Computational methods and applications (AMS 147)

Homework 6 - Due Wednesday, March 14

Please submit to CANVAS a .zip file that includes the following Matlab functions:

`tridiag_solver.m`

`AB3.m`

`solve_ODE_system.m`

Exercise 1 Write a Matlab function that implements the Thomas algorithm to solve tridiagonal linear systems of equations in the form

$$\begin{bmatrix} a_1 & c_1 & 0 & \cdots & 0 \\ e_1 & a_2 & c_2 & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & & a_{n-1} & c_{n-1} \\ 0 & \cdots & 0 & e_{n-1} & a_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{n-1} \\ b_n \end{bmatrix} \tag{1}$$

The function should be in the form

```
function x = tridiag_solver(e,a,c,b)
```

Input:

`e`=$[e_1\ e_2\ \cdots\ e_{n-1}]$ `a`=$[a_1\ a_2\ \cdots\ a_n]$ `c`=$[c_1\ c_2\ \cdots\ c_{n-1}]$ `b`=$[b_1\ b_2\ \cdots\ b_n]^T$

Output:

`x`: solution to the linear system (1) (column vector)

Exercise 2 Consider the system of nonlinear ordinary differential equations

$$\begin{cases} \dfrac{dy(t)}{dt} = f(y(t), t) \\ y(0) = y_0 \end{cases} \tag{2}$$

where $f : \mathbb{R}^n \times [0, T] \to \mathbb{R}^n$, $y : [0, T] \to \mathbb{R}^n$. Write a Matlab function that solves the system (2) by using the third-order Adams-Bashforth scheme. To start-up the scheme (first two steps) use the

Heun scheme. The function should be of the form

```
function [y,t] = AB3(fun,y0,T,DT,IOSTEP)
```

Input:
`fun`: function handle representing $f(y,t)$
`y0`: column vector representing the initial condition
`T`: period of integration
`DT`: time step
`IOSTEP`: Input/output step. The solution is saved in the output matrix `y` every `IOSTEP` steps.

Output:
`y`: $n \times S$ matrix collecting the time snapshots of the solution to (2). Note that the total number of snapshots S (including the initial condition) is `floor(T/(IOSTEP*DT))+1`.
`t`: vector collecting the time instants at which the solution is saved in the output matrix `y`.

Exercise 3 Consider the following nonlinear dynamical system

$$\begin{cases} \dfrac{dy_1(t)}{dt} = -y_1(t) + y_2(t)y_3(t) \\ \dfrac{dy_2(t)}{dt} = -y_2(t) + (y_3(t) - 2)y_1(t) \\ \dfrac{dy_3(t)}{dt} = 1 - y_1(t)y_2(t) \end{cases} \tag{3}$$

It is known that the solution to (3) is chaotic in time and it settles on a strange attractor. By using the function `AB3.m` you coded in Exercise 2, compute the numerical solution to (3). To this end, set `NSTEPS`=1000000, `IOSTEP`=50, `DT`= 1e-3, `y0`=$[1\ 2\ 3]^T$, and write a function

```
function [y,t]=solve_ODE_system()
```

Output:
`y`: $3\times S$ matrix collecting S time snapshots of the solution to (3). Here, `S=floor(NSTEPS/IOSTEP)+1`.
`t`: vector collecting the time instants at which the solution is saved in the output matrix `y`.

The function `solve_ODE_system` should also return the following items:

1. The plots of the trajectories $y_1(t)$, $y_2(t)$ and $y_3(t)$ versus time in `figure(1)`.
2. A three-dimensional plot of the curve $(y_1(t), y_2(t), y_3(t))$ in `figure(2)` (use the command `plot3()`).