ASTR 260 - EXAM 2 - FALL 2014

NAME:

INSTRUCTIONS: You are to finish all the problems in this exam. Choose two problems to solve in class and one to solve for Wednesday November 19. **Clearly indicate which problem you are solving as take home.** When you work on the take-home problem you are to do so without help from anyone.

Problem 1: The biochemical process of *glycolysis*, the breakdown of glucose in the body to release energy, can be modeled by the equations

$$\frac{\mathrm{d}x}{\mathrm{d}t} = -x + ay + x^2 y, \qquad \frac{\mathrm{d}y}{\mathrm{d}t} = b - ay - x^2 y.$$

Here x and y represent concentrations of two chemicals, ADP and F6P, and a and b are positive constants. One of the important features of nonlinear linear equations like these is their *stationary points*, meaning values of x and y at which the derivatives of both variables become zero simultaneously, so that the variables stop changing and become constant in time. Setting the derivatives to zero above, the stationary points of our glycolysis equations are solutions of

$$-x + ay + x^2 y = 0, \qquad b - ay - x^2 y = 0.$$

a) Show that the equations can be rearranged to read

$$x = y(a + x^2), \qquad y = \frac{b}{a + x^2}$$

and write a program to solve these for the stationary point using the relaxation method with $a = 1$ and $b = 2$. You should find that the method fails to converge to a solution in this case.

b) Find a different way to rearrange the equations such that when you apply the relaxation method again it now converges to a fixed point and gives a solution.

Problem 2:

In laulima you will find the file `piano.txt` which contains data representing the waveform of a single note played on a piano. The data file contains the signal at a rate of 44,100 samples per second.

a) How long does the sound last? Express your result in seconds. Show all your work, don't just give the answer.

b) Write a program that loads the waveform from this file and plots it.

c) Include in your program a way to calculate its discrete Fourier transform. Use the built-in FFT function from numpy.fft called rfft, i.e. import `rfft` from `numpy.fft`. The function `rfft` will calculate the first 50,001 Fourier coefficients. To use it, invoke:
Fourier-Coefficient = rfft(array)
You will not need more than 50,001 coefficients.

d) Make your program plot the power spectrum of the first 10 000 coefficients.

e) In laulima you will find a second file `piano2.txt` that contains only the even lines of `piano.txt`. Compare the peak of the first Fourier coefficient of the two signals (are they the same?, which one is larger, which one is smaller?).

f) Extra Credit: The waveform was recorded at the industry-standard rate of 44 100 samples per second. From your Fourier transforms of `piano.txt` and `piano2.txt` results calculate what note they were playing. Hint: The musical note middle C has a frequency of 261 Hz and multiples of 2 result higher octaves while multiples of $\frac{1}{2}$ result in lower octaves.

Problem 3: Poker

a) Write a program that generates and prints out five cards between 1 and 13 (Ace, 2-10, JQK), to simulate a deal.

b) Modify your program to simulate a total of $10,000 in bets each at $5 and count the number of times you get a double ace. Divide by the number of bets to get the *fraction* of times you get a double ace.

c) Is this fraction always the same?

d) If you had $10,000,000 to bet, how does the *fraction* of times you get a double ace change, if at all?