Lab Session 7

MA581: Numerical Computations Lab — R. Alam — September 13, 2021

---

1. **Flame problem:** If you light a match, the ball of flame grows rapidly until it reaches a critical size. Then it remains at that size because the amount of oxygen being consumed by the combustion in the interior of the ball balances the amount available through the surface. The simple model is given by $y' = y^2 - y^3$, $y(0) = \delta$ and $0 \leq x \leq 2/\delta$. The function $y(x)$ represents the radius of the ball. The critical parameter is the initial radius δ, which is small. We seek the solution over a length of time that is inversely proportional to δ. If δ is not very small then the problem is not very stiff.

   Your task is to solve this problem using `ode45` with $\delta = .01$ and `RelTol` $= 10^{-4}$. For example, you can run the following

   ```
   delta = 0.01;
   F = inline('y^2 - y^3','x','y');
   opts = odeset('RelTol',1.e-4);
   ode45(F,[0 2/delta],delta,opts);
   ```

   With no output arguments, `ode45` automatically plots the solution as it is computed. You should get a plot of a solution that starts at $y = .01$, grows at a modestly increasing rate until x approaches 100, which is $1/\delta$, then grows rapidly until it reaches a value close to 1, where it remains.

   Now to see stiffness in action, decrease δ by a couple of orders of magnitude.

   ```
   delta = 0.0001;
   ode45(F,[0 2/delta],delta,opts);
   ```

   It will take a long time to complete the plot. You can click the stop button in the lower left corner of the window, turn on zoom, and use the mouse to explore the solution near where it first approaches steady state.

   If you want an even more dramatic demonstration of stiffness, decrease the tolerance `RelTol` to 10^{-5} or 10^{-6}.

   This problem is not stiff initially. It only becomes stiff as the solution approaches steady state. Any solution near $y(x) = 1$ increases or decreases rapidly toward that solution.

   Now change the method to stiff solver (stiff methods are implicit) `ode23s`:

   ```
   delta = 0.0001;
   ode23s(F,[0 2/delta],delta,opts);
   ```

   It will show the solution and the zoom detail. You can see that `ode23s` takes many fewer steps than `ode45`.

   The flame problem is also interesting because it involves something called the Lambert W function, which is the solution to the equation $W(z)e^{W(z)} = z$. The exact solution of the flame problem is given by $y(x) = \dfrac{1}{W(ae^{a-x}) + 1}$, where $a = 1/\delta - 1$.

   The MATLAB command

   ```
   y = dsolve('Dy = y^2 - y^3','y(0) = 1/100');
   y = simplify(y);
   pretty(y)
   ezplot(y,0,200)
   ```

produce the exact solution $y(x)$ in terms of Lambert W function given by

```
             1
----------------------------
lambertw(99 exp(99 - x)) + 1
```

and the plot of the exact solution $y(x)$. If the initial value $1/100$ is decreased and the time span $0 \leq x \leq 200$ increased then the transition region becomes narrower.

2. The error function $\mathrm{erf}(x)$ is defined $\mathrm{erf}(x) := \int_0^x e^{-t^2} dt$ but it can also be defined as the solution to the differential equation

$$y' = \frac{2}{\sqrt{\pi}} e^{-x^2}, \quad y(0) = 0.$$

Use the MATLAB function `ode23` to solve this differential equation on the interval $[0, 2]$. Compare the results with the built-in MATLAB function $\mathrm{erf}(x)$ at the points chosen by `ode23`.

3. The ODE problem $y' = -1000(y - \sin x) + \cos x, \quad y(0) = 1$ on the interval $[0, 1]$ is mildly stiff. Roughly speaking, a differential equation is stiff when certain numerical methods for solving the equation are numerically unstable, unless the step size is taken to be extremely small.

   (a) Compute the solution with `ode23`. How many steps are required? The MATLAB function `ode23` is a non-stiff ode solver.

   (b) Compute the solution with the MATLAB stiff ode solver `ode23s`. How many steps are required?

   (c) Plot the two computed solutions on the same graph, with line style '.' for the `ode23` solution and line style 'o' for the `ode23s` solution.

   (c) Zoom in, or change the axis settings, to show a portion of the graph where the solution is varying rapidly. You should see that both solvers are taking small steps.

   (d) Show a portion of the graph where the solution is varying slowly. You should see that `ode23` is taking much smaller steps than `ode23s`.

4. The problems $y' = \cos x, \ y(0) = 0$ and $y' = \sqrt{1 - y^2}, \ y(0) = 0$ have the same solution on the interval $[0, \pi/2]$. The MATLAB function `ode45` uses Runge-Kutta method to solve an ode.

   (a) Use `odeset` command to set both `reltol` and `abstol` to 10^{-9}. For example, consider `opts = odeset('abstol', 1e-9, 'reltol', 1e-9)` and use `opts` in the ode solver `ode45`. How much work does `ode45` require to solve each problem?

   (b) What happens to the computed solutions if the interval is changed to $[0, \pi]$?

   (c) What happens on the interval $[0, \pi]$ if the second problem is changed to $y' = \sqrt{|1 - y^2|}$, $y(0) = 0$?

*** End ***