

Mat1062: Computational Methods for PDE

Mary Pugh

January 24, 2008

1 Ownership

These notes are built upon those of Rob Almgren who taught an analogous course in 2003. Whatever you learn of value from them is due to him. All mistakes and sources of confusion are to be blamed on me.

2 Rigorous convergence for heat equation

We have two purposes in doing this: First, it is interesting to know that it is possible to prove convergence rigorously, at least for a simple problem and under the assumption that all functions are very smooth. Second, the method of proof illustrates the fundamental interplay of consistency and stability: If you make only a small error on each step, and the errors don't amplify, then the final error is small.

We consider the heat equation

$$u_t = Du_{xx} \quad \text{on } -\infty < x < \infty, \quad t > 0,$$

with initial condition $u(x, 0) = u_0(x)$. Let $v(x, t)$ denote the true solution to this problem. (Please suspend your disbelief on why I'm calling the true solution v rather than u .)

We consider only very smooth initial data; specifically we suppose that the function $u_0(x)$ has globally bounded 4th derivative.

There is $M < \infty$ so that $|u_0^{(4)}(x)| \leq M$ for all $-\infty < x < \infty$.

The maximum principle then guarantees us that the 4th x -derivative of $v(x, t)$ is bounded for all $t > 0$:

$$|v^{(4)}(x, t)| = |v_{xxxx}(x, t)| \leq M \quad \text{for all } x \text{ and all } t \geq 0.$$

This is because v 's being a solution of $u_t = Du_{xx}$ implies that $v^{(4)}$ is also a solution of $u_t = Du_{xx}$. And it's bounded by its initial data $|v_0^{(4)}|$ by the maximum principle. Furthermore, since

$$v_{tt} = (v_t)_t = (Dv_{xx})_t = D(v_t)_{xx} = D(Dv_{xx})_{xx} = D^2v_{xxxx},$$

we also have the bound

$$|v_{tt}(x, t)| \leq MD^2 \quad \text{for all } x \text{ and all } t \geq 0.$$

Now introduce a grid with space step h and time step k , and write $v_j^n = v(jh, nk)$ for samples of the true continuous solution on the grid. As part of proving consistence, we used the local asymptotic expression

$$\frac{1}{k} \left(v_j^{n+1} - v_j^n \right) \sim v_t(jh, nk) + \frac{1}{2}kv_{tt}(jh, nk) + \mathcal{O}(k^2), \quad k \rightarrow 0.$$

This means that there is some $K > 0$ such that

$$\left| \frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) \right| \leq k \|v_{tt}\|_\infty$$

for all j, n, h and all $k < K$. (See the explanation why at the end of this section.) But since we have a global bound on $|v_{tt}|$, we can assert

$$\left| \frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) \right| \leq kMD^2 \quad (1)$$

for all j, n, h and all $k < K$. Rather than using asymptotic expansions, this statement can also be proved using the Mean Value Theorem of calculus. It is not a particularly surprising conclusion, but I wanted to state it rigorously so you know what is involved (the bound on the second t -derivative).

Similarly, using the bound on $|v_{xxxx}|$, we can show that there is a $H > 0$ so that

$$\left| \frac{1}{h^2} \left(v_{j-1}^n - 2v_j^n + v_{j+1}^n \right) - v_{xx}(jh, nk) \right| \leq h^2M \quad (2)$$

for all j, n, k and all $h < H$.

Now let us turn to our discrete computation. For any particular values of h and k , we define a set of numbers u_j^n for $-\infty < j < \infty$ and $n \geq 0$ by the rule

$$u_j^0 = u_0(jh), \quad \text{for } -\infty < j < \infty,$$

and

$$\begin{aligned} u_j^{n+1} &= u_j^n + \lambda(u_{j-1}^n - 2u_j^n + u_{j+1}^n) \\ &= (1 - 2\lambda)u_j^n + \lambda(u_{j-1}^n + u_{j+1}^n) \end{aligned}$$

for $-\infty < j < \infty$ and $n \geq 0$, where

$$\lambda = \frac{Dk}{h^2}.$$

This is the formula we implement in our computer program.

