**MTH 453/MTH 553**

**Numerical Solution of Partial Differential Equations**

**Homework 3**

1. (**453 and 553**) Consider Laplace's equation posed on the unit square with Dirichlet boundary conditions:

$$\nabla^2 u = 0 \text{ in } \Omega = (0,1) \times (0,1) \tag{0.1}$$

$$u = g \text{ on } \partial\Omega \tag{0.2}$$

where $g$ is defined as

$$g(x,y) = \begin{cases} \sin(\pi x), & \text{if } y = 1 \text{ and } 0 \le x \le 1 \\ 0 & \text{otherwise} \end{cases} \tag{0.3}$$

(a) Verify that the solution to this boundary value problem is

$$u(x,y) = \frac{\sin(\pi x)\sinh(\pi y)}{\sinh(\pi)} \tag{0.4}$$

where we recall that the hyperbolic sine function is defined as $\sinh(z) = (\mathrm{e}^z - \mathrm{e}^{-z})/2$.

(b) Download the code `lap2d_jacobi.m`. This code implements the Jacobi method on the problem described above using a grid with $(m+2) \times (m+2)$ gridpoints in which the unknowns constitute an $m \times m$ mesh with $h = 1/(m+1)$ (since the boundary values are known). It produces plots of the analytical solution, the solution obtained by directly solving the discrete matrix system, and the solution obtained through Jacobi iteration of the matrix system. As output parameters, it computes the error between the iterative and exact solutions, as well as the error between the iterative and direct solutions.

Using this code as a template, you should implement the following in MATLAB:

(c) In a single window, plot the convergence history (the error versus the number of iterations). Do this for $m = 7$ ($h = 1/8$) and also for $m = 15$ ($h = 1/16$). The error should be relative to the analytical (exact) solution. Use logarithmic (base 2 preferably) scaling for the $y$-axis. See the sample file `plothistory.m`.

(d) We can see in the plots from the previous question that after a certain number of iterations the error stops improving. Why isn't the error going to zero? What type of error is left? What is the order of this error with respect to $h$? To help understand what is going on, consider computing the error between the exact and direct solutions, or plotting the error between the direct and iterative solutions versus the iteration number.

Again, use a logarithmic scale (base 2) for the plot, and recall that *machine epsilon* for double precision is $2^{-52}$.

(e) You should be able to estimate the "left-over error" described in the previous problem as a function of $h$ (careful not to round too much!). Implement a stopping criteria in the code by changing the error tolerance to be this amount (why is this not reasonable to do in general?!). On your own, re-do the plots above and check whether you have done this correctly. Then run your code for each of the following: $m = \{3, 7, 15, 31, 63\}$. On the same graph, make a plot of $k$ (the number of iterations required to satisfy your stopping criteria) versus $m$. Compare your results to the theoretical predicitions. It may be useful to also plot $k/log_2(m)$ versus $m$ with logarithmic scaling on both axes (use `axis('equal')` as well).

(f) Do a grid-refinement analysis. That is, using $m = \{3, 7, 15, 31, 63\}$, plot the final error versus $h$ (using logarithmic scaling on both axes and `axis('equal')`) to verify that the global truncation error is $O(h^2)$.

2. Consider the initial value problem

$$\begin{aligned} u_t - u_x &= 0, \\ u(x, 0) &= \sin(\omega x), \end{aligned}$$

where $\omega$ is a parameter. The exact solution to this problem is

$$u(x, t) = \sin(\omega(t + x)).$$

Let us solve this problem on the interval $[-\pi, \pi]$ with periodic boundary conditions, i.e.,

$$u(t, -\pi) = u(t, \pi), \ t > 0.$$

We perform a discretization of the spatial interval $[-\pi, \pi]$ with spatial step size $h$ and we use a temporal step size $k$ on the time interval [0,1]. Let $m + 1 = \frac{2\pi}{h}$. We want to calculate approximate solution values $u_j^n$ for $j = 1, \ldots, m + 1$. The periodic boundary conditions give

$$u_0^n = u_{m+1}^n, \ \forall n$$

It is helpful to imagine the spatial mesh points sitting on a ring, rather than on a straight line, with $x_{m+1}$ being identified with $x_0$. Thus, when $j = m$, then $u_{j+1} = u_0$ and for $j = 0, u_{j-1} = u_m$. (This gets further complicated in MATLAB as zero index is not allowed, so we add one to each index.)

(a) (**Both 453 and 553**) Download the code `leapfrog.m`. This code implements the Leapfrog (midpoint) method (CC), given in (10.13) in the text, to obtain a numerical solution to the given problem.

(b) (**Both 453 and 553**) Modify this code to create a new file that implements the forward in time-backward in space (FB) finite difference scheme given in (10.23) to solve the given problem.

(c) (**553 only**) Create a new file that implements the forward in time-centered in space (FC) finite difference scheme given in (10.5) to solve the given problem.

(d) Calculate the maximum errors at $t = 1$ for all three (**453** only two ) schemes using the values of $\omega$, $m$ and $\nu$ $(= k/h)$ given in the table below and fill in the missing entries in the table

| $\omega$ | $m+1$ | $\nu$ | Error in FB | Error in FC | Error in CC |
|---|---|---|---|---|---|
| 2 | 20 | 0.5 | 0.2473 | 0.3304 | 0.0788 |
| 2 | 200 | 0.5 | ? | ? | ? |
| 2 | 2000 | 0.5 | ? | ? | ? |

What order of accuracy does each method seem to exhibit?

(e) Fix $m + 1 = 20$. What is the effect on the error of each method when $\omega$ is increased or decreased?

(f) Fix $m + 1 = 20$. What is the effect on the error of each method when $\nu$ is increased or decreased? In particular, try $\nu = 1$ and $\nu > 1$.

3. (**553 only**) Consider the advection equation

$$u_t + au_x = 0,$$

The forward in time-centered in space (let's call this FC) scheme is consistent but unconditionally unstable. The *Lax-Friedrichs* scheme is a variation of the FC scheme and is given as

$$U_j^{n+1} = \frac{1}{2}\left(U_{j-1}^n + U_{j+1}^n\right) - \frac{\nu}{2}\left(U_{j+1}^n - U_{j-1}^n\right),$$

where $\nu = \frac{ak}{h}$ is the *Courant number*.

(a) Show that the Lax-Friedrichs scheme is consistent and find the order of accuracy.

(b) Show that the Lax-Friedrichs scheme is stable, provided that the CFL condition holds, i.e., $|\nu| \leq 1$.

4. (**BONUS**) Suppose that we use the *red black ordering* to order the unknowns in the two-dimensional finite difference system represented by the 5-point stencil for Poisson's problem. This ordering is significant because all four neighbors of a red point are black points, and vice versa, and leads to a matrix structure as shown in equation (3.13). What is the matrix $H$ for the grid shown in Figure 3.2(b)?