Ordinary Differential Equations and Chaotic Systems

HW 4:	Wednesday, Sep. 19, 2018
DUE:	Wednesday, Oct. 3, 2018 (you have two weeks to do this)
READ:	*Numerical Recipes in C++*, Section 17.0 and 17.1, page 899-910 on 4th order Runge-Kutta solution of differential equations
	OPTIONAL: Landau, Paez, and Bordeianu, ODE and Runge-Kutta, section 9.1-9.5.2 chaotic systems, section 12.1-12.14
	F.C.Moon, *Chaotic and Fractal Dynamics*, Wiley 1992 discusses chaotic systems in more detail. The simulation of one or more chaotic systems may make an interesting final project.

PROBLEM (10 points):

The following set of equations results from modeling a light wave in a system of two level atoms:

$$\begin{aligned} \frac{dx}{dt} &= -y && (1)\\ \frac{dy}{dt} &= x + ez && (2)\\ \frac{dz}{dt} &= -ey && (3)\\ \frac{d^2e}{dt^2} + \mu^2 e &= \beta\frac{dy}{dt} && (4) \end{aligned}$$

$x(t)$, $y(t)$, and $z(t)$ are related to the quantum mechanical amplitudes of each level (and their complex conjugates) and $e(t)$ is related to the overall field strength. This is the Jaynes-Cummings model described in section 3 of Ackerhalt[1] et al. Please refer to this article (and its references) for a more complete derivation of this set of equations if you are interested. A short explanation is that the atom wave function is in a superposition of two quantum mechanical states ψ_1 and ψ_2:

$$\psi(t) = c_1(t)\psi_1 + c_2(t)\psi_2 \qquad (5)$$

where $c_1(t)$ and $c_2(t)$ are complex valued weighting amplitudes. The first three Jaynes-Cummings equations (above) result from the Schrödinger equation using the following real variables:

$$\begin{aligned} x(t) &= c_1^*(t)c_2(t) + c_1(t)c_2^*(t) \\ y(t) &= i[c_1^*(t)c_2(t) - c_1(t)c_2^*(t)] && (6)\\ z(t) &= |c_2(t)|^2 - |c_1(t)|^2 && (7) \end{aligned}$$

The last equation for $e(t)$ describes an oscillating electric field (proportional to $e(t)$) interacting with the atom (i.e. a light wave). Many constants have been absorbed into the definitions of the constants μ and β to make a simple dimensionless set of equations to solve. The goal of this homework is to solve this system numerically.

This is a nonlinear system and can exhibit chaotic behavior for some values of the parameters. Although there is no simple and precise definitions of a chaotic system, chaotic behavior is roughly defined as being bounded but not periodic or semi-periodic. Some other references are listed at the end.

To solve this system, first write a program using the 4th order Runge-Kutta method and test it. Write a general purpose subroutine for arbitrary order N (using dynamic memory with new/delete), similar to that described in rk4() in *Numerical Recipes*[5] that performs one time step. This should be a general routine so that you can test the same code on a problem with a known solution and then apply the identical code to an unknown problem. You may use rk4() directly or write your own version. The code in N.R. in C++ uses classes for the arrays. You may change these back to plain arrays or try the NR-vector class if you like. Vector class construction may be discussed in class shortly.

When developing code like this it is always advisable to test it on a system with a known answer to verify that it is working correctly. Therefore first test your program on the harmonic oscillator equation (without damping):

$$\frac{d^2y}{dt^2} + \omega^2 y \;=\; 0 \tag{8}$$

where ω is a constant (angular frequency of oscillation). With initial conditions $y(t = 0)$=1 and $y'(t = 0)$=0 this has a known solution:

$$y(t) = \cos(\omega t) \tag{9}$$

The Runge-Kutta method requires this equation to be written as two first order equations. The harmonic oscillator equation can also be written as:

$$\frac{dy_0}{dt} \;=\; y_1 \quad ; \quad \frac{dy_1}{dt} = -\omega^2 y_0 \tag{10}$$

Plot $y(t)$ vs. t (as produced by your program) for several cycles using ω =1.0 to verify that your code is working properly. You should quantitatively test both the period and the amplitude. You might also make a phase space plot of $y'(t)$ versus $y(t)$ which should produce a circle that closes on itself.

Next modify your program (using the 4th order Runge-Kutta method) to find the solution for $x(t), y(t), z(t)$ and $e(t)$ described by the equations in the Jaynes-Cumming model. Use the following parameter values and initial conditions:

$$\begin{aligned}
\mu &= \beta = 1 && (11)\\
x(0) &= y(0) = e'(0) = 0 && (12)\\
z(0) &= 1 && (13)\\
e(0) &= 10^{-6} = 1e-6 && (14)
\end{aligned}$$

Calculate $x(t)$, $y(t)$, $z(t)$, $e(t)$ and $e'(t)$ for $0 < t < 200$. Vary the sampling size $h = \Delta t$ to get a stable answer. To save time you may just look at $x(t)$ vs. t to judge the stability (reproducibility). The chaotic behavior of this system also causes numerical stability problems over long times. You may find that you need many 1,000's of points. To save disk space etc. you only need to print out a few 1,000 points but evaluate many points in-between those you print out (i.e. you need a small h to get a stable numerical solution but you don't need to see everything in-between).

Beware that (in C) 1/6=0 but 1.0/6.0=0.166666667 because the first is done using integer arithmetic and the second is done using floating point arithmetic.

Plot $z(t)$ vs. t and $y(t)$ vs. $x(t)$. You will find that the values of the parameters given above produce a chaotic behavior.

OPTIONAL(1): Graph $y(t)$ vs. $x(t)$ for other values of β. Some values may not produce chaotic behavior ($\beta = 0$ produces an orderly behavior).

OPTIONAL(2): Try different starting point. Chaotic system are very sensitive to small changes in the initial conditions. Small changes produce similar behavior in the near term but large changes in the long term solution.

OPTIONAL(3): Try plotting (x,y,z) in a 3D plot (use plot3 in Matlab, python or equivalent). Many programs will let you grab the drawing with the mouse and move it around to get a better idea of the 3D shape. The following python code segment produces a 3D plot.

```
from pylab import *
from mpl_toolkits.mplot3d import Axes3D
:
fig = figure(5)
ax = fig.gca(projection='3d')
:
plot( x, y, z, 'k' )
:
```

References

[1] J. R. Ackerhalt, P. W. Milonni and M.-L. Shih, "Chaos in Quantum Optics", Physics Reports Vol. 128, 1985, pages 205-300.

[2] H. Haken, "Analogy Between Higher Instabilities in Fluids and Lasers", Phys. Lett. 53A, 1975, p.77-78

[3] H. J. Korsch and H.-J. Jodl, *Chaos, A Program Collection for the PC*, 2nd edit., (Springer-Verlag, 1999), section 12.4.

[4] E. N. Lorenz, "Deterministic Nonperiodic Flow", J. Atmos. Sci. 20, 1963, p.130.

[5] W. H. Press, S. A, Teukolsky, W. T. Vetterling and B. P. Flannery *Numerical Recipes, The Art of Scientific Computing, 3rd edit.*, Camb. Univ. Press 2007.

[6] S. H. Strogatz, *Nonlinear Dynamics and Chaos*, (Perseus Publishing, 1994), section 4.6.