Numerical Methods II, Courant Institute, Spring 2016
http://www.math.nyu.edu/faculty/goodman/teaching/NumMethII2016/index.html
Always check the `classes` message board before doing any work on the assignment.

Assignment 4, due April ??

1. (*Strang splitting*) Consider the differential equation system

$$\dot{u} = f(u) + g(u) \ .$$

Splitting means alternating between solving the different terms separately. Suppose $S_f(u,t)$ and $S_g(u,t)$ are the solution operators for the separate parts. That is, $u(t) = S_f(u,t)$ is the solution at time t for $\dot{u} = f(u)$, etc.

$$\frac{d}{dt} S_f(u,t) = f(S_f(u,t)) \ , \quad S_f(u,0) = u \ ,$$
$$\frac{d}{dt} S_g(u,t) = g(S_g(u,t)) \ , \quad S_g(u,0) = u \ .$$

Choose a time step Δt. Simple splitting means solving first $\dot{u} = f$ for time Δt, then solving $\dot{u} = g$ for time Δt, etc. If $U_n \approx u(t_n)$ is the numerical solution, simple splitting is

$$U_{n+1} = S_g(S_f(U_n, \Delta t), \Delta t) \ .$$

It may be clearer to express this as

$$U_n \quad \xrightarrow{S_f(\cdot,\Delta t)} \quad V_n = S_f(U_n, \Delta t) \quad \xrightarrow{S_g(\cdot,\Delta t)} \quad U_{n+1} = S_g(V_n, \Delta t) \ .$$

This may be a bit abstract if we do not have algorithms to calculate S_f or S_g exactly.

   (a) Show that simple splitting is formally first order accurate. That is

   $$S_{f+g}(u, \Delta t) = S_g(S_f(u, \Delta t), \Delta t) + \Delta t R(u, \Delta t) \ ,$$

   where $R = O(\Delta t)$.

   (b) *Strang splitting* is the more symmetrical version with two half steps of f on the outside and one full step of g on the inside:

   $$\begin{aligned} U_n \quad &\xrightarrow{S_f(\cdot,\frac{1}{2}\Delta t)} \quad V_n = S_f(U_n, \tfrac{1}{2}\Delta t) \\ &\xrightarrow{S_g(\cdot,\Delta t)} \quad W_n = S_g(V_n, \Delta t) \\ &\xrightarrow{S_f(\cdot,\frac{1}{2}\Delta t)} \quad W_n = S_f(W_n, \tfrac{1}{2}\Delta t) \ . \end{aligned}$$

   Show that Strang splitting is second order (in the same sense in which simple splitting is first order).

(c) Show that n steps of Strang splitting (computing U_n from U_0) may be computed using 2 applications of $S_f(\cdot, \frac{1}{2}\Delta t)$, and n applications of $S_g(\cdot, \Delta t)$ and $n-1$ applications of $S_f(\cdot, \Delta t)$. Conclude that the work for Strang splitting is almost the same as the work for simple splitting even though Strang splitting has three "steps" per time step while simple splitting has just two.

(d) Suppose that if $S^h_f(u, \Delta t)$ and $S^h_g(u, \Delta t)$ are second order accurate approximations to S_f and S_g respectively, in the sense of part (a). Show that the Strang splitting using S^h_f and S^h_g instead of the exact versions results in a time stepping method that is second order accurate.

(e) Consider the two dimensional advection-diffusion equation

$$\partial_t u + \partial_x (v_x(x,y)u) + \partial_y (v_y(x,y)u) = D \triangle u \; . \tag{1}$$

The function $u(x, y, t)$ is the density of particles that are advected (carried) by an *advection velocity* $v(x, y) = (v_x(x, y), v_y(x, y))$ while at the same time diffusing with diffusion coefficient D. Equations like this are used to model the dispersal of pollution in atmospheric winds. Writing the equation in *conservation form* ensures that the total amount of particle (number of particles) doesn't change, which is

$$\int\int u(x,y,t)\, dxdy = Const \; .$$

We solve this equation on an $n \times n$ mesh with $\Delta x = \Delta y$, and fixed time step Δt. The advection equation, without diffusion, is

$$\partial_t u + \partial_x (v_x(x,y)u) + \partial_y (v_y(x,y)u) = 0 \; .$$

We take one time step of this using the Richtmeyer predictor/corrector conservative version of Lax Wendroff. The diffusion part, without advection, is

$$\partial_t u = D \triangle u \; .$$

We take one time step of this using second order 5 point differencing in space (three points in x and three points in y) and Crank Nicholson in time. Show that these ingredients, plus Strang splitting, gives a second order and stable scheme for the combined advection diffusion equation.

