CS 770: Assignment 6

Ronghao Yang
ID: 20511820

December 6, 2017

1 Exercise 1

The general form of RK method is:

$$y_n = y_{n-1} + h\sum_{i=1}^{s} b_i Ki$$

Where

$$K_i = f(y_{n-1} + h\sum_{j=1}^{s} a_{ij}K_j, t_{n-1} + c_i h)$$

For explicit two stage Runge-Kutta methods, we have

$$K_1 = f(y_{n-1}, t_{n-1})$$
$$K_2 = f(y_{n-1} + haK_1, t_{n-1} + ch)$$
$$y(t_n) = y_{n-1} + h(b_1K_1 + b_2K_2)$$

Using explicit two stage Runge-Kutta methods,

$$y_n = y(t_{n-1}) + h(b_1 f + b_2(f + ahff_y + chf_t) + O(h^2))$$

Compared with $y(t_n)$

$$y(t_n) = y(t_{n-1}) + hf + (f_y f + f_t)\tfrac{h^2}{2} + O(h^3)$$

When all the terms are matched, the error is smallest, we have:

$$b_1 + b_2 = 1,\ ab_2 = \tfrac{1}{2},\ cb_2 = \tfrac{1}{2}$$

Using b_2 to represent all variables, we have:

$$b_1 = 1 - b_2,\ a = \tfrac{1}{2b_2},\ c = \tfrac{1}{2b_2}$$

The local error is defined by $d_n = y(t_n) - y_n$, then after simplifying, only $O(h^3)$ term is left in d_n.
Therefore, no choice of coefficients in the one parameter family of the explicit two stage Runge-Kutta methods derived in class will result in the local error of order 4

2 Exercise 2

2.1 Exercise 2a

$$d_n = y(t_n) - y_n$$
$$d_n = y(t_n) - y(t_{n-1}) - h(\theta f(y_n) + (1-\theta)f(y_{n-1}))$$
$$d_n = y(t_n) - y(t_{n-1}) - h\theta f(y_n) - h(1-\theta)f(y_{n-1})$$

Using Taylor's expansion:

$$y(t_n) = y(t_{n-1}) + hy'(t_{n-1}) + \tfrac{h^2}{2}y''(t_{n-1}) + \tfrac{h^3}{6}y'''(t_{n-1}) + O(h^4)$$

$$y'(t_n) = y'(t_{n-1}) + hy''(t_{n-1}) + \frac{h}{2}y'''(t_{n-1}) + O(h^3)$$
$$d_n = (-\theta + \frac{1}{2})h^2 y''(t_{n-1}) + \frac{1}{2}(-\theta + \frac{1}{3})h^3 y'''(t_{n-1}) + O(h^4)$$

When $\theta = \frac{1}{2}$, d_n is the smallest, d_n is bounded to $O(h^3)$, otherwise d_n is bounded by $O(h^2)$

2.2 Exercise 2b

$$y(t_n) = y(t_{n-1}) + h(\theta f(y_n) + (1-\theta)f(y_{n-1}))$$
$$y(t_n) = y(t_{n-1}) + h(\lambda\theta y(t_n) + \lambda(1-\theta)y(t_{n-1}))$$
$$(1 - h\theta\lambda)y(t_n) = (1 + h\lambda(1-\theta))y(t_{n-1})$$
$$\frac{y(t_n)}{y(t_{n-1})} = \frac{1+(1-\theta)h\lambda}{1-\theta h\lambda}$$
$$\mid \frac{1+(1-\theta)h\lambda}{1-\theta h\lambda} \mid \leq 1$$

Let $z = h\lambda$. then

$$\mid \frac{1+(1-\theta)z}{1-\theta z} \mid \leq 1$$
$$\mid 1 + (1-\theta)z \mid \leq \mid 1 - \theta z \mid$$
$$\text{Let } z = a + bi$$

After simplifying, we get

$$2a \leq (a^2 + b^2)(2\theta - 1)$$

When $\theta \in [\frac{1}{2}, 1]$, the absolute stability region contains the whole left half plane of the complex plane

3 Exercise 3

3.1 Exercise 3a

```
function [tout,yout] = myrk4(F,tspan,y0,h,varargin)
%MYRK4  My own version of classical fourth order Runge-Kutta.
%   Usage is same as ODE23TX, except fourth argument is fixed step size, h.
%   MYRK4(F,TSPAN,Y0,H) with TSPAN = [T0 TFINAL] integrates the system
%   of differential equations y' = f(t,y) from t = T0 to t = TFINAL.
%   The initial condition is y(T0) = Y0.
%   With two output arguments, [T,Y] = MYRK4(...) returns a column
%   vector T and an array Y where Y(:,k) is the solution at T(k).
%   With no output arguments, MYRK4 plots the emerging solution.
%   More than four input arguments, MYRK4(F,TSPAN,Y0,H,P1,P2,...),
%   are passed on to F, F(T,Y,P1,P2,...).

   t0 = tspan(1);
   tfinal = tspan(2);
   t = t0;
   y = y0(:);
```

```
    tout = t;
    yout = y.';


    while t ~= tfinal

      s1 = F(t, y, varargin{:});

      if 1.1*abs(h) >= abs(tfinal - t)
        h = tfinal - t;
      end

      s2 = F(t+h/2, y+(h/2)*s1, varargin{:});
      s3 = F(t+h/2, y+(h/2)*s2, varargin{:});
      s4 = F(t+h, y+h*s3, varargin{:});

      tnew = t + h;
      ynew = y + h*(s1 + 2*s2 + 2*s3 + s4)/6;

      tout(end+1,1) = tnew;
      yout(end+1,:) = ynew.';

      t = tnew;
      y = ynew;
    end

end
```

3.2 Exercise 3b

When the size of the h is cut by half, the error goes down to $\frac{1}{16}$. The reason is the following. At step size h, the error is estimated by $e_h = Ch^4$. When h goes to $\frac{h}{2}$, we have $e_{\frac{h}{2}} = C(\frac{h}{2})^4 = C\frac{h^4}{16}$.
To illustrate this, let h = π and $\frac{\pi}{2}$. When h = π, the error is around 0.8169, when h = $\frac{\pi}{2}$, the error is around 0.0818 which is roughly about $\frac{1}{16}$ of 0.8169.

3.3 Exercise 3c

When having a system of equations, let v = y', u = y, then we have u' = v, v' = -u.
By running

ode23(@(t,y) [0 1;-1 0]*y , tspan, [1;0], pi/50)

And

ode45(@(t,y) [0 1;-1 0]*y , tspan, [1;0], pi/50)

Ode23 requires more steps than myrk4 and ode45 requires less step when the relative tolerance is set to be 1e-6. With relative tolerance being 1e-3, both methods require less steps than 100.