Tutorial problems for Lectures 8-11. Due Monday May 14, 2017.

Problem 1: Write linear least-squares code to fit sines and cosines to evenly sampled data. Pick the sines and cosines to have integer numbers of periods, so you pick 100 numbers, should have $\sin/\cos(2\pi n(0:50)/100)$. Compare your fit parameters to the FFT of the data. Note - sin should go from 1 to 49 instead of 0 to 50. Why is this? (10)

Problem 2: Take the mcmc sample code. Add a Lorentzian class $f(x) = \frac{a}{b+(x-c)^2}$. Run the fit, and show you get correct answers. (10)

Problem 3: Write a finite-volume advection solver similar to the one we saw in class. Make this one have a negative velocity, and give it periodic boundary conditions. Plot the solution as a function of time - how does it behave? How does the total mass behave with time? (10)

Problem 4: Show that k=0 (infinitely large scale) is still stable even when CFL condition is violated. (5) What other k values are still stable when the CFL condition is violated? (5)

Problem 5: Look at hydro1d.py. youll see an assert guaranteed to fail at the end of get_bc, the boundary condition routine. Why did I do this? (5)

Problem 6: Further on in hydro1d, where the derivatives are getting calculated, theres a factor of 1/2 in the pressure gradient. Why? (5)

Bonus: Modify the mcmc sample code to run a short chain, use that to estimate the parameter errors, and then run a longer chain using the error estimates. What is your accept fraction? Can you get this to $\sim$0.25 by rescaling the covariance? You can do this for either the Gaussian example or the Lorentzian from Problem 4.(5)

Bonus 2: Look at the time step calculator. Right now it doesnt implement the CFL condition. Put in a proper timestep calculator. This should find the globally smallest stable timestep and return the input times this value. So, if global CFL limit is 0.3 and we pass in 0.1, the return value should be 0.03 (5).