We are interested in whether the numbers u_j^n converge to the true solution $v(x, t)$. The rule that generated u_j^n depends on both h and k . So the first thing we have to do is specify a relationship between h and k , so that we have a single parameter tending to zero. For this explicit method, we choose h to be given in terms of k as

$$k = \lambda h^2/D \quad \implies \quad h = \sqrt{\frac{kD}{\lambda}} \quad (3)$$

where λ is fixed. For implicit methods, we can choose $k = \mu h$, with μ fixed. Since h^2 is much smaller than h when h is small, explicit methods take much smaller time steps, and hence take many more steps to calculate to the same time t .

With this choice, the whole set of numbers u_j^n are determined by only the single parameter k . Convergence then means the following thing. Pick any particular position and time (x, t) . For each value of k and hence of h , there will be a particular pair of indices $(j_*(k), n_*(k))$ so that the corresponding grid point (j_*h, n_*k) is the closest grid point to (x, t) . As $h, k \rightarrow 0$, this closest grid point will get closer and closer to (x, t) . We want to show that

$$u_{j_*(k)}^{n_*(k)} \longrightarrow v(x, t) \quad \text{as } h \rightarrow 0, \quad \text{where } j_*h \rightarrow x \text{ and } n_*k \rightarrow t.$$

As $k \rightarrow 0$, j_* and n_* will $\rightarrow \infty$, since you need more and more grid points to reach the same location in space and time.

This definition sounds pretty complicated. In practice, you generally simplify things by looking only at a restricted set of h and k . For example, if you want to look at the solution at a fixed time T , you choose the parameters so $T = mk$, where m is an integer; then you simply look at the m th slice v_j^m as j varies. As $h \rightarrow 0$, also $k \rightarrow 0$, and $m \rightarrow \infty$ for fixed T . Furthermore,

if you have an exact expression for the solution $u(x, t)$, then you evaluate that expression at all the points (jh, T) . That is, you vary the point (x, t) at which you evaluate u as h decreases. Formally, though, it is more correct to hold (x, t) fixed and look at a suitable sequence of grid values so that $(jh, nk) \rightarrow (x, t)$.

In fact, what we will show is more like the description in the last paragraph. We want to show that as $k \rightarrow 0$, $u_j^n \rightarrow v_j^n$, for each j and n ; the key point is that this convergence must be *uniform* over all j and over all n with $nk \leq T$ for some fixed T . By contrast, if we picked only a single (j, n) , then as $h, k \rightarrow 0$, the value u_j^n would converge to the single value $v(0, 0)$ regardless of how well the method was working.

We must prove this convergence without knowing the exact solution $v(x, t)$ (otherwise we would have no need to construct a numerical method). The amazing fact is that we can do this using only the fact that v satisfies the PDE, which gives *consistency*, and using the fact that the discrete scheme for u is *stable*.

First, we define the **truncation error**

$$\begin{aligned} \epsilon_j^n &= v_j^{n+1} - (1 - 2\lambda)v_j^n - \lambda(v_{j-1}^n + v_{j+1}^n) \\ &= k \left(\frac{1}{k} (v_j^{n+1} - v_j^n) - \frac{D}{h^2} (v_{j-1}^n - 2v_j^n + v_{j+1}^n) \right) \end{aligned}$$

This is the amount by which the true solution fails to satisfy the difference formula. By the analysis above, we know that

$$\left| \epsilon_j^n - k \left(v_t(jh, nk) - Dv_{xx}(jh, nk) \right) \right| \leq k^2 MD^2 + kh^2 MD$$

(we suppose that h and k are always small enough to use the above results). But since v satisfies the PDE $u_t = Du_{xx}$, this gives

$$|\epsilon_j^n| \leq k^2 MD^2 + kh^2 MD \quad (4)$$

Furthermore, since $k = \lambda h^2/D$, we may write this as

$$|\epsilon_j^n| \leq h^2 k (\lambda MD + MD) = h^2 k MD (\lambda + 1)$$

The important part is that the right side is asymptotically smaller than k : we must make an error per step that is smaller than the time we spend, otherwise we have no hope of the cumulative error being small.

