Computational Lab 6 : Solving ODEs, part I

Due October 18th, 2019 (@ 17:00)

- Read this document and do its suggested readings to help with the pre-labs and labs.
- Ask questions if you don't understand something in this background material: maybe we can explain things better, or maybe there are typos.
- Whether or not you are asked to hand in pseudocode, you **need** to strategize and pseudocode **before** you start coding. Writing code should be your last step, not your first step.
- Test your code as you go, **not** when it is finished. The easiest way to test code is with `print('')` statements. Print out values that you set or calculate to make sure they are what you think they are.
- Practice modularity. It is the concept of breaking up your code into pieces that as independent as possible form each other. That way, if anything goes wrong, you can test each piece independently.
- One way to practice modularity is to define external functions for repetitive tasks. An external function is a piece of code that looks like this:

```
def MyFunc(argument):
    "''A header that  explains the function
    INPUT:
    argument [float] is the angle in rad
    OUTPUT:
    res [float] is twice the argument"''
    res = 2.*argument
    return res
```

  In this lab, **you will implement your own RK4 integrator** and likely a wrapper to integrate your ODEs. You will then use this RK4 integrator many times. You should be able write the integrator to take any set of ODEs and solve them without changing the integrator code. You will lose 'code' marks if there is excessive copy-pasting of code!

Physics Background

van der Pol oscillators Harmonic oscillators are found in many physical systems. The motion can be expressed in the simple harmonic oscillator equation:

$$\frac{d^2x}{dt^2} = -\omega^2 x, \tag{1}$$

where ω is the angular frequency.
The van der Pol oscillator[1] is found in electronic circuits involving vacuum tubes and in laser applications. The van der Pol oscillator adds an additional term to the oscillator equation to give the equation

$$\frac{d^2x}{dt^2} - \mu\left(1-x^2\right)\frac{dx}{dt} + \omega^2 x = 0, \tag{2}$$

where $\mu = 0$ is the original simple harmonic oscillator.

[1] If you have an hour spare, the Youtube video https://www.youtube.com/watch?v=O1lQrHemPsw discusses the van der Pol oscillator in various limits.

The van der Pol oscillator can be driven by a sinusoid by adding a forcing term on the right hand side of equation 2:

$$\frac{d^2x}{dt^2} - \mu\left(1-x^2\right)\frac{dx}{dt} + \omega^2 x = A\sin\left(\omega_2 t\right), \tag{3}$$

The SEIR disease model simulates the spread of infectious disease through a population. This model divides the population into the categories *Susceptible*, *Exposed*, *Infectious*, and *Recovered*. The rate at which the categories change is shown schematically in figure 1. The initial rate **b** is the population birth rate. The outgoing arrows represent death rates **d** in each group. In each case the rate is a fraction of the population, so that the contribution of births to the S population is

$$\left(\frac{dS}{dt}\right)_{\text{births}} = bS \tag{4}$$

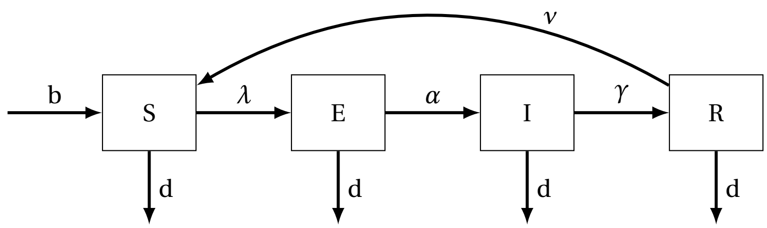

Figure 1: Population dynamics according to the SEIR model.

If newborn babies are vaccinated against a disease but the current population is not (this does happen!) at a rate of P of the newborn population the model is altered to look like figure 2

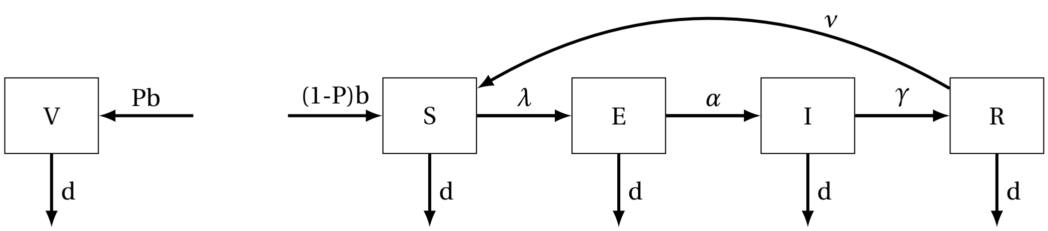

Figure 2: Population dynamics according to the SEIRV model with newborn vaccinations.

