Homework 8
Jingmin Sun
661849071

1. (a)

$$
\begin{aligned}
I &= \int_1^4 \ln(x)dx \\
&= x\ln x|_{x=1}^4 - \int_{x=1}^4 x d\ln(x) \\
&= x\ln x|_{x=1}^4 - \int_{x=1}^4 x\frac{1}{x}dx \\
&= 4\ln 4 - \ln 1 - (4-1) \\
&= 4\ln 4 - 3
\end{aligned}
$$

(b)

Listing 1: two_point_Gaussian.m

```
nvec = 2.^(1:4);
nvec = nvec';
mid = @(a,b) (a+b)/2;
first_point = @(a,b) mid(a,b)-(b-a)/(2*sqrt(3));
second_point = @(a,b) mid(a,b)+(b-a)/(2*sqrt(3));
f = @(x) log(x);
I_2point = @(a,b) f(first_point(a,b))*(b-a)/2 + f(second_point(a,b))*(b-a)/2;
real_I = 4*log(4) - 3;
error = @(x) abs(x - real_I);
I_2vec = zeros(4,1);
errorvec = zeros(4,1);
reduc_fac = zeros(4,1);
for i=1:4
    n = nvec(i);
    avec = 1+3/n*(0:n-1);
    bvec = 1+3/n*(1:n);
    I = 0;
    for j = 1:n
      I = I + I_2point(avec(j), bvec(j));
    end
    I_2vec(i) = I;
    errorvec(i) = error(I);
    if i~=1
     reduc_fac(i) = errorvec(i-1)/errorvec(i);
    end
end
T = table(nvec,I_2vec, errorvec,reduc_fac);
```

and the output table is

Listing 2: output

```
T

T =

  4 4 table
```

```
    nvec        I_2vec              errorvec             reduc_fac
    ----    ----------------  --------------------  ----------------

      2     2.54658880527456   0.00141136079499438                 0
      4     2.54529878064998  0.000121336170421316  11.6318224820654
      8     2.54518601309424  8.56861468268022e-06  14.1605352690877
     16     2.54517800029212  5.55812556868318e-07  15.4163747774239
```

Listing 3: three_point_Gaussian.m

(c)

```
nvec = 2.^(1:4);
nvec = nvec';
mid = @(a,b) (a+b)/2;
first_point = @(a,b) mid(a,b)-(b-a)*sqrt(3)/(2*sqrt(5));
second_point = @(a,b) mid(a,b);
third_point = @(a,b) mid(a,b)+(b-a)*sqrt(3)/(2*sqrt(5));
f = @(x) log(x);
I_3point = @(a,b) 5/9*f(first_point(a,b))*(b-a)/2 + ...
    8/9*f(second_point(a,b))*(b-a)/2 + 5/9*f(third_point(a,b))*(b-a)/2;
real_I = 4*log(4) - 3;
error = @(x) abs(x - real_I);
I_3vec = zeros(4,1);
errorvec = zeros(4,1);
reduc_fac = zeros(4,1);
for i=1:4
    n = nvec(i);
    avec = 1+3/n*(0:n-1);
    bvec = 1+3/n*(1:n);
    I = 0;
    for j = 1:n
      I = I + I_3point(avec(j), bvec(j));
    end
    I_3vec(i) = I;
    errorvec(i) = error(I);
    if i~=1
     reduc_fac(i) = errorvec(i-1)/errorvec(i);
    end
end
T = table(nvec,I_3vec, errorvec,reduc_fac);
```

