,y,z,t) - \frac{\partial j_y}{\partial y}(x,y,z,t) - \frac{\partial j_z}{\partial z}(x,y,z,t) = \boxed{-\vec{\nabla}\cdot\vec{j}(x,y,z,t)}$$

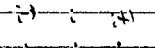
Application to Diffusion: Fick's Law: $\vec{j}(\vec{x},t) = -D\vec{\nabla}\rho(\vec{x},t)$

D ← diffusion constant

thus, in 3D: $\boxed{\frac{\partial \rho(\vec{x},t)}{\partial t} = D\nabla^2 \rho(\vec{x},t)}$

Solving PDE's via finite difference (real-space lattice) methods

Illustrate 1st w/ 1D Diffusion $\left[\frac{\partial \rho(x,t)}{\partial t} = D\frac{\partial^2}{\partial x^2}\rho(x,t)\right]$



$$j(i+\tfrac{1}{2}) \approx -\frac{(\rho_{i+1}-\rho_i)D}{a}$$

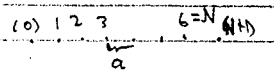
1D flux = #particles/sec

thus, for interior points:

$$\dot{\rho}_i = \frac{(\rho_{i+1}-\rho_i)D}{a^2} - \frac{(\rho_i-\rho_{i-1})D}{a^2} = \frac{D}{a^2}\left[\rho_{i+1} - 2\rho_i + \rho_{i-1}\right]$$

What about points at or next to the boundary?

For concreteness, consider absorbing b.c.'s ← "boundary conditions"



$\rho_0(t) = 0 = \rho_{N+1}(t)$

Thus, $\dot{\rho}_1 = \frac{D}{a^2}[\rho_2 - 2\rho_1]$; $\dot{\rho}_N = \frac{D}{a^2}[-2\rho_N + \rho_{N-1}]$

Put all this together:

$$\begin{pmatrix} \dot{\rho}_1 \\ \dot{\rho}_2 \\ \dot{\rho}_3 \\ \dot{\rho}_4 \\ \dot{\rho}_5 \\ \dot{\rho}_6 \end{pmatrix} = \frac{D}{a^2} \underbrace{\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix}}_{\underline{\underline{\Delta}}} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \end{pmatrix}$$

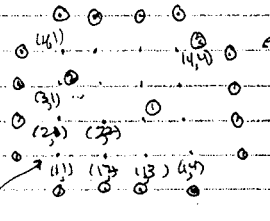
↑ Reduction to coupled ODE's (doable!)

Note: special techniques for Δ solutions:

$$\vec{\rho}(t) = \exp\left\{ \frac{Dt}{a^2} \underline{\underline{\Delta}} \right\} \vec{\rho}(0)$$

Comment on multi-d finite-diff. Eqs.:

In 2D:



For concreteness, take the concentration on the external boundary points as fixed, and for simplicity, $\rho_{\alpha\beta} = 0$ ← Bounding surfaces are sinks

external boundary points

Then, for completely internal points like ①

$$\frac{a^2}{D}\dot{\rho}_{i,j} = \left[\rho_{i+1,j} - 2\rho_{i,j} + \rho_{i-1,j}\right] + \left[\rho_{i,j+1} - 2\rho_{i,j} + \rho_{i,j-1}\right]$$

For a point on one edge, e.g. ②

$$\frac{a^2}{D}\dot{\rho}_{3,1} = \left[\rho_{3,2} - 2\rho_{3,1}\right] + \left[\rho_{4,1} - 2\rho_{3,1} + \rho_{2,1}\right]$$

For a point on one of the corners, e.g. ③

$$\frac{a^2}{D}\dot{\rho}_{4,4} = \left[-2\rho_{4,4} + \rho_{4,3}\right] + \left[-2\rho_{4,4} + \rho_{3,4}\right]$$

Drift-Diffusion: when there is a systematic (external) force $\vec{F}(\vec{x})$, there is a drift flux:

$$\vec{j}_{drift} = \beta D\, \vec{F}(\vec{x})\, \rho(\vec{x})$$

β: $\left(\frac{1}{kT}\right)$

Total flux is then:

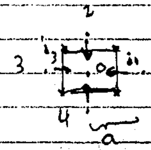
$$\vec{j} = \vec{j}_{diff} + \vec{j}_{drift} = -D\vec{\nabla}\rho + D\beta\vec{F}\rho = -D\left[\vec{\nabla}\rho - \beta\vec{F}\rho\right]$$