Computational Background

Runge-Kutta: The Runge–Kutta fourth order method (RK4) is a widely used algorithm for solving systems of ODEs. In general it is much more accurate than the Euler or Euler–Cromer methods for ODEs used in initial value problems.
The RK4 method involves using the model function to predict the next timestep in the ODE (as in the Euler method) but using those predictions *not* as final value, but as a way to improve the prediction using intermediate values to better approximate the ODE evolution.
As discussed in the text, this method involves calculating the RHS vector of $d\vec{r}/dt = \vec{f}(\vec{r},t)$ at various intermediate points between steps. Full implementation requires coding the following 5 lines which are iterated over t values:

$$
\begin{aligned}
\vec{k}_1 &= h\vec{f}(\vec{r},t) && (5)\\
\vec{k}_2 &= hf(\vec{r}+\frac{1}{2}\vec{k}_1, t+\frac{1}{2}h) && (6)\\
\vec{k}_3 &= hf(\vec{r}+\frac{1}{2}\vec{k}_2, t+\frac{1}{2}h) && (7)\\
\vec{k}_4 &= hf(\vec{r}+\vec{k}_3, t+h) && (8)\\
\vec{r}(t+h) &= \vec{r}(t)+\frac{1}{6}(\vec{k}_1+2\vec{k}_2+2\vec{k}_3+\vec{k}_4) && (9)
\end{aligned}
$$

Questions

1. **[40%] The van der Pol oscillator**

   (a) Convert the simple harmonic oscillator into two coupled first order equations and write a program to integrate an 1 Hertz pendulum for 60 seconds. You should have 60 oscillations starting from a stationary pendulum at $x = 1$. Plot position vs time and phase space plot (velocity vs position) to confirm your code is working. *It's much easier to test your integrator with a known function like the harmonic oscillator than a new function like the Van der Pol oscillator.*

   NOTHING TO SUBMIT

   (b) Modify your equations and code to simulate the van der Pol oscillator given in equation 2. Plot graphs of position vs time and the phase space plot for $\mu = 0.1$, $\mu = 2$, $\mu = 5$. Include the simple harmonic oscillator on the plot with the same frequency as a reference. The amount of time to include in your position vs time plot can be altered to make the plots reasonable, but should be 60 seconds or less.

   SUBMIT YOUR CODE AND PLOTS

   (c) Add a sinusoidal driving term to the van der Pol oscillator with amplitude A and angular frequency ω_2. Test the code with the following condtions. Describe the behaviour of the oscillator for each set of conditions.

   - $\omega_2 = 0.5\omega$,
   - $\omega_2 = \omega$,
   - $\omega_2 = 4\omega$,

   using $\mu = 1, A = 8.53, \omega = 1$

   SUBMIT YOUR CODE AND PLOTS, AND WRITTEN DESCRIPTION OF EACH OSCILLATION.

2. **[60%] The SEIR model**

   (a) Write the four rate equations that correspond to the schematic. To model the infection spreading you should make the `S` to `E` transition occurs at a rate

   $$\lambda = \beta \frac{I}{N}, \qquad (10)$$

   where β is another constant, `I` is the infected population, and `N` is the total population $N = S+E+I+R$ (in other words, your likelihood of being infected depends on a base rate β and the fraction of the population already infected). Show that the population is static if the birth rate is equal to the death rate.

   SUBMIT THE EQUATIONS YOU DERIVE

   (b) What does this model imply about the mortality of the infection?

   NOTHING TO SUBMIT

(c) Solve the SEIR equations with the following input parameters: $\beta = 0.2$, $\gamma = 0.1$, $b = 10^{-4}$, $d = 10^{-4}$, $\alpha = 0.1$, $\nu = 10^{-3}$. Initial conditions are $I = 10^{-6}$, $N = 1$, and $S_0 = N - I_0$. Plot the population fractions (a log scale might help) as a function of time assuming that the rate terms carry units of `people/day` where appropriate and the population terms carry units of `people`. Integrate for 10 years.

SUBMIT YOUR CODE AND PLOT OF THE DIFFERENT POPULATION GROUPS

(d) Update the SEIR model to include an increased mortality factor. That is, the disease causes death at a higher rate than the background rate, but only during an infection. Set this mortality rate to be twice the background rate (2×10^{-4}). Integrate for long enough to to determine the number of years it would take for the population to fall by half.

SUBMIT THE PLOT OF POPULATION GROUPS, AND THE WRITTEN ANSWER TO THE QUESTION

(e) After first identifying the infectious disease from the previous part, doctors work for fifteen years before finding a vaccine that can only be given to newborn babies. At first the vaccine is tested on a small population and the fraction of the newborn population that receive the vaccine doubles each year. i.e. the vaccine rate applied to newborns is 0 for the first 15 years, then the fraction of new babies that are vaccinated is given by

$$P = 1 - 2^{\frac{-(t-1825)}{365}},$$

assuming the time units are in days. Add a vaccinated population term where vaccinated people do not become infected and experience only the background mortality rate.

Integrate this model for long enough to determine the new population level at the same time you found in the last part for halving the population. Comment on the stability of the new population.

SUBMIT YOUR CODE AND THE PLOT INCLUDING THE VACCINATED POPULATION. INCLUDE YOUR WRITTEN ANSWERS TO THE QUESTIONS

Question:	1	2	Total
Points:	40	60	100