MS6021 Scientific Computing

Lecture/Work package #4

+

Assignment #4

(20%; due end-Week 11,
recommended Wed-Week 11**)**

TOPICS:

A. **Initial-Value Problems for ODEs**

B. **SIR model for Spread of Disease**

A. Initial-Value Problems for ODEs

- Initial-value problem for first-order ODE:

$$\frac{d}{dt}y(t) = f\big(t, y(t)\big), \qquad y(0) = y_0$$

--Here $y(t)$ is the unknown scalar or vector function (understood as a column vector).

--Note that if $y(t)$ is a vector column, then f is also a vector column of the same length.

EXERCISE: consider Problems 1+2+3 from [Higham&Higham, p. 175, 178, 179] and specify $y(t)$ and $f\big(t, y(t)\big)$ (for Problems 2 +3, $y(t)$ and f will be vectors)

Build-in adaptive-step MatLab ODE solvers

- MatLab has a number of solvers for ODEs
  (see [Higham&Higham, p.190] for a list).
  They are easy to use and employ adaptive time step
- ode45 – see [Higham&Hagham, Section 12.2.1]
  EXERCISE: apply it to Problems 1+2+3 in [Higham&Hagham] using anonymous functions.
  Example for Problem 2:

```
f = @(t,y) [ y(2) ; -sin(y(1)) ]  % define a function with 2 inputs
    % second input y is a column vector of length 2
    % output is a column vector also of length 2
[t,y] = ode45(f, [0 10], [0.5;2]);  % run ode45 solver
                                     %with the initial conditions 0.5 and 2
plot(t,y,'-o')  % try to play with other initial conditions
 % NOTE; we solve a system of 2 ODEs, with 2 unknown functions
%  so on the graph there are 2 curves
plot(y(:,1),y(:,2),'-o')  %  plots y1(t)  v  y2(t); include axis names!
```

NOTE: for Assignment #4, you should include axis names and legends for all graphs that you produce…
LEARN to set Absolute and Relevant TOLERANCEs (see p.179).

Three explicit methods of ODEs:

For most initial-value problems one can use the so-called explicit methods: they are inexpensive as they don't involve solving nonlinear equations or linear/nonlinear systems. We shall consider 3 such methods

- **Explicit Euler method:**
  see Section 23.1 in Natalia's notes
  http://www.staff.ul.ie/natalia/MA4002/MA42_lec_17_32.pdf ;
  also Section 4.1.2 in [*Programming for Computations]*

  $$y^0 = y(0), \quad y^{n+1} = y^n + h\, f(t^n, y^n) \text{ where } t^n = nh, \; n = 1,2, \ldots$$

- **Predictor-corrector method:**
  see Section 23.2 in Natalia's notes
  http://www.staff.ul.ie/natalia/MA4002/MA42_lec_17_32.pdf ;
  also Section 4.3.5 in [*Programming for Computations]*

  $$y^0 = y(0), \quad y^* = y^n + h\, f(t^n, y^n)$$
  $$y^{n+1} = y^n + 0.5\, h\, [f(t^n, y^n) + f(t^{n+1}, y^*)], n = 1,2, \ldots$$

- One popular **4th-order Runge-Kutta method**:
  https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods
  (read Section 1 only!)

Assignment #4, part I (10%)

- Implement the above 3 methods as MatLab functions, with inputs and outputs similar to ode45, e.g.

  [t,y]=ExplicitEuler(f, tspan, y0, N)

  where

  the outputs [t,y] are as in ode45,

  f is a function handle for $f(t, y)$,

  tspan is a vector [a,b] with 2 entries, describing the interval on which the ODE is solved,

  y0 is the initial condition,

  N is the number of subintervals (i.e. $\mathrm{h} = (\mathrm{b} - \mathrm{a})$/N)

  NOTE: these functions should work for the case of y and f being vectors (as in Problems 2+3 that were considered above)

Assignment #4, part I (continued)

- Apply the above 3 functions to Problem #1 (t in [0,3]), Problem #2 (t in [0,10], with initial condition [1;1]) and Problem #3 (t in [0,50], with initial condition [1;1;1] and a=b=0.2, c=2.5).
- Compute max$|y^n - y(t^n)|$. Hint: For Problem 1, use its exact solution. For Problems 2+3, we don't know the analytical solution. Instead, for Problems 2+3, use the so-called reference solution, obtained by calling ode45 with Absolute and Relative Tolerances being 1e-20 and 1e-13, respectively (i.e. extremely small; see p. 179, Higham&Higham)
- Hence, present the max$|y^n - y(t^n)|$ values in the following table and explain the results:

	Prob#1, N=600	Prob#1, N=1200	Prob#2, N=500	Prob#2, N=1000	Prob#3, N=20000	Prob#3, N=40000
Explicit Euler						
Predictor-Corrector						
4th-order Runger-Kutta						

B. SIR model for Spread of Disease

- Let $S(t)$ = the population fraction of *susceptible* (healthy) individuals,
  $I(t)$ = the population fraction of infected individuals,
  $R(t)$ = the population fraction of “recovered” or “removed” individuals.

