Problem 8.1  Heun Method

Let the Heun method:
\[ \begin{align*}
\tilde{y}_{n+1} &= y_n + hf(t_n, y_n), \\
y_{n+1} &= y_n + \frac{h}{2} [f(t_n, y_n) + f(t_{n+1}, \tilde{y}_{n+1})]
\end{align*} \]

(8.1a) Show that the method is consistent.

**Solution:** By definition,
\[ \Phi(t, y, h) := \frac{1}{2} (f(t_i, y_i) + f(t_i + h, y_i + hf(t_i, y_i))), \]
thus
\[ \Phi(t, y, 0) = \frac{1}{2} (f(t_i, y_i) + f(t_i, y_i)) = f(t_i, y_i), \]
which means that the method is consistent.

(8.1b) Implement in Matlab a program for the method using template `Heun.m`, and verify with `scriptheun.m` that the order of accuracy with respect to \( h \) for the problem is 2:

\[ \begin{align*}
y' &= \sin(t) + y, \quad t \in (0, 1] \\
y(0) &= 0
\end{align*} \]

**Solution:**

```matlab
Listing 8.1: Script of order problem with Heun method
1 % Heun Method Script
2 % Start time
3 t0=0;
4 
5 % Initial Value
6 y0=0;
7 
8 % End time
9 T=1;
10 
11 f=inline('sin(t)+y','t','y','y');
```

Problem Sheet 8  Page 1  Problem 8.1
% Exact solution
y=inline('0.5*(exp(t)-sin(t)-cos(t))','t');

% Here we apply 2^1, ..., 2^10 steps on time interval [0,1]
% and record the
% errors at time t=1.
Nh=2;
for k=1:10
    [tt,u]=heun(f,[t0,T],y0,Nh);
    e(k)=abs(u(end)-feval(y,tt(end)));
    Nh=2*Nh;
end

% Output of order
p=log(abs(e(1:end-1)./e(2:end)))/log(2);
p(1:2:end)

Listing 8.2: Heun Method

% Implement of Heun Method
% Odefun: function f on RHS of IVP
% Tspan: Time span
% y0: Initial value
% Nh: Number of steps

function [tt,u]=heun(odefun,tspan,y0,Nh)

    % Set up Time grid
    tt=linspace(tspan(1),tspan(2),Nh+1);

    % Step size
    h=(tspan(2)-tspan(1))/Nh;
    hh=h*0.5;

    % Initial value
    u=y0;

    % Iteration
    for t=tt(1:end-1)
        y=u(end,:);
        k1=feval(odefun,t,y);
        t1=t+h;
        y=y+h*k1;
    end
You can find another version in folder Problem 1_alternative.

### Problem 8.2 One Example of Runge-Kutta Method

Let us consider the following Runge-Kutta method:

\[
y^{n+1} = y^n + h\left(\frac{1}{6}k_1 + \frac{2}{3}k_2 + \frac{1}{6}k_3\right)
\]

- \(k_1 = f(t_n, y_n)\)
- \(k_2 = f(t_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1)\)
- \(k_3 = f(t_n + h, y_n - hk_1 + 2hk_2)\)

**(8.2a)** Show that the method is consistent.

**Solution:** Compare to the formlular (4.55) in lecture notes, we find that \(b_1 = b_3 = \frac{1}{6}, b_2 = \frac{2}{3}\) in this method. Since \(b_1 + b_2 + b_3 = 1\), by section 4.4.3, we learn that this Runge-Kutta scheme is consistent.

**(8.2b)** Implement in Matlab a program for the method using template `rk3.m` and verify with `RKmethodscript.m` the order of accuracy is 3 with respect to \(h\) for the problem

\[
y' = \sin(t) + y, t \in (0, 1]
\]

\(y(0) = 0\)

**Solution:**

```matlab
% Script of RK method

% Start time
t0=0;

% Initial Value
y0=0;

% End time
T=1;

f=inline(’sin(t)+y’,’t’,’y’);

% Exact solution
y=inline(’0.5*(exp(t)-sin(t)-cos(t))’,’t’);
```

Listing 8.3: Script of order problem with RK Method
% Here we apply $2^1, \ldots, 2^{10}$ steps on time interval $[0,1]$
% and record the
% errors at time $t=1$.
Nh=2;
for k=1:10
    [tt,u]=rk3(f,[t0,T],y0,Nh);
e(k)=abs(u(end)-feval(y,tt(end)));
    Nh=2*Nh;
end

% Output of order
p=log(abs(e(1:end-1)./e(2:end)))/log(2);
p(1:2:end)

Listing 8.4: RK Method

% Implement of RK Method
%
% Odefun: function $f$ on RHS of IVP
% Tspan: Time span
% y0: Initial value
% Nh: Number of steps
function [tt,u]=rk3(odefun,tspan,y0,Nh)
    % Set up Time grid
    tt=linspace(tspan(1),tspan(2),Nh+1);

    % Step size
    h=(tspan(2)-tspan(1))/Nh;
    hh=h*0.5;
    h2=2*h;

    % Initial value
    u=y0;
    h6=h/6;

    % Iteration
    for t=tt(1:end-1)
        y=u(end,:);
        k1=feval(odefun,t,y);
        t1=t+hh;
        y1=y+hh*k1;
        k2=feval(odefun,t1,y1);
        t1=t+h;
        y1=y+h*(2*k2-k1);
        k3=feval(odefun,t1,y1);
        [tt,u]=rk3(f,[t0,T],y0,Nh);
e(k)=abs(u(end)-feval(y,tt(end)));
    end

    Nh=2*Nh;
end

% Output of order
p=log(abs(e(1:end-1)./e(2:end)))/log(2);
p(1:2:end)
Problem 8.3  Vibrant system

The displacement \( y(t) \) of a vibrant system is given by a body with a certain weight and a spring and is subject to a resistive force proportional with the velocity. This can be described by the differential equation \( y'' + 5y' + 6y = 0 \).

Using the Heun method implemented in the Problem 1, solve and plot the solution of the differential equation for \( y(0) = 1, y'(0) = 0, t \in [0, 5], h_n = 0.3 \).

**Hint:** Write the equation in the form of 1st order ODE and implement in MATLAB using template ex3.m. Please copy Heun.m in Problem 1 to the same directory as ex3.m, so that the function can be called.

**Solution:** Let \( x = [y, y']^T \). Then we have

\[
\frac{d}{dt}x = \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix}x
\]

and \( x(0) = [1, 0]^T \). See the attached code for implementatio of the Heun method.

---

```matlab
% Script for Vibrant system

% Initial Time
t0=0;

% Initial Value
y0=[1 0];

% End time
T=5;

% Call heun method function here
[t,u]=heun(@fmolle,[t0,T],y0,floor(5/0.3));

% Plot
figure;
plot(u,'red');

% Define function f here
function fn=fmolle(t,y)
    b=5;
    k=6;
    [n,m]=size(y);
```

---
fn = zeros(n, m);
fn(1) = y(2);
fn(2) = -b*y(2) - k*y(1);
end