Then:

$$\frac{\partial\rho}{\partial t} = -\vec{\nabla}\cdot\vec{j} = D\vec{\nabla}\cdot\left[\vec{\nabla}\rho - \beta\vec{F}\rho\right]$$

Comment on the discretization procedure when $\vec{F}(\vec{x}) = -\vec{\nabla}V(\vec{x})$ (V: potential energy function)

Fig. 2



Now: $j_3 = -D\dfrac{[\rho_0 - \rho_3]}{a} - \beta D\dfrac{[V_0 - V_3]}{a}\dfrac{(\rho_0 + \rho_3)}{2}$

note direction of fluxes in Fig 2 — *same basic structure as diff. Eq. (wrong)*

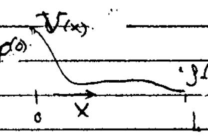
Completing this exercise ⇒ $\dot{\rho}_0 = j_1 + j_2 + j_3 + j_4$

The special case of steady state: $\vec{\nabla}\cdot\vec{j} = -\frac{\partial \rho}{\partial t} = 0$ (general)

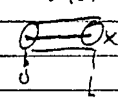
For drift-diffusion: $0 = \vec{\nabla}\cdot[\nabla\rho - \beta\vec{F}\rho]$, where $-\beta\vec{F} = +\beta(\vec{\nabla}V)$

Example:

1st study 1D process: $\rho = \rho(0)$, $\rho(L) = \rho_L$ ⇔



Physically: $\rightarrow F(x) = -\frac{\partial V}{\partial x}$



what is?: $\rho(x, t\to\infty) \equiv \rho(x)$; what is?: $j(x, t\to\infty) \equiv j$

$$0 = \frac{\partial}{\partial x}\left(\frac{\partial}{\partial x}\rho + \beta\frac{\partial V}{\partial x}\rho\right) \quad ; \quad \text{Let } \rho = e^{-\beta V} f$$

$$-\frac{j(x)}{D} = -\frac{j}{D} = \text{constant!}$$

$$-\frac{j}{D} = -\beta V' e^{-\beta V} f + e^{-\beta V} f' + \beta V' e^{-\beta V} f$$

$$f' = -\frac{j}{D} e^{\beta V} \Rightarrow f = -\frac{j}{D}\int_0^x e^{\beta V(x')}\,dx' + f_0$$

$$f_0 = \rho_0 e^{\beta V_0} \leftarrow V(x_0)$$

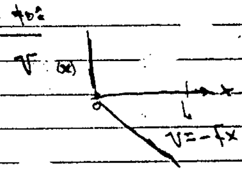
$$f(L) = \rho_L e^{\beta V_L} = -\frac{j}{D}\int_0^L e^{\beta V(x')}\,dx' + \rho_0 e^{\beta V_0}$$

Thus:

$$\frac{j}{D} = \frac{-\left(\rho_L e^{\beta V_L} - \rho_0 e^{\beta V_0}\right)}{\int_0^L e^{\beta V(x')}\,dx'}$$

Note: In Fig 3,
① ⇔ asymptotic slope of: $D\beta F\rho_0$
② ⇔ asymptotic slope of: $-D\beta|F|\rho_L$

finally, specialize to:



Note: $\int_0^L dx\, e^{-\beta F x}\, dx = \frac{1}{\beta F}\left(1 - e^{-\beta F L}\right)$

$$\Rightarrow \quad j = \frac{\left(\rho_0 - \rho_L e^{-\beta F L}\right) D\beta F}{\left(1 - e^{-\beta F L}\right)} =$$

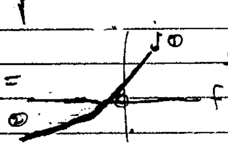


Fig.3

$\rho_0 > \rho_L$

In the case of 2,3... Dim., use "Relaxation" principles. In 2D, for Diff-Diff Eq:

$$0 = [j_1 + j_2 + j_3 + j_4]/D \Rightarrow$$

$$0 = (\rho_0 - \rho_1) + \frac{\beta}{2}(v_0 - v_1)\frac{(\rho_0 + \rho_1)}{2} + (\rho_0 - \rho_2) + \frac{\beta}{2}(v_0 - v_2)\frac{(\rho_0 + \rho_2)}{2}$$
$$(\rho_0 - \rho_3) + \frac{\beta}{2}(v_0 - v_3)\frac{(\rho_0 + \rho_3)}{2} + (\rho_0 - \rho_4) + \frac{\beta}{2}(v_0 - v_4)\frac{(\rho_0 + \rho_4)}{2}$$

$$0 = 4\rho_0 - \sum_{k=1}^{4}\rho_k\left(1 - \frac{\beta}{2}(v_0 - v_k)\right) + \rho_0\frac{\beta}{2}\sum_{k=1}^{4}(v_0 - v_k)$$

$$\rho_0 = \frac{\sum_{k=1}^{4}\rho_k\left[1 + \frac{\beta}{2}(v_k - v_0)\right]}{4 + \frac{\beta}{2}\sum_{k=1}^{4}(v_0 - v_k)}$$

