Assignment 8

1. Using Heun's method, approximate $y(1)$ with $h = 0.2$ and again with $h = 0.1$ for the initial-value problem defined by

$$y^{(1)}(t) = 2y(t) + t - 1$$
$$y(0) = 1$$

```
f = @(t, y)(2*y + t - 1);
h = 0.2;
ts = 0:h:1;
ys = zeros( 1, 6 );
ys(1) = 1.0;
for k = 1:5
    s0 = f( ts(k), ys(k) );
    s1 = f( ts(k) + h, ys(k) + h*s0 );
    ys(k + 1) = ys(k) + h*(s0 + s1)/2.0;
end
ys
    ys =
         1.0    1.26    1.6928    2.381344    3.44838912    5.0756158976

h = 0.1;
tss = 0:h:1;
yss = zeros( 1, 11 );
yss(1) = 1.0;
for k = 1:10
    s0 = f( tss(k), yss(k) );
    s1 = f( tss(k) + h, yss(k) + h*s0 );
    yss(k + 1) = yss(k) + h*(s0 + s1)/2.0;
end
yss
    yss = 1.0                 1.115               1.2663
          1.461886            1.71150092          2.0270311224
          2.422977969328      2.917033122580161   3.530780409547796
          4.290552099648311   5.228473561570939
```

2. In Question 1, you approximated $y(0.2)$ with $h = 0.2$, and $y(0.1)$ with $h = 0.1$. The correct solutions to sixteen significant digits are $y(0.2) = 1.268868523230952$ and $y(0.1) = 1.116052068620128$. Show that the error of one step of Heun's method is $O(h^3)$ by showing that the error of your approximation at $t = 0.1$ is approximately one eighth the error at $t = 0.2$. You should consider doing one step by hand, and then multiple steps using a program.

```
% Error of one step with h = 0.2
abs( 1.268868523230952 - 1.26 )
        ans = 0.008868523230951997

% Error of one step with h = 0.1
abs( 1.116052068620128 - 1.115 )
        ans = 0.001052068620128077

% Error of one step with h = 0.2 divided by eight
abs( 1.268868523230952 - 1.26 )/8
        ans = 0.001108565403869000
```

As you can see, the error is indeed reduced by approximately a factor of eight.

3. Using the 4th-order Runge-Kutta method, approximate $y(1)$ with $h = 0.2$ and again with $h = 0.1$ for the initial-value problem defined by

$$y^{(1)}(t) = 2y(t) + t - 1$$
$$y(0) = 1$$

```
h = 0.2;
ts = 0:h:1;
ys = zeros( 1, 6 );
ys(1) = 1.0;
for k = 1:5
    s0 = f( ts(k), ys(k) );
    s1 = f( ts(k) + 0.5*h, ys(k) + 0.5*h*s0 );
    s2 = f( ts(k) + 0.5*h, ys(k) + 0.5*h*s1 );
    s3 = f( ts(k) + h, ys(k) + h*s2 );
    ys(k + 1) = ys(k) + h*(s0 + 2*s1 + 2*s2 + s3)/6.0;
end
ys
    ys = 1.000000000000000   1.268800000000000   1.718951253333334
         2.439630216305778   3.563864381337206   5.290095293120089

h = 0.1;
tss = 0:h:1;
yss = zeros( 1, 11 );
yss(1) = 1.0;
for k = 1:10
    s0 = f( tss(k), yss(k) );
    s1 = f( tss(k) + 0.5*h, yss(k) + 0.5*h*s0 );
    s2 = f( tss(k) + 0.5*h, yss(k) + 0.5*h*s1 );
    s3 = f( tss(k) + h, yss(k) + h*s2 );
    yss(k + 1) = yss(k) + h*(s0 + 2*s1 + 2*s2 + s3)/6.0;
end
yss
   yss = 1.000000000000000   1.116050000000000   1.268863470000000
         1.466579842258000   1.719140619333921   2.038688352454451
         2.440053953687867   2.941351899034361   3.564707209480569
         4.337143385659567   5.291666931244595
```

4. In Question 1, you approximated $y(0.2)$ with $h = 0.2$, and $y(0.1)$ with $h = 0.1$. The correct solutions to sixteen significant digits are $y(0.2) = 1.268868523230952$ and $y(0.1) = 1.116052068620128$. Show that the error of one step of 4th-order Runge-Kutta method is $O(h^5)$ by showing that the error of your approximation at $t = 0.1$ is approximately one thirty-second the error at $t = 0.2$. You should consider doing one step by hand, and then multiple steps using a program.

```
% Error of one step with h = 0.2
abs( 1.268868523230952 - 1.2688 )
       ans = 0.00006852323095207780

% Error of one step with h = 0.1
abs( 1.116052068620128 - 1.11605 )
       ans = 0.00002068620128081733

% Error of one step with h = 0.2 divided by 32
abs( 1.268868523230952 - 1.2688 )/32.0
       ans = 0.00002141350967252431
```

5. When we do the error analysis on Euler's method, are we using a convex combination of the errors, or a more general weighted average of the errors? Thus, may we say that the error is bounded above and below by the second derivative evaluated at some point on the interval, or may this not be the case?

