Numerical Methods II, Courant Institute, Spring 2012
http://www.math.nyu.edu/faculty/goodman/teaching/NumMethII/index.html
Always check the class bboard on the blackboard site from home.nyu.edu (click on academics, then on Numerical Methods II) before doing any work on the assignment.

Assignment 3, due February 23

Corrections: (none yet)

1. Suppose A is an $d \times d$ matrix. Show that the following are equivalent:

   - (*Power boundedness*) There is a C independent of n so that $\|A^n\| \leq C$ for all $n \geq 0$.
   - (*Root condition*) If λ is an eigenvalue of A, then either (1) $|\lambda| < 1$, or (2) $|\lambda| = 1$ and this λ has no Jordan structure. That condition is equivalent to the statement that if $(A - \lambda I)^k v = 0$ for some $v \in \mathbb{C}^d$ and $k > 0$, then $(A - \lambda I)v = 0$. Informally, we say eigenvalues must be in the unit disk, and eigenvalues on the boundary must be simple. Simple here does not mean that the eigenspace is one dimensional, but that there are no non-trivial Jordan blocks.

   This "if and only if" has an easy direction and a hard direction. The easy direction is that if the root condition is not satisfied then A is not power bounded. We did that in class. To prove the hard direction, you can prove a harder theorem

   - If A satisfies the root condition, then there is a norm $\|x\|$ so that $\|Ax\| \leq \|x\|$ for all x. This is a *contraction norm* for A, a norm in which A is a contraction.

   You can prove this in a sequence of lemmas

   (a) If $\|x\|_1$ is a contraction norm for matrix A, then $\|x\|_2 = \|Px\|_1$ is a contraction norm for $B = PAP^{-1}$ (or is it $B = P^{-1}AP$?).

   (b) Suppose $\|x_k\|_k$ is a contraction norm for J_k, and $x = (x_1, \ldots, x_l)$, with $x_k \in \mathbb{C}^{d_k}$ and $d = d_1 + \cdots + d_l$. Suppose that $y = Bx$ means $y_k = J_k x_k$ for $k = 1 \ldots, l$. (This is an indirect way of saying that B is block diagonal with the matrices J_k as the diagonal blocks.) Then $\|x\| = \left(\|x_1\|_1 + \cdots + \|x_l\|_l^2\right)^{1/2}$ is a contraction norm for B.

   (c) If J is a Jordan block with $|\lambda| = 1$, then J is diagonal and $\|x\| = (x^* x)^{1/2}$ is a contraction norm for J. (I write x^* instead of x^t because x might be complex.)

(d) If $|\lambda| < 1$, then the matrix

$$M = \sum_{n=0}^{\infty} (J^n)^* J^n$$

is finite, and $\|x\| = (x^* M x)^{1/2}$ is a contraction norm for J.

2. Use the result of the previous problem to prove convergence of linear multistep methods. More precisely, suppose the characteristic polynomial of the multistep method satisfies the root condition. Use a norm in which the companion matrix is a contraction. Let $X_n = (x_n, \ldots, x_{n-p})$. Then $X_{n+1} = \widetilde{A} X_n + \Delta t F(X_n)$, where $\widetilde{A}$ is d "copies" of A, one for each component of x. You will be able to show that if $Y_{n+1} \widetilde{A} Y_n + \Delta t F(Y_n) + \Delta t R_n$, then

$$\|Y_n - X_n\|_* \leq C(t_n) \|X_0 - Y_0\|_* + \Delta t \sum_{k \leq n} C(t_n - t_k) \|R_k]\|_* ,$$

where $\|\cdot\|_*$ is the contraction norm for $\widetilde{A}$. Use this to conclude that if the method has formal order of accuracy q and is stable by the root condition criterion, then it actually is accurate with order q provided the initial steps are done accurately enough. (*Part of the purpose of this problem is to have you write a consistency/stability argument from beginning to end. It's one of the most important theoretical things in the class.*)

3. This exercise does the stability of the discrete Laplace operator in l^∞ norm. It is a discrete version of *maximum principle* and *comparison principle* arguments that are common in the study of elliptic PDE.