Relaxation method

In practice: $\rho_{new} = (1-\omega)\rho_{old} + \omega\bar{\rho}$ $\quad 0 < \omega \leq 2$ ← cycle around lattice of "live" points; update them one by one.

Final variant on the PDE/relaxation theme: Poisson Eq. $\quad \nabla^2\phi = -4\pi\rho$ ← in vacuum (using Gaussian/CGS units)

$\phi(\vec{x})$ = electrostatic potential

Consider the discretized 2-D version:

$$\frac{1}{a^2}(\phi_1 - 2\phi_0 + \phi_3) + \frac{1}{a^2}(\phi_2 - 2\phi_0 + \phi_4) = -4\pi\rho_0$$

$$\frac{4\pi\rho_0 a^2 + \sum_{k=1}^{4}\phi_k}{4} = \phi_0$$

from Mathcad manual:

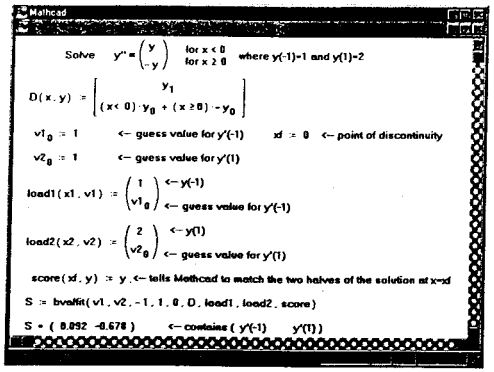


Figure 16-8: Using bvalfit to match solutions in the middle of the integration interval.

Partial differential equations

A second type of boundary value problem arises when you are solving a partial differential equation. Rather than fixing the value of a solution at two points as was done in the previous section, we now fix the solution at a whole continuum of points representing some boundary.

Two partial differential equations that arise often in the analysis of physical systems are Poisson's equation:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y)$$

and its homogeneous form, Laplace's equation.

Mathcad has two functions for solving these equations over a square boundary. You should use the *relax* function if you know the value taken by the unknown function $u(x, y)$ on all four sides of a square region.

If $u(x, y)$ is zero on all four sides of the square, you can use *multigrid* function instead. This function will often solve the problem faster than *relax*. Note that if the boundary condition is the same on all four sides, you can simply transform the equation to an equivalent one in which the value is zero on all four sides.

The *relax* function returns a square matrix in which:

- An element's location in the matrix corresponds to its location within the square region, and
- Its value approximates the value of the solution at that point.

This function uses the relaxation method to converge to the solution. Poisson's equation on a square domain is represented by:

$$a_{j,k}u_{j+1,k} + b_{j,k}u_{j-1,k} + c_{j,k}u_{j,k+1} + d_{j,k}u_{j,k-1} + e_{j,k}u_{j,k} = f_{j,k}$$

The arguments taken by these functions are shown below:

Pro relax(**a**, **b**, **c**, **d**, **e**, **f**, **u**, *rjac*)

a, **b**, **c**, **d**, **e** = Square matrices all of the same size containing coefficients of the above equation.

f = Square matrix containing the source term at each point in the region in which the solution is sought.

u = Square matrix containing boundary values along the edges of the region and initial guesses for the solution inside the region.

rjac = Spectral radius of the Jacobi iteration. This number between 0 and 1 controls the convergence of the relaxation algorithm. Its optimal value depends on the details of your problem.

If the boundary condition is zero on all four sides of the square integration domain, use the *multigrid* function instead. An example is shown in Figure 16-9. The same problem solved with the *relax* function instead is shown in Figure 16-10.

Pro multigrid(**M**, *ncycle*)

M = $(1 + 2^n)$ row square matrix whose elements correspond to the source term at the corresponding point in the square domain.

ncycle = The number of cycles at each level of the *multigrid* iteration. A value of 2 will generally give a good approximation of the solution.
