Final Project, due May 5

In this final project, we consider to solve the following stiff first-order ordinary differential equation

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}' = \begin{bmatrix} -500.5 & 499.5 \\ 499.5 & -500.5 \end{bmatrix} \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix},$$

with initial condition $[y_1(0),\ y_2(0)] = [1,\ 3]$, by using forward Euler, backward Euler, trapezoid, Runge-Kutta, and the build-in Matlab ODE solvers. We know the exact solution of this problem is

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} 2e^{-t} - e^{-1000t} \\ 2e^{-t} + e^{-1000t} \end{bmatrix}.$$

1. Write Matlab functions to implement the forward Euler method, the backward Euler method, the trapezoid method, and the 4-stage 4th-order explicit Runge-Kutta method, respectively, to solve this ODE, on the time interval $t \in [0\ 1]$, with initial condition $[y_1(0),\ y_2(0)] = [1,\ 3]$.

Plot the approximate solutions and have them compared with the exact solution. Try different step size h. For each algorithm, is there any restrictions on h to avoid the oscillation in the numerical solutions? Which algorithm is the most restrictive on the choice of h?

2. Implement the Matlab built-in ODE solvers, ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb, to solve the same ODE.

Generate plots to show if all of those algorithms give accurate approximation of the exact solution.

To call these functions to solve a general ODE

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), \quad \mathbf{y}(t0) = \mathbf{y}_0,$$

on a time interval $[t0\ \ tf]$, you need to use

\>> [T, Y] = solver(odefun, tspan, y0);

Here `solver` is any one of those ode45, ode23, etc.; `odefun` is the name of the function which returns the right hand vector $\mathbf{f}(t, \mathbf{y})$ with input scalar t and input column vector $\mathbf{y}$; `tspan` is simply $[t0\ \ tf]$, the vector corresponding to the time interval; `y0` is the initial data vector $\mathbf{y}_0$. In the outputs, `T` is a column vector of time points; `Y` is the solution matrix with each row corresponding to the solution at a time returned in the corresponding row of `T`. More information can be found through the Matlab help.