2. (*The same thing, but simpler*) A linear constant coefficient differential equation system takes the form

$$\dot{u} = Au \; ,$$

where $u(x) \in \mathbb{R}^n$ and A is a fixed $n \times n$ matrix. Let $S_A(u, t)$ be the solution operator. This means the same as in exercise (1), namely that if

$u(t) = S_A(u_0, t)$, then $\dot{u}(t) = Au(t)$ and $u(0) = u_0$. The *matrix exponential* is written e^{tA}, and is defined by the formula

$$e^{tA} = I + tA + \tfrac{1}{2}t^2A^2 + \cdots = \sum_{n=0}^{\infty} \frac{t^n}{n!} A^n \; . \tag{2}$$

More properly, the matrix exponential is

$$e^{A} = I + A + \tfrac{1}{2}A^2 + \cdots = \sum_{n=0}^{\infty} \frac{1}{n!} A^n \; . \tag{3}$$

The matrix exponential of tA is e^{tA}.

(a) Show that the infinite sum (3) converges. One way to do this is to let $a = \|A\|$. Then $\|A^n\| \leq a^n$. Each term in the sum (3) is bounded by the corresponding $\frac{a^n}{n!}$. The sum

$$e^a = \sum_{n=0}^{\infty} \frac{a^n}{n!}$$

converges for any a. In fact, the sum "starts converging" when $n > a$, because then

$$\frac{a^{n+1}}{(n+1)!} < \frac{a^n}{n!} \; .$$

(b) Show that the matrix exponential gives the solution to the linear differential equation system in the sense that

$$\frac{d}{dt} e^{tA} = Ae^{tA} = e^{tA}A \; ,$$

and if $\dot{x} = Ax$, then

$$x(t) = e^{tA}\, x(0) \; .$$

One way to do this is to differentiate the sum (2) with respect to t.

(c) Show that

$$\left\| e^{tA} - I - tA \right\| = O(t^2)$$

.

$$\left\| e^{tA} - I - tA - \frac{t^2}{2} A^t \right\| = O(t^3)$$

.

etc.

(d) Show that

$$e^{\Delta t(A+B)} = e^{\Delta tA} e^{\Delta tB} + O(\Delta t^2) \; .$$

Show that this is equivalent to the first order accuracy of simple splitting for the linear ODE system

$$\dot{x} = Ax + Bx \; . \tag{4}$$

(e) Show that

$$e^{\Delta t(A+B)} = e^{\frac{1}{2}\Delta tA} e^{\Delta tB} e^{\frac{1}{2}\Delta tA} + O(\Delta t^3) \ .$$

Be careful – matrix multiplication is not commutative. Show that this is equivalent to the second order accuracy of Strang splitting for the linear ODE system (4). If you want to spend an hour looking this up in Wikipedia rather than half an hour doing it yourself, look for the *Baker Campbell Housdorff* formula.

3. Write a code in Python to solve the PDE (1) on an $n \times m$ mesh in the domain with $0 \leq x \leq L$ and $0 \leq y \leq 1$ with periodic boundary conditions. It is natural to choose $\Delta x = \Delta y$, so the relation between n and m is determined by the aspect ratio, L. Take $L \neq 1$ only if you are interested in different shapes. Use the velocity field

$$\begin{aligned} v_x(x, y) &= +A \sin(2\pi x) \cos(2\pi y) \ , \\ v_y(x, y) &= -A \cos(2\pi x) \sin(2\pi y) \ . \end{aligned}$$

The PDE depends on parameters A, which governs the speed of advection, and D, which governs the diffusion. Write a code that makes a movie of the solution by plotting the solution every so many time steps. Each movie frame should be a contour plot of the solution at some time.

(a) Write a program to make a movie with initial conditions $u(x, y, 0) = \exp(5(\sin(y) - 1))$. The contour plot of this initial condition should resemble a vertical bar near $y = \frac{1}{4}$. The solution at later time distorts the bar. Draw the vector field v to understand this distortion.

(b) Run the program with a variety of D values and $A = 1$ to see the effect of diffusion.

(c) Do a mesh refinement study $\Delta x \to 0$ and $\Delta y \to 0$. The code should calculate Δt as a function of Δx and A and D so you don't have to adjust Δt manually (which is clumsy and error prone). You should see that you need a finer mesh to run the solution for a long time if D is small.

(d) Experiment with initial data different from zero only when $frac12 - r \leq x \leq \frac{1}{2} + r$ and $frac12 - r \leq y \leq \frac{1}{2} + r$. See how the advection velocity field moves the solution around. Experiment with various values of D and A.