Mat1062: Introductory Numerical Methods for PDE

Mary Pugh

January 8, 2009

1 Ownership

These notes are the joint property of Rob Almgren and Mary Pugh.

2 Numerical methods for the diffusion equation

Now let us construct a numerical model for the diffusion equation

$$u_t = Du_{xx} = (Du_x)_x.$$

We shall first talk about the pure initial value problem: find $u(x,t)$ defined for all $-\infty < x < \infty$ and for $t \geq 0$, so that $u_t = Du_{xx}$ on this region, and so that $u(x,0) = u_0(x)$ where $u_0(x)$ is given initial data. We shall discretize first in space, then in time.

2.1 Space discretization

Thus to begin, let us choose a *grid spacing* h and introduce grid points $x_j = jh$ where $j \in \mathbb{Z}$. We want to approximate the PDE's solution, $u(x,t)$, on $\mathbb{R} \times [0,\infty)$ by an infinite collection of functions, $\{U_j(t)\}$, on $[0,\infty)$ such that $U_j(t)$ will be a good approximation for $u(x_j,t)$ as $h \to 0$.

The $U_j(t)$ will be solutions of an infinite collection of ODEs; the ODEs are determined by the PDE (the diffusion equation, in this case). To find the system of ODEs, we need to approximate $(Du_x)_x$ by a *difference formula.* We do this by computing first derivatives twice, mimicking the two first derivatives (gradient and divergence) which we used to construct the diffusion equation in the January 6 notes.

First, let us note that if the $U_j(t)$ are a good approximation to the grid values of a smooth function $u(x,t)$, $(U_j(t) \approx u(jh,t))$ then they can be used to approximate the first x-derivative of $u(x,t)$:

$$u_x((j+\tfrac{1}{2})h,t) \approx \frac{1}{h}\big(u(x_{j+1},t) - u(x_j,t)\big) \approx \frac{1}{h}\big(U_{j+1}(t) - U_j(t)\big).$$

This approximates the first derivative at the "half-grid point" $j+\frac{1}{2}$, centered between the points j and $j+1$ at which we evaluate u itself. Next, we take the second derivative by differencing this expression:

$$\begin{aligned} Du_{xx}(x_j,t) = (Du_x)_x(x_j,t) &\approx \frac{1}{h}\big(\, Du_x((j+\tfrac{1}{2})h,t) \; - \; Du_x((j-\tfrac{1}{2})h,t)\,\big) \\ &\approx \frac{1}{h}\left(\, D\frac{U_{j+1}(t) - U_j(t)}{h} \; - \; D\frac{U_j(t) - U_{j-1}(t)}{h}\,\right) \\ &= \frac{D}{h^2}\big(\, U_{j+1}(t) \; - \; 2U_j(t) \; + \; U_{j-1}(t)\,\big). \end{aligned}$$

In this way, we approximate the PDE $u_t = Du_{xx}$ which holds on $\mathbb{R} \times (0,\infty)$ with the infinite system of ODEs $d/dt\, U_j = D(U_{j+1} - 2U_j + U_{j-1})/h^2$ which holds for $\mathbb{Z} \times (0,\infty)$.

Above, there were a lot of $\approx$ signs. We now try to understand some of them; specifically, we show that if $u(x,t)$ is "sufficiently smooth" in x then the finite difference approximation

$$u_{xx}(x_j,t) \approx \frac{1}{h^2}\left(u(x_{j+1},t) - 2u(x_j,t) + u(x_{j-1},t)\right) \tag{1}$$

means

$$\left| u_{xx}(x_j,t) - \frac{1}{h^2}\left(u(x_{j+1},t) - 2u(x_j,t) + u(x_{j-1},t)\right) \right| \le Ch^2 \tag{2}$$

for some constant C. That is,

$$\frac{1}{h^2}\left(u(x_{j+1},t) - 2u(x_j,t) + u(x_{j-1},t)\right) \qquad \text{is } \mathcal{O}(h^2) \text{ close to} \qquad u_{xx}(x_j,t).$$

Note: the definition of what it means for one thing to be $\mathcal{O}(h^2)$ close to something else is "Given $\delta > 0$ there is a C so that the absolute value of their difference is less than or equal to h^2 for all $h \le \delta$." The constant C is allowed to depend on δ. And in the above example, it will also depend on time and on x derivatives of u.

