PH 235 SPRING 2017

EXTRA CREDIT ASSIGNMENT: 5 POINTS TOWARDS TOTAL COURSE GRADE

Assigned Date: April 19, 2017

Due Date: 11:30pm, May 5, 2017 only via moodle (No late submissions)

Total Points: 125

All problems in this assignment are from the optional textbook for the course *Computational Physics* authored by Mark Newman.

Instructions:

1. Create a folder titled YourLastName-EC.
2. In this parent folder, create subfolders, one for each of the problems below and name each subfolder as YourLastName-ExerciseNumber for easier tracking.
3. Populate the folders with (a) **appropriately commented** solution code for each problem, (b) screen shots of results when possible and (c) a text document with a short description of your approach, whether you successfully solved the problem. If you did not solve the problem, explain solution status (partial credit will be given for reasonable attempts).
4. Submit a zipped version of the parent folder to the moodle assignment link.

Problem 1: Ex 8.18 Oscillating chemical reactions (30 points)

The *Belousov–Zhabotinsky reaction* is a chemical oscillator, a cocktail of chemicals which, when heated, undergoes a series of reactions that cause the chemical concentrations in the mixture to oscillate between two extremes. You can add an indicator dye to the reaction which changes color depending on the concentrations and watch the mixture switch back and forth between two different colors for as long as you go on heating the mixture.

Physicist Ilya Prigogine formulated a mathematical model of this type of chemical oscillator, which he called the "Brusselator" after his home town of Brussels. The equations for the Brusselator are

$$\frac{\mathrm{d}x}{\mathrm{d}t} = 1 - (b+1)x + ax^2y, \qquad \frac{\mathrm{d}y}{\mathrm{d}t} = bx - ax^2y.$$

Here x and y represent concentrations of chemicals and a and b are positive constants.

Write a program to solve these equations for the case $a = 1$, $b = 3$ with initial conditions $x = y = 0$, to an accuracy of at least $\delta = 10^{-10}$ per unit time in both x and y, using the adaptive Bulirsch–Stoer method. Calculate a solution from $t = 0$ to $t = 20$, initially using a single time interval of size $H = 20$. Allow a maximum of $n = 8$ modified midpoint steps in an interval before you divide in half and try again.

Make a plot of your solutions for x and y as a function of time, both on the same graph, and have your program add dots to the curves to show where the boundaries of the time intervals lie. You should find that the points are significantly closer together in parts of the solution where the variables are changing rapidly.

Hint: The simplest way to do this calculation is to make use of recursion, the ability of a Python function to call itself. Write a user-defined function called, say, `step(r,t,H)` that takes as arguments the position vector $\mathbf{r} = (x, y)$ at a starting time t and an interval length H, and returns the new value of $\mathbf{r}$ at time $t + H$. This function should perform the modified midpoint/Richardson extrapolation calculation until either the calculation converges to the required accuracy or you reach the maximum number $n = 8$ of modified midpoint steps. If it fails to converge in eight steps, have your function call itself, twice, to calculate separately the solution for the first then the second half of the interval from t to $t + H$, something like this:

```
r1 = step(r,t,H/2)
r2 = step(r1,t+H/2,H/2)
```

(Then *these* functions can call themselves, and so forth, subdividing the interval as many times as necessary to reach the required accuracy.)

Problem 2: Ex 10.8 Importance sampling (30 points)

Calculate a value for the integral

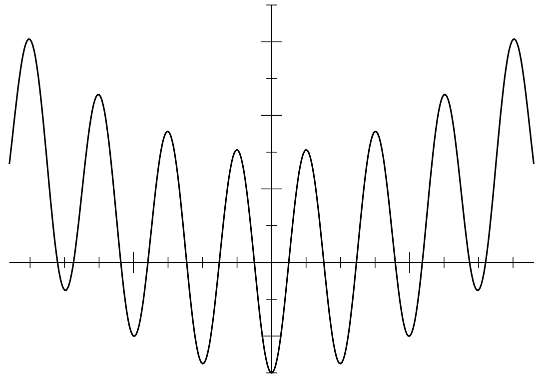
$$I = \int_0^1 \frac{x^{-1/2}}{e^x + 1}\,dx,$$

using the importance sampling formula, Eq. (10.42), with $w(x) = x^{-1/2}$, as follows.

1. Show that the probability distribution $p(x)$ from which the sample points should be drawn is given by
$$p(x) = \frac{1}{2\sqrt{x}}$$
and derive a transformation formula for generating random numbers between zero and one from this distribution.
2. Using your formula, sample $N = 1\,000\,000$ random points and hence evaluate the integral. You should get a value around 0.84.

Problem 3: Ex 10.10 Global minimum of a function (40 points)

Consider the function $f(x) = x^2 - \cos 4\pi x$, which looks like this:



Clearly the global minimum of this function is at $x = 0$.

1. Write a program to confirm this fact using simulated annealing starting at, say, $x = 2$, with Monte Carlo moves of the form $x \to x + \delta$ where δ is a random number drawn from a Gaussian distribution with mean zero and standard deviation one. Use an exponential cooling schedule and adjust the start and end temperatures, as well as the exponential constant, until you find values that give good answers in reasonable time. Have your program make a plot of the values of x as a function of time during the run and have it print out the final value of x at the end. You will find the plot easier to interpret if you make it using dots rather than lines, with a statement of the form `plot(x,".")` or similar.
2. Now adapt your program to find the minimum of the more complicated function $f(x) = \cos x + \cos \sqrt{2}x + \cos \sqrt{3}x$ in the range $0 < x < 50$.

Hint: The correct answer for part (b) is around $x = 16$, but there are also competing minima around $x = 2$ and $x = 42$ that your program might find. In real-world situations, it is often good enough to find any reasonable solution to a problem, not necessarily the absolute best, so the fact that the program sometimes settles on these other solutions is not necessarily a bad thing.