- Using dimensional time (in days), the full equations are

$$\frac{dS}{dt} = -\beta SI, \qquad \frac{dI}{dt} = +\beta SI - \gamma I, \qquad \frac{dR}{dt} = +\gamma I,$$

where $S(t) + I(t) + R(t) = 1$.

$1/\gamma$ = the average recovery period of the disease

$\beta = \gamma * R_{\text{number}}$

- Initial conditions : 1 infected individual at initial time $t0$, i.e.

$$I(t0) = 1/\text{population}, \qquad S(t0) = 1 - I(t0)\,, \qquad R(t0) = 0$$

- NOTE: no need to solve the equation for R(t) !

Assignment #4, part II (10%)

Fitting the parameters of an SIR model to influenza data using Least Squares

Q1. Consider the above SIR model for modelling B influenza with the following parameters (US Midwest, 2007)

$1/\gamma$ = 3 days the average recovery period of the disease

Population = 5.2e+7

R_{number} = 1.3

$t0$ = 250 (i.e. 250th day of year)

- For this model (equations for S and I only), employ ode45 with Absolute and Relative Tolerances 1e-20 and 1e-13.
  HINT: define f(t,y) as in part I, where y=[S;I], i.e. y(1)=S and y(2)=I

- The following tspan should be used:

  tspan = [$t0$, 7*(46:72)]

  RATIONAL: your numerical results will be compared with available incidence data for weeks 47:72 in Q2.

- Plot the graph of your computed solutions S(t) and I(t), and R(t)=1-S(t)-I(t)

Q2. Compare the numerical results obtained in Q1 with the following data using Least Squares metric as follows

- Download the **data** from http://sherrytowers.com/midwest_influenza_2007_to_2008.dat

  --The data can be easily transformed into a MatLab matrix by adding ";" at the end of rows, and then inserting the data in [].

  --We are interested only on columns #1 (week number) and column #5 (weekly incidence of B influenza in the tested fraction of population).

  --Furthermore, note that for weeks <47, the incidence numbers may be 0, so discard all data for those weeks.

  --Plot a graph (using the option '-o' or similar) of measured incidence (y axis) with the corresponding days (week*7) on the x axis.

  --Compute the area under the piecewise-linear curve as $\mathrm{SUM}_{\mathrm{inc}}$ of all incidence data.
- Posprocessing the numerical data for comparison:

  --Discard the initial values at initial time $t0$

  --Note that the incidence data ignore recovering patients, so they actually model the decrease in S(t) during the corresponding week, hence the incidence data correspond to –diff(S), so set incidence_modelled = –diff(S),

  --Note that only a fraction of the population was tested, hence before comparison, one needs to normalize the numerical data:

  --compute $\mathrm{SUM}_{\mathrm{numerical}}$ = sum of the all elements in incidence_modelled,

  --normalize incidence_modelled by multiplying it by ($\mathrm{SUM}_{\mathrm{inc}}$ /$\mathrm{SUM}_{\mathrm{numerical}}$)

  --plot incidence and normalized incidence_modelled on the same graph. Note that the areas under both curves should be equal, and the lengths of the 2 vectors should be equal.
- Finally compute the **least squares metric of the quality of the model** as the

  sum of squares of the elements in (incidence – incidence_modelled)

NOTE: The recovery period can roughly be estimated from observational studies of infected people, but the reproduction number is not easily available. The initial time is also unknown and needs to be estimated. In Q3+Q4, you shall measure the quality of the fit using the least squares metric (see Q2): the smaller, the better.

Note also that the least squares metric is not the best for this model (as all data points are "equally important" in this metric…)

Q3. Fitting the parameters of a SIR model: fitting the R number with fixed $t0$ =250

- **For the fixed initial time , compute the best fit for the reproduction number within the range** R_{number} = [1.21 : 0.001 : 1.24]'

  TIPS:

  --Start coding with a simpler range, such as R_{number} = [1.22; 1.23; 1.24]

  --Note that since R_{number} is a column vector, so is β .

  --To get full marks, avoid loops. Instead, set k = length or the range for R_{number} , and modify the definition of f(t,y) so that

  y is column vector of length 2k (rather than 2),
  y(1:k)=the vector of S values for the range of R_{number} ,
  y(k+1,2k)= the vector of I values for the range of R_{number}

Q4. Fitting the parameters of a SIR model: fitting the R number and t0

- Modify the code for Q3 by adding a loop for the range $t0$ = [200:250]

  Hence find the best pair of parameters ($t0$, R_{number}) that fits the available data.