Numerical Analysis Seminar Spring 2016 Notes

These notes contain sample problems for various methods examined in the Spring 2106 Numerical Analysis Seminar organized by Professor Adam Larios.

Our first objective is to run the march*.m files through some tests. Think of these files as a kind of "Swiss army knife" of convection-diffusion problems. We make no claims as to efficiency – the code certainly could be sped up for special instances. Nor suitability – caveat emptor. Right now we just want to get correct answers (or understand why we don't!). In all these examples a suitable number of steps and nodes was determined by trial and error, the error being pretty spectacular with bad choices!

There are lots of user parameters that need to be inspected in order to set up a problem. Here are the parameters that should be set:

- marchRHS.m: This function file defines the right hand side of the Fourier transformed version for our problem. In general user must define the functions at the end of the file according as the true flags are set in marchTest.m: marchP(v2,v,t), marchQ(v2,v,t), marchB(t), marchD(x) and marchF(x,t).
- marchTest.m: This is the driver function file that runs everything. User must set the parameters gpFlag, gqFlag, gfFlag, gbFlag, marchMTD, numProfiles, numSteps, T, N, gL (variables with prefix g are global – get over it!). Also, user must define functions marchIC and, if defined, marchSoln.

Example 1. Consider the model problem

$$
\begin{aligned}
u_t &= Du_{xx} + f(x,t),\ 0 < x < L,\ 0 < t < T \\
u(x,0) &= g(x),\ 0 \le x \le L. \\
u(0,t) = B(t), & \qquad u(1,t) = B(t),\ t > 0.
\end{aligned}
$$

We will use the forcing term $f(x,t)$ to enable us to cook up solutions (a nice way to test our algorithms and programming, but not necessarily enlightening about the nature of the unvarnished problem). Our first test solution uses $D = 1$ and is given by

$$
\begin{aligned}
u(x,t) &= e^{-t}\sin x, \quad 0 < x < 2\pi,\ 0 < t < 2, \\
u_t &= -e^{-t}\sin x, \\
u_{xx} &= -e^{-t}\sin x.
\end{aligned}
$$

This test solution defines the rest of the problem:

$$
\begin{aligned}
u(0,t) &= u(2\pi,t) = 0, \\
u(x,0) &= \sin(x), \\
f(x,t) &= 0.
\end{aligned}
$$

So in this simple case a forcing term is not really needed. The resulting calculation using marchTest.m with N = 32, gL = 2π, T = 2, numProfiles =

5, numSteps = 50, mtd = 'ERK4', uFlag true and gpFlag, gqFlag, gbFlag and gfFlag set to false (no need for $f(x,t)$ here) yields an absolute L2 norm error ~3.7e-11 and infinity norm error of final profile ~9.3e-12.

Example 2. Repeat the model problem of Example 1 with $D = 1/2$, $L = 2\pi$, $T = 3$ and ignore the boundary conditions,

$$\begin{aligned}
u(x,t) &= e^{-t/2}\sin^2 x, \quad 0 < x < 2\pi,\ 0 < t < 3,\\
u_t &= -\frac{e^{-t/2}}{2}\sin^2 x,\\
Du_x &= De^{-t/2}\sin(2x)\\
Du_{xx} &= 2De^{-t/2}\cos(2x)\,.
\end{aligned}$$

This test solution defines the rest of the problem, but in this case we are going to use $f(x,t)$ with

$$\begin{aligned}
u(x,0) &= \sin^2(x)\,,\\
f(x,t) &= -\frac{e^{-t/2}}{2}\left(4D\cos(2x) + \sin^2(x)\right).
\end{aligned}$$

The resulting calculation using convectTest.m with N = 16, gL = 2π, T = 3, numProfiles = 5, numSteps = 25, mtd = 'ERK4', and using marchHS.m with pFlag, qFlag and bFlag set to false and fFlag, uFlag set to true, yields an L2 norm error ~1.3e-09 and infinity norm error of final profile ~4.9e-10.

Example 3. Repeat the model problem of Example 1 with $D = 1/4$, $gL = 5$, $T = 4$ and ignore the boundary conditions, but this time we don't have a solution to check, so we'll just plot pictures. We are going to test a pulse of the form

$$u(x,0) = \begin{cases} 0, & 0 \le x < 2\\ 1, & 2 \le x < 3\\ 0, & 3 \le x \le 5. \end{cases}$$

The resulting calculation using marchTest.m with N = 32, L = 5, T = 3, numProfiles = 5, numSteps = 50, mtd = 'ERK4', gpFlag, gqFlag, gfFlag, gbFlag and uFlag set to false will yield a picture of the profiles. So the result is that the pulse is nicely smoothed out with decreasing L^2-norm on the interval $[0, L]$.