Let us assume that $u(x,t)$ is "smooth enough" in x and do some Taylor expansions:

$$u(x_j+h,t) = u(x_j,t)+hu_x(x_j,t)+\frac{h^2}{2}u_{xx}(x_j,t)+\frac{h^3}{6}u_{xxx}(x_j,t)+\frac{h^4}{24}u_{xxxx}(\xi_2,t)$$

where $\xi_2 \in (x_j, x_j+h)$ (by Taylor's Theorem). Similarly,

$$u(x_j-h,t) = u(x_j,t)-hu_x(x_j,t)+\frac{h^2}{2}u_{xx}(x_j,t)-\frac{h^3}{6}u_{xxx}(x_j,t)+\frac{h^4}{24}u_{xxxx}(\xi_1,t)$$

where $\xi_1 \in (x_j-h, x_j)$. Combining these,

$$\frac{1}{h^2}\left(u(x_j+h,t)-2u(x_j,t)+u(x_j-h,t)\right) \quad (3)$$

$$= u_{xx}(x_j,t)+\frac{h^2}{12}\frac{u_{xxxx}(\xi_1,t)+u_{xxxx}(\xi_2,t)}{2} \quad (4)$$

$$= u_{xx}(x_j,t)+\frac{h^2}{12}u_{xxxx}(\xi_3,t) \quad (5)$$

for some $\xi_3 \in (\xi_1, \xi_2)$ (by the Intermediate Value Theorem). As a result,

$$\left|u_{xx}(x_j,t)-\frac{1}{h^2}\left(u(x_{j+1},t)-2u(x_j,t)+u(x_{j-1},t)\right)\right| = \frac{h^2}{12}|u_{xxxx}(\xi_3,t)| \le Ch^2.$$

From the above, we see that we have the desired $\mathcal{O}(h^2)$ closeness if two things are true: 1) $u_{xxxx}(x,t)$ is continuous in x and 2) $u_{xxxx}(x,t)$ is uniformly bounded on $\mathbb{R}$. Since we would then take

$$C = \frac{1}{12}\max_{x\in\mathbb{R}} u(x,t).$$

We could have written down the second-order difference formula (1) directly, but by respecting the physics and constructing it via two first derivatives we see the proper way to handle nonconstant coefficients. Consider the general case

$$c(x,t,u)\, u_t = (\kappa(x,t,u)\, u_x)_x$$

If c and κ are not constant, then we will see that a good approximation is

$$\frac{1}{c}\left(\kappa u_x\right)_x\Big|_{x=x_j} \approx \frac{1}{c_j}\frac{1}{h}\left(\kappa_{j+\frac{1}{2}}\frac{U_{j+1}-U_j}{h} - \kappa_{j-\frac{1}{2}}\frac{U_j-U_{j-1}}{h}\right) \quad (6)$$

where c_j and $\kappa_{j\pm\frac{1}{2}}$ are approximations to c at x_j and to κ at $x_{j\pm 1/2}$, respectively. For example, $\kappa_{j+\frac{1}{2}} = (\kappa_j + \kappa_{j+1})/2$. A *bad* approximation would be

$$\textbf{BAD:} \qquad \frac{1}{c}(\kappa u_x)_x\Big|_{x=x_j} \;\approx\; \frac{\kappa_j}{c_j}\frac{1}{h^2}\big(U_{j+1} - 2U_j + U_{j-1}\big), \tag{7}$$

since we will see that it does not properly respect conservation.

2.2 Time discretization

Now that we have approximated the space derivatives, we have approximated the PDE on $\mathbb{R} \times (0, \infty)$ with a system of ODEs for $U_j(t)$. Returning to the case of constant c, κ, and hence D, we write this system as

$$\frac{dU_j}{dt}(t) \;=\; \frac{D}{h^2}\big(U_{j+1}(t) - 2U_j(t) + U_{j-1}(t)\big), \tag{8}$$

on the lattice $j = \ldots, -2, -1, 0, 1, 2, \ldots$ in $\mathbb{R}$.

Following our approach to space discretization, we choose a *time step* k, introduce *time levels* $t_n = nk$, for $n = 0, 1, \ldots$, and let u_j^n denote an approximation to the (ODE) solution $U_j(t)$ at the time level $t_n = nk$:

$$u_j^n \approx U_j(t_n) = U_j(nk)$$

Recalling that $U_j(t_n) \approx u(x_j, t_n)$ we see that u_j^n is an approximation to the (PDE) solution $u(x,t)$ at the space-time point $(x_j, t_n) = (jh, nk)$. In short:

$$u_j^n \approx U_j(nk) \approx u(jh, nk)$$

Note: the superscript ^{n} is a *label* rather than an *exponent*.

