Homework 3. Due Feb. 22

Please upload a single pdf file on ELMS. Link your codes to your pdf (i.e., put your codes to dropbox, Github, google drive, etc. and place links to them in your pdf file with your solutions.

1. **(5 pts)** Solve exercise 13 in my lecture notes `ODEsolvers.pdf`.

2. **(5 pts)** Solve exercise 14 in my lecture notes `ODEsolvers.pdf`.

3. **(6 pts)** There is a curious phenomenon called *order reduction* observed when DIRK methods are applied to stiff problems with time steps h such that $h\lambda$ where $-\lambda$ is the largest negative eigenvalue of the linear part of the right-hand side of an ODE is not small – see papers by B. Seibold's group https://arxiv.org/pdf/1712.00897.pdf and https://arxiv.org/pdf/1811.01285.pdf. The goal of this exercise is to examine the performance of DIRK methods of orders 2 and 3 from the previous exercises on the Prothero-Robinson problem, plot the maximum absolute error as a function of the time step, and observe the two orders of convergence for each method, one for large hL, and one for small hL. You need to plot graphs similar to those in Fig. 2 in https://arxiv.org/pdf/1811.01285.pdf.

   Consider the Prothero-Robinson problem

   $$y' = -L(y - \phi(t)) + \phi'(t), \quad y(0) = y_0 \tag{1}$$

   with $L = 10^4$ and $\phi(t) = \sin(t + \pi/4)$. Set the time interval $0 \le t \le T_{\max} = 10$. The exact solution to this problem is

   $$y = e^{-Lt}(y_0 - \phi(0)) + \phi(t). \tag{2}$$

   (a) Pick the initial condition $y(0) = \sin(\pi/4)$. Compute the numerical solution using $\mathsf{DIRK2}$ on the interval $[0, T_{\max}]$ with time step h for each h from the following set:

   $$h = 10^{-p}, \quad \text{where} \quad p \in \{1, 1+d, 1+2d, \dots, 6\},\ d = 5/24. \tag{3}$$

   Plot the numerical error $e(h) = \max_{0 \le t_n \le T_{\max}} |u_n - y(t_n)|$ vs h. Use the log-log scale. Observe error decay $e = C_1 h$ and $e = C_2 h^2$ for large and small values of h, respectively. For reference, plot lines with slopes 1 and 2, i.e., $e = C_1 h$ and $e = C_2 h^2$ where you need to choose C_1 and C_2 so that the plot looks nice. Do the same for the DIRK of order 3. What orders of error decay do you observe? Also, plot reference lines.

(b) Repeat the task with $y(0) = \sin(\pi/4) + 10$. You will obtain a bit puzzling set of graphs. To understand what is going on, plot $|e(t)|$ for each method where $e(t)$ is the difference between the numerical and the exact solutions for three values of h: $h = 10^{-1}$, $h = 10^{-2}$, and $h = 10^{-3}$. Set the log scale in the y-axis. Do so for $T_{\max} = 10$ and $T_{\max} = 1$.

(c) Summarize what you have learned about the behavior of the error for DIRK2 and DIRK of order 3 from the numerical experiments in this problem.

4. **(6 pts)** Consider stiff Robertson's problem from chemical kinetics that has become a popular test problem for ODE solvers:

$$\begin{cases} x' = -ax + byz \\ y' = ax - byz - cy^2 \\ z' = cy^2 \end{cases}, \quad x(0) = 1,\ y(0) = 0,\ z(0) = 0. \tag{4}$$

The parameters a, b, c are the reaction rates: $a = 0.04$, $b = 10^4$, $c = 3 \cdot 10^7$. Eq. (4) describes the concentrations of substances X, Y, and Z in an autocatalytic reaction system where the substance X is converted to the substance Z via an intermediate substance Y. The dependent variables x, y, and z represent concentrations of X, Y, and Z, respectively. The concentration of Y rapidly grows from zero to some finite value as soon as the reaction starts and then the concentrations of X and Y slowly decay to 0 while the concentration of Z slowly approaches 1.

My codes `RobertsonHW3.ipynb` and `RobertsonHW3.m` implement DIRK2 to time integrate Eq. (4) on the time interval $[0, 100]$ in Python and Matlab, respectively. You can do the tasks below in either programming language.

(a) Implement DIRKo3 of order 3 from Problem 2, with Butcher's array

$$\begin{array}{c|cc} \gamma & \gamma & 0 \\ 1-\gamma & 1-2\gamma & \gamma \\ \hline & 1/2 & 1/2 \end{array}$$

where $\gamma = 1/2 + \sqrt{3}/6$.

(b) Implement the two-step BDF method. Use DIRK2 for the first time step.

(c) Compute and plot the numerical solutions on $t \in [0, 100]$ for all three methods on the time interval $[0, 100]$ at $h = 10^{-3}$, $h = 10^{-2}$, and $h = 10^{-1}$. Measure the CPUtime in all cases. Also, plot the CPUtime vs h for all three methods. Use log scale for the CPUtime and for h in Figure 4.

Compare the performance of these three methods and write a summary of your observations.