Example 4. Ok, time for something a bit different. Let's try a viscous Burgers' equation of the form

$$\begin{aligned}
u_t &= Du_{xx} - uu_x,\ 0 < x < L,\ 0 < t < T\\
u(x,0) &= g(x),\ 0 \le x \le L.\\
u(0,t) = B(t), &\quad u(1,t) = B(t),\ t > 0.
\end{aligned}$$

with $L = 6$, $T = 4$ and $D = 1/4$. Except that we will not enforce a boundary condition, nor do we have a solution in hand. We impose the initial condition

$$u(x,0) = \frac{1}{\sqrt{0.2 \cdot 2\pi}} e^{-(x-3)^2/0.4}$$

The resulting calculation using marchTest.m with N = 32, gL = 6, T = 4, numProfiles = 5, numSteps = 50, mtd = 'ERK4', gqFlag, gfFlag and gbFlag set to false, gpFlag to true, marchP(v2,v,t) defined as -0.5*v2, will yield a picture of the profiles. So we can see that the initial bell curve is smeared out a bit and moving to the right.

Example 5. Let's run a check on our Burger solution of the preceding example by using a known solution as in Example 2. We'll use D = 1/4, N = 32, gL = 2π , T = 4, numProfiles = 5, numSteps = 50, mtd = 'ERK4', qFlag, fFlag and bFlag set to false, pFlag to true, marchP(v2,v,t) defined as -0.5*v2, Our known solution is

$$u(x,t) = t\sin(x)$$

which leads to rhs function

$$f(x,t) = \sin(x)\left(1 + \frac{1}{4}t + t^2\cos(x)\right).$$

The result is impressive: absolute errors on the order of 1e-14. So let's try a harmless looking variation:

$$u(x,t) = t\cos(x)$$

which leads to rhs function

$$f(x,t) = cos(x)\left(1 + \frac{1}{4}t - t^2\sin(x)\right).$$

The result is equally impressive: errors on the order of 1e-14, which increases confidence in the preceding example.

Example 6. Let's try something else a bit different, namely a Fisher's equation of the form

$$\begin{aligned} u_t &= Du_{xx} + ru(1-u),\ 0 < x < 6,\ 0 < t < 1 \\ u(x,0) &= g(x),\ 0 \le x \le L. \\ u(0,t) = B(t), &\quad u(1,t) = B(t),\ t > 0. \end{aligned}$$

Once again we will not enforce a boundary condition, nor do we have a solution in hand. As with Burgers' equation we impose the initial condition

$$u(x,0) = \frac{1}{\sqrt{0.2 \cdot 2\pi}} e^{-(x-3)^2/0.4}$$

The resulting calculation using marchTest with D = 1/4, r = 1, N = 64, L = 6, T = 4, numProfiles = 5, numSteps = 50, mtd = 'ERK4', and using marchTest.m with pFlag, fFlag and bFlag set to false, qFlag to true, marchQ(v2,v,t) defined as v - v2, fails. This is an interesting problem. Let's try to find a culprit. At one extreme, we can set $D = 0$ and essentially solve an ode. Try it and we get a solution that is finite but definitely growing and expanding. Now reset $D = 1/4$ but take $N = 32$. Surprise: we get a reasonable solution which is growing fairly rapidly. Evidently more is not always better (think CFL.)

Example 7. Let's run a check on our Fisher solution of the preceding example by using a known solution from Example 5. We'll use D = 1/4, r = 1, N = 32, L = 2π, T = 4, numProfiles = 5, numSteps = 50, mtd = 'ERK4', pFlag, fFlag and bFlag set to false, qFlag to true, marchQ(v2,v,t) defined as v - v2, and the solution

$$u(x,t) = t\sin(x),$$

which leads to rhs function

$$f(x,t) = \sin(x)\left(t^2\sin(x) - \frac{3t}{4} + 1\right).$$

We get pretty good results with errors in the order of 1e-7. Let's improve it a bit by increasing numSteps to 100. Doing so only decreases the error to about 1e-8 – still pretty good.