Once we have figured out how to generate u_j^n for all $n > 0$ from a collection of initial values u_j^0 we will have gone from $u(x,t)$ which is a function on $\mathbb{R} \times [0, \infty)$ to a collection of numbers $\{u_j^n\}$ which are located on the lattice $\{(jh, nk) \mid j \in \mathbb{Z}, n \in \mathbb{N}\}$. The hope is that if we do our job well then as $h \to 0$ and $k \to 0$ these numbers will become better and better approximations of $u(x_j, t_n)$.

At this point, we are concerned with the approximating solutions of the system of ODEs (8). And so we start by considering a single ODE $y_t = f(y)$. The simplest way to approximate a solution of $y_t = f(y)$ is the *forward Euler* approximation: set $y^{n+1} = y^n + k\, f(y^n)$. (That is, approximate the function $f(y(t))$ on $[t_n, t_{n+1}]$ by the constant value $f(y^n)$ and then integrate the ODE $y_t = f(y^n)$ to the time t_{n+1} using the initial data $y(t_n) = y^n$.) Applied to the system of ODEs (8), we get the approximation

$$u_j^{n+1} \;=\; u_j^n \;+\; \frac{Dk}{h^2}\left(u_{j+1}^n \;-\; 2u_j^n \;+\; u_{j-1}^n\right). \tag{9}$$

Every term on the right side of this equation involves only grid values at level n, and the result is an explicit formula for grid values at level $n+1$.

This formula is very easy to implement: At $t = 0$, set the initial grid value $u_j^0 = u_0(jh)$, where $u_0(x)$ is the given initial data. Then for each $n = 1, 2, \ldots$, evaluate all grid points u_j^n at level n by evaluating formula (9) in terms of the previous level $n-1$. Once level n is done, then we have all the information we need to continue on to level $n+1$; in this way we can proceed forward as far as we like. We hope that each computed value u_j^n will be reasonably close to the corresponding value $u(x_j, t_n)$ of the true solution $u(x,t)$, especially if h and k are chosen small enough; this desirable property is called *convergence*.

Before we study convergence, let us consider the properties of diffusion equation that we discussed in the January 6 notes, to see whether they are preserved by the discrete model (9). As a preliminary, let us note that the scheme may be written as

$$u_j^{n+1} \;=\; \lambda\, u_{j+1}^n \;+\; (1-2\lambda)\, u_j^n \;+\; \lambda\, u_{j-1}^n, \qquad \lambda = \frac{Dk}{h^2}. \tag{10}$$

Note that the parameter λ is dimensionless; it is roughly the grid time step divided by the diffusion time on one spatial grid cell.

Conservation If $a = Ah$, $b = Bh$, and $t = nk$ for integers A, B, and n then the analogue of $\int_a^b u(x,t)\,dx$ is the sum $S_n = \sum_{j=A}^{B} u_j^n h$. The change in this quantity across one time step is

$$\begin{aligned} \frac{S_{n+1} - S_n}{k} \;&=\; \frac{D}{k}\sum_{j=A}^{B}\left(u_j^{n+1} - u_j^n\right) h \;=\; D\sum_{j=A}^{B}\left(\frac{u_{j+1}^n - u_j^n}{h} \;-\; \frac{u_j^n - u_{j-1}^n}{h}\right) \\ &=\; D\frac{u_{B+1}^n - u_B^n}{h} \;-\; D\frac{u_A^n - u_{A-1}^n}{h} \end{aligned}$$

(in the second sum, change $j \mapsto j+1$, then the middle part all cancels). Because the two terms on the right-hand side are the discrete approximation of u_x at $x = a$ and $x = b$, we see that this spatial discretization reults in the discrete analogue of the conservation law

$$\frac{d}{dt}\int_a^b u(x,t)\,dx \;=\; \int_a^b Du_{xx}(x,t)\,dx \;=\; Du_x(x,t)\Big|_{x=a}^{b}.$$

whatever the value of h, k, and D. It's clear that this property is preserved even if c and κ are not constant, as long as the "good" discretization (6) is used. But it's not preserved if the "bad" discretization (7) is used. (Check both of these facts!)