We may turn around the definition of the truncation error and write

$$v_j^{n+1} = (1 - 2\lambda)v_j^n + \lambda(v_{j-1}^n + v_{j+1}^n) + \epsilon_j^n.$$

This is the difference formula we use to generate the u_j^n , plus a small correction. That is, ϵ_j^n is the correction we should add into the difference scheme at each step to get the exact correct solution. From the consistency analysis, we know that this correction is uniformly small.

Now let us define the actual error

$$e_j^n = u_j^n - v_j^n.$$

We have $e_j^0 = 0$ for all j , since we took exact initial data. And e_j^n satisfies

$$\begin{aligned} e_j^{n+1} &= u_j^{n+1} - v_j^{n+1} \\ &= (1 - 2\lambda)u_j^n + \lambda(u_{j-1}^n + u_{j+1}^n) \\ &\quad - (1 - 2\lambda)v_j^n - \lambda(v_{j-1}^n + v_{j+1}^n) - \epsilon_j^n \\ &= (1 - 2\lambda)e_j^n + \lambda e_{j-1}^n + \lambda e_{j+1}^n - \epsilon_j^n. \end{aligned}$$

Taking absolute values on each side and using the triangle inequality,

$$|e_j^{n+1}| \leq |1 - 2\lambda| |e_j^n| + |\lambda| |e_{j-1}^n| + |\lambda| |e_{j+1}^n| + |\epsilon_j^n|. \quad (5)$$

We introduce the globally maximum error:

$$E^n = \|e^n\|_\infty = \max_j |e_j^n|.$$

Taking maximums on each side in (5), we find

$$\begin{aligned} E^{n+1} &\leq \max_j \left[|1 - 2\lambda| |e_j^n| + |\lambda| |e_{j-1}^n| + |\lambda| |e_{j+1}^n| + |\epsilon_j^n| \right] \\ &\leq |1 - 2\lambda| \max_j |e_j^n| + |\lambda| \max_j |e_{j-1}^n| + |\lambda| \max_j |e_{j+1}^n| + \max_j |\epsilon_j^n| \\ &\leq (|1 - 2\lambda| + 2|\lambda|) E^n + \max_j |\epsilon_j^n| \end{aligned}$$

Now we note that (stability)

$$\text{If } \lambda \leq \frac{1}{2}, \text{ then } |1 - 2\lambda| + 2|\lambda| = 1.$$

If $\lambda > \frac{1}{2}$, then this is false, which does not by itself prove instability; but we know from explicit Fourier solutions that it is unstable if $\lambda > \frac{1}{2}$. Assuming from now on that $\lambda \leq \frac{1}{2}$, we have

$$E^{n+1} \leq E^n + \max_j |\epsilon_j^n|. \quad (6)$$

That is, the error at the previous step is *not amplified*, since E^n is not multiplied by a coefficient greater than 1. The error at later steps is simply the accumulation of errors introduced at earlier steps, and we now have enough information to show that the total accumulated error is small.

Indeed, we simply sum (6) over n (using $E^0 = 0$) to see

$$E^n \leq \sum_{\ell=0}^{n-1} \max_j |\epsilon_j^\ell| \leq \sum_{\ell=0}^{n-1} \widetilde{M} k h^2 = \widetilde{M} n k h^2 = \widetilde{M} t h^2$$

where $\widetilde{M} = MD(\lambda + 1)$ and $nk = t$. That is, the final maximum error is the product of (1) a coefficient \widetilde{M} that depends on derivatives of the initial data and λ and D , (2) the time t we're computing up to (natural, since we are describing the final error as an accumulation of small errors), and (3) h^2 , which goes to zero. Thus for a fixed $\lambda \leq \frac{1}{2}$, $E^n \rightarrow 0$ as $h \rightarrow 0$, and the discrete solution converges to the true solution of the PDE.