(a) Suppose u_{ij} is a grid function on the $2D$ square with mesh size h. Suppose $\triangle_h u = f$ with $f_{ij} \leq 0$ for all i, j. Show that $\max_{i,j} u_{ij}$ is attained on the discrete boundary, which means either $i = 0$ or $i = N$ or $j = 0$ or $j = N$. Hint: in the interior (i.e. not at the boundary), u_{ij} is related to the average of its four neighbors.

(b) Assuming $f_{ij} \geq 0$ for all i, j, show that the minimum of u is attained at the boundary.

(c) Suppose $|\triangle_h u| \leq M$ for all i, j, and that $u_{ij} = 0$ on the discrete boundary. Show that

$$\begin{aligned} u_{ij} &\geq \frac{-M}{2} x_i (1 - x_i) \quad \text{for all } (i,j) \\ u_{ij} &\leq \frac{M}{2} x_i (1 - x_i) \quad \text{for all } (i,j) \\ |u_{ij}| &\leq \frac{M}{8} \quad \text{for all } (i,j) . \end{aligned}$$

where $x_i = ih = i/N$. Hint: Define $v_{ij} = \frac{-M}{2} x_i (1 - x_i)$, and $w_{ij} = u_{ij} \pm v_{ij}$ and figure out the sign of $\triangle_h w$ and the sign of w on the discrete boundary. This proves that the discrete Laplacian is stable in l^∞.

(d) Suppose that $u(x,y)$ is a smooth function that satisfies $\triangle u = f$ (with a smooth f) in the unit square and $u = 0$ on the boundary. Suppose that u_h is the finite difference approximation. Use the stability bound from part (c) to show that $|u_{h,ij} - u(x_i, y_j)| \leq Ch^2$.

4. **Programming.** Download the VisIt graphics package and install it on your system. Also download the tarball posted with this assignment and open it in a directory of its own. Type `make heat` and the code should compile (in a Unix or Linux window). Type `./heat` and you should get a plotfile `something.vtk`. If this doesn't work for you, please post ASAP on the class message board. Be sure to say what system you are using. Then open visit and use the `open` button to open the `.vtk` file. Click on the `+ add` button and select `Temperature`. Then (or possibly before) choose a `pseudocolor` plot and click on the `draw` button. The plot that appears should be the same as the one you can download with the assignment.

The code solves the PDE

$$\nabla \cdot (D(u)\nabla u) \; = \; 0 \; , \tag{1}$$

on the $2d$ domain $0 \leq x \leq L_x$ and $0 \leq y \leq L_y$, with boundary conditions

$$\partial_x u(0,y) \; = \; \partial_x u(L_x, y) \; = \; 0 \tag{2}$$

and

$$u(x,0) \; = \; u_{\text{bot}}(x) \;\; , \quad u(x, L_y) \; = \; u_{\text{top}} \; . \tag{3}$$

This u describes a temperature field. The heat conduction coefficient is temperature dependent with

$$D(u) \; = \; a \; + \; bu \; . \tag{4}$$

(a) The discretization is like the one we gave in class, with explicitly computed fluxes and flux differencing. Show that the method used (which you will have to figure out from the code) is second order accurate in the sense of local truncation error.

(b) There are several bugs in the code (some of which I am aware of). One is that the top and bottom boundary conditions (3) are applied a distance $L_y + \Delta y$ apart, which is off by Δy. Change the initialization code in `main` to fix this.

(c) Change the bottom boundary condition to make u_{bot} a constant. Then the whole solution should be independent of x. Moreover, it is easy to see that the solution has the form $u(y) = \alpha + (\beta + \gamma y)^{1/2}$. Show that this is true for any u_{top} and u_{bot} and a and b (if all those are non-negative) and find explicit the formula for u in terms of u_{top}

and u_{bot} when $a = 0$. Check that the computed solution agrees with this formula when the mesh is fine enough. Warning: another bug makes the code give the wrong answer if $\Delta x \neq \Delta y$. We will fix that one next week.

(d) Type `make tarball` when you are done to put all your work into a tarball and upload the tarball on the blackboard site. I will post instructions on exactly how to do this. Hand in only printouts of results, not the code.

(e) If you have time and want to play, try looking for boundary layer behavior. If u_{bot} is very close to zero the and $a = 0$, the diffusion coefficient becomes very small near the bottom boundary. This makes it possible for the solution to change rapidly there. Make some plots to see these small boundary layers. Warning: you need to take fine grids to see thin boundary layers. With fine grids, it takes more iterations to converge. The code could be slow.