Maximum principle We first note that the local maxima decrease and local minima increase. Assume $u_j^n > u_{j\pm1}^n$. Because $\lambda > 0$ it follows that

$$u_j^{n+1} = \lambda\, u_{j+1}^n + (1-2\lambda)\, u_j^n + \lambda\, u_{j-1}^n < \lambda\, u_j^n + (1-2\lambda)\, u_j^n + \lambda\, u_j^n = u_j^n.$$

And so $u_j^{n+1} < u_j^n$; the local maximum moves downwards. Similarly, if $u_j^n < u_{j\pm1}^n$ then $u_j^n < u_j^{n+1}$; the local minimum moves upwards.

In terms of global upper and lower bounds, this will hold for discrete model if and only if $\lambda \leq \frac{1}{2}$, for only then are all the weights in the expression (10) positive. Assume that $u^n l \leq M$ for all j. Then if $\lambda \leq 1/2$ the rule (10) yields

$$u_j^{n+1} = \lambda\, u_{j+1}^n + (1-2\lambda)\, u_j^n + \lambda\, u_{j-1}^n < \lambda\, M + (1-2\lambda)\, M + \lambda\, M = M.$$

Similarly, lower bounds will be respected. Also, you can check that the discrete version of "a local maximum moves down" by checking that if u_j^n is greater than its nearest neighbours u_{j+1}^n and u_{j-1}^n then this will force $u_j^{n+1} < u_j^n$. Similarly, the discrete version of "a local minimum moves up" holds.

If $\lambda > 1/2$ then you can find positive initial data that results in solutions that become negative a certain locations at certain later times, violating the maximum principle. For example, the initial data

$$u_j^0 = \begin{cases} 1 & \text{for } j = 0 \\ 0 & \text{otherwise} \end{cases}$$

results in

$$u_j^1 = \begin{cases} \lambda & \text{for } j = \pm 1 \\ 1-2\lambda & \text{for } j = 0 \\ 0 & \text{otherwise} \end{cases}$$

From this, it's clear that if $\lambda > 1/2$ then $u^0 \geq 0$ but $u^1 \geq 0$ is not greater than equal to zero, violating the maximum (in this case the minimum) principle.

The condition $\lambda \leq 1/2$ is a constraint on the size of the time step k:

$$\lambda = \frac{Dk}{h^2} \leq \frac{1}{2} \quad \Longrightarrow \quad 2Dk \leq h^2.$$

That is, the time step must be short enough to resolve the diffusion time on one grid cell.

Speed of Propagation Assume that $\lambda \leq 1/2$ so that the maximum principle is obeyed. Consider the initial data

$$u_0(x) = \begin{cases} 1 & \text{if } x \leq 0 \\ 0 & \text{otherwise} \end{cases}$$

This initial data has a "front" at $x = 0$. This initial data results in a solution that has $u(x,t) > 0$ for *all* $x \in \mathbb{R}$ if $t > 0$; the front at $x = 0$ propagates with infinite speed, reaching "$x = \infty$" instantaneously.

The discretization (10) will have finite speed of propagation:

$$u_1^1 = \lambda u_0^0 = \lambda > 0; \qquad u_j^1 = 0 \quad \forall j > 1,$$

corresponding to a front at $x = h$. Similarly,

$$u_2^2 = \lambda u_1^1 = \lambda^2 > 0; \qquad u_j^2 = 0 \quad \forall j > 2,$$

corresponding to a front at $x = 2h$. In general, after n timesteps there will be a front at $x = nh$ and $u_n^n = \lambda^n$, which is positive (but tiny).

Fix $\epsilon > 0$ and use ϵ to set the time-step k

$$k = \frac{\epsilon}{n} \qquad \text{for some } n \in \mathbb{N}$$

It takes n timesteps to reach time $T = \epsilon$, at which time the front will be at location nh.

$$\lambda = \frac{Dk}{h^2} = \frac{D\epsilon}{nh^2} \quad \Longrightarrow \quad nh = \frac{D\epsilon}{\lambda h}.$$

And so, at time $T = \epsilon$ the front will be at $x = D\epsilon/(\lambda h)$. Recall that D and ϵ are fixed. Since $\lambda \leq 1/2$ implies $1/\lambda \geq 2$, the location of the front will satisfy

$$x_f = nh = \frac{D\epsilon}{\lambda h} \geq 2\frac{D\epsilon}{h}.$$

This shows that the smaller h is, the further to the right the front will be at time $T = \epsilon$. We will see that this is consistent with the desired inifinite speed of propagation in the $h \to 0$ limit.