Let's summarize the reasoning one more time. We did two things. First, we checked *consistency*; by plugging some hypothetical exact solution into the difference formula, we computed that the truncation error was small (4). Hence, using the discrete formula introduces a small error at each time step.

Next, we checked *stability* (6). This told us that the error did not amplify from one step to the next, except by the accumulation of the truncation errors. Stability combined with consistency gave us convergence.

As a postscript, let us remember that this result is not quite as powerful as it may sound. It depends essentially on having very smooth initial data, so that the solution remains smooth and we get calculus results like (1,2). But the heat equation has very strong smoothing properties: initial data that is not smooth quickly becomes so. Finite difference methods often work just fine on such problems, even though this theory does not apply. Much more is known about this kind of problems than we have time for.

One final note: in the above, I was very careful about my constants. I kept track of the D s and the λ s and ended up with a \widetilde{M} in which I could

see how things depended on what. This is not something I'd normally do — the mathematician in me would usually do the following: with each subsequent inequality I'd keep using the same notation M for the constant but I'd comment that each subsequent M might be larger than the prior M . After all, all I need is some upper bound in the end. If, for some reason, I found that I really needed to keep track of how things depended on λ and D then I could redo things and be super-careful. But normally if all I want to do is prove convergence, I wouldn't do this.

Some might argue that this is the difference between a theoretician: "There is a constant such that..." and someone who actually implements the theory: "Hey wait a minute! Your constant is huge and the numerical scheme's really slow, as a result!" This exchange could certainly happen if I'd just proven that the running time of the scheme is CNM^2 where N is the number of intervals in space and M is the number of intervals in time.

2.1 Our friend $\mathcal{O}(k^2)$

I want to show why

$$\frac{1}{k} \left(v_j^{n+1} - v_j^n \right) \sim v_t(jh, nk) + \frac{1}{2} k v_{tt}(jh, nk) + \mathcal{O}(k^2), \quad k \rightarrow 0. \quad (7)$$

implies that there is some $K > 0$ such that

$$\left| \frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) \right| \leq k \|v_{tt}(jh, nk)\|_\infty$$

for all $k < K$.

The statement (7) can be rewritten as

$$\frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) - \frac{1}{2} k v_{tt}(jh, nk) \sim \mathcal{O}(k^2), \quad k \rightarrow 0.$$

By definition of $\mathcal{O}(k^2)$, this means that there is a $C > 0$ and a $K_0 > 0$ such that

$$\left| \frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) - \frac{1}{2} k v_{tt}(jh, nk) \right| \leq C k^2$$

for all $k < K_0$. By the triangle inequality, if A and B are numbers then $|A| = |A - B + B| \leq |A - B| + |B|$ hence $|A| - |B| \leq |A - B|$. Therefore we have

$$\left| \frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) \right| - \left| \frac{1}{2} k v_{tt}(jh, nk) \right| \leq C k^2$$

and thus

$$\left| \frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) \right| \leq \frac{1}{2} k |v_{tt}(jh, nk)| + C k^2 \leq \frac{1}{2} k \|v_{tt}\|_\infty + C k^2$$

Now, let $K_1 = \|v_{tt}\|_\infty / (2C)$. Then

$$k < K_1 \quad \implies \quad C k^2 < \frac{1}{2} k \|v_{tt}\|_\infty$$

It then follows that if $K := \min\{K_0, K_1\}$ then

$$\left| \frac{1}{k} \left(v_j^{n+1} - v_j^n \right) - v_t(jh, nk) \right| \leq k \|v_{tt}\|_\infty$$

for all $k < K$.