It is a convex combination of the errors.

Therefore, if the second derivative is always positive, then the error will always be positive, for the approximation will always underestimate the next approximation.

Similarly, if the second derivative is always negative, then the error will always be negative, for the approximation will always overestimate the next approximation.

6. For argument's sake, how small would h have to be so that the error is less than 0.01 using Euler's method to find the approximation to the initial-value problem

$$y^{(1)}(t) = -y(t)$$
$$y(0) = 1$$

to approximate the value of $y(10)$. You can find h because you know the solution to this initial-value problem and you can calculate an upper bound for the solution's second derivative on the interval [0, 10].

Taylor's series is $y(h) = y(0) + y^{(1)}(0) + \frac{1}{2}y^{(2)}(\tau)h^2$ for one step, and the error for multiple steps is

$$\frac{1}{2}(b - a)hy^{(2)}(\tau)$$

so for this problem, as $y^{(1)}(t) = -y(t)$ so $y^{(2)}(t) = -y^{(1)}(t)$ and thus $y^{(2)}(t) = y(t)$, and this simplifies to $5hy(\tau)$ for some value of $0 \le \tau \le 10$. The solution to the IVP with this ODE and the initial condition $y(0) = 0$ is the constant zero function so a lower bound, and $y^{(1)}(0) = -1$, so we may conclude that $0 \le y(\tau) \le 1$, so as an over-estimate, we want $5h = 0.01$ or $h = 0.002$.

7. Apply one step of our adaptive Euler-Heun to approximate a solution to the initial-value problem

$$y^{(1)}(t) = 2y(t) + t - 1$$
$$y(0) = 1$$

starting with an $h = 0.1$ with the per unit time error $e_{abs} = 0.1$. What value of h would you use with the next step, and would you be recalculating the previous step, or calculating the next step?

```
f = @(t, y)(2*y + t - 1);
t0 = 0;
y0 = 1;
eps_abs = 0.1;
h = 0.1;
s0 = f(t0,     y0);
s1 = f(t0 + h, y0 + h*s0);
y = y0 + h*s0
        y = 1.1
z = y0 + h*(s0 + s1)/2
        z = 1.115
a = (h*eps_abs)/(2*abs(y - z))
        a = 0.333333333333333
h = 0.9*a*h
        h = 0.03
```

We would be required to recalculate but now with a value of $h = 0.03$, not $h = 0.1$; however, as a precaution, we do not divide h by more than a factor of two, so we should use $h = 0.05$.

> As an aside, it happens that $h = 0.05$ is still too large, and we find (again) that the optimal value of h should be $h = 0.03$, at which point, the value of a is 1.111111111, and $t_1 = 0.03$ and $y_1 = 1.03135$ and so with our next step, we would again use $h = 0.9ah = 0.03$.

8. Suppose you were applying the Dormand-Prince method and you started with $h = 0.1$ with a per unit time acceptable error of $\varepsilon_{abs} = 0.00001$, and your two approximations of the next point were

$$y = 1.1160522588 \text{ and } z = 1.11605208.$$

What would your value of a be in this case, and what step size would you use with the next step? Would you be recalculating this point with the same h value, or would you continue to approximate the next point?

```
h = 0.1;
eps_step = 0.00001;
y = 1.1160522588;
z = 1.11605208;
a = ( (h*eps_step)/(2*abs(y - z)) )^0.25
          a = 1.293155116610072
h = 0.9*a*h
          h = 0.116383960494907
```

We would thus use the value z to approximate the solution at $t_{k+1} \leftarrow t_k + 0.1$, so $y_{k+1} \leftarrow z$ for the above value of z, and the step size with the next calculation would be the value of h found above.

9. Incidentally, the approximations y and z are the approximations to the solution to the initial-value problem given in Question 7 approximating the solution at $y(0.1)$. Given that the exact solution to sixteen significant digits is $y(0.1) = 1.116052068620128$, demonstrate that the error introduced is indeed less than $h\varepsilon_{\text{abs}}$.

```
h = 0.1;
eps_abs = 0.00001;
h*eps_abs
            ans = 0.000001
abs( z - 1.116052068620128 )
            ans = 0.00000001137987193366996
```