Here's a variation where our known solution is simple, but a little less trig friendly:

$$u(x,t) = tx(2\pi - x)$$

which leads to rhs function

$$f(x,t) = \frac{1}{2}\left(t + 4\pi(1-t)x + \left(8\pi^2t^2 + 2t - 2\right)x^2 - 8\pi t^2x^3 + 2t^2x^4\right).$$

The result is not very good: relative infinity error of about 0.12. Try it with numSteps = 100 and virtually no improvement. So let's try this: replace the first x by x^2. This gives solution

$$u(x,t) = tx^2(2\pi - x)$$

and rhs function

$$f(x,t) = \frac{1}{2}\left(-2\pi t + 3tx + 4\pi(1-t)x^2 + 2(t-1)x^3 + 8\pi^2t^2x^4 - 8\pi t^2x^5 + 2t^2x^6\right).$$

This time numSteps = 50 simply fails. However, there is a surprise at numSteps = 100: errors are better: relative infinity norm of about 0.07. And yet doubling the number of steps gives virtually no improvement. For that matter, the error norms are pretty dismal, in contrast to earlier examples with larger step sizes. What's going on here?

The fact that there was no improvement with decreased time step tells us that space step dominates the error, so the only path to improvement is a smaller space step and then much smaller time step. (Try it by doubling space steps N and quadrupling time steps numSteps, and this time compare both infinity and L2 norms of error. Something curious there as well.) Examination of the graphs strongly suggests that the problem is at the endpoints. Now consider our first variation test function: if you assume it is periodic and defined by its values in the interval $[0, 2\pi]$, then it is certainly continuous everywhere, but is not smooth at $x = 0$. Similarly, the second variation is smooth everywhere, but not analytic (expandible in a power series at every point, which implies infinitely many derivatives.) However, our first test function is periodic and analytic. Moral: when using this spectral method, approximating continuous solutions may be ok, but smooth solutions are better and analytic solution are best.

Example 8. OK, now we move in a new direction with this example; specifically, we are going to use this example to text variable step methods. We would like something that puts stiff problems to the test. Let's start with a solution:

$$v_{(x)}(t) = \sin^2(x)\, e^{\lambda t} = u(x,t).$$

As a definition of $v_{(x)}(t)$ think of this formula as defining functions of time with x as a parameter, $0 \le x \le 2\pi$, whereas as a definition of $u(x,t)$ think of the formula as defining a function of space x and time t. Then $v_{(x)}$ satisfies the IVP

$$\begin{aligned} \frac{dv_{(x)}}{dt} &= \lambda v_{(x)},\ t > 0, \\ v_{(x)}(0) &= \sin^2(x). \end{aligned}$$

This is a parametrized family of the model problem $v' = \lambda v$ that is important for determining the linear stability domain of various methods. Let's turn this problem into something a bit more relevant to the discussion of the seminar, namely PDEs. We leave it to the reader to check that $u(x,t)$ satisfies the IVP-BVP problem

$$\begin{aligned} u_t &= Du_{xx} + (\lambda + 4D)\, u + F(x,t),\ t > 0,\ 0 < x < 2\pi, \\ u(x,0) &= \sin^2(x), 0 \le x \le 2\pi, \\ u(t,0) &= 0 = u(t,2\pi), t > 0, \end{aligned}$$

where the forcing term is given by

$$F(x,t) = -2De^{\lambda t}$$

and $\lambda < 0$, $D > 0$ are fixed parameters. The difficulties of the model problem for linear stability domains is embedded as a family of such problems with parameter x. Skeptical? Take $D = 0$ and see what you get. So the total effect is that diffusion acts as a mitigating factor for the model problem.

Example 9. This example is a fairly complex Burgers equation that incorporates some features of the linear stability domain model. Let's start with the solution:

$$u(x,t) = \sin(x)\, e^{t\cos(x)},\ 0 \leq x \leq \pi,\ t \geq 0.$$

Next the PDE in question, which we choose to be a viscid Burgers equation with transport term added on

$$u_t = Du_{xx} - uu_x - G(x,t)\, u$$

where the time and space dependent coefficient is

$$G(x,t) = D + \cos(x)\left(e^{t\cos(x)} - 3Dt - 1\right) + \sin^2(x)\left(t^2 D - te^{t\cos(x)}\right).$$

Amazingly, our solution satisfies this PDE along with initial/boundary conditions

$$u(x,0) = \sin(x),\ 0 \leq x \leq \pi,$$
$$u(0,t) = u(\pi,t) = 0,\ t > 0.$$

Example 10. Let's put the solution of the previous example

$$u(x,t) = \sin(x)\, e^{t\cos(x)},\ 0 \leq x \leq \pi,\ t \geq 0.$$

into a simpler diffusion equation with a transport term, namely

$$u_t = Du_{xx} + c(x,t)\, u$$

where the time and space dependent coefficient is

$$c(x,t) = \cos(x) + D\left(1 + 3t\cos(x) - t^2\sin^2(x)\right).$$

Our solution satisfies this PDE along with initial/boundary conditions

$$u(x,0) = \sin(x),\ 0 \leq x \leq 2\pi$$
$$u(0,t) = u(\pi,t) = 0,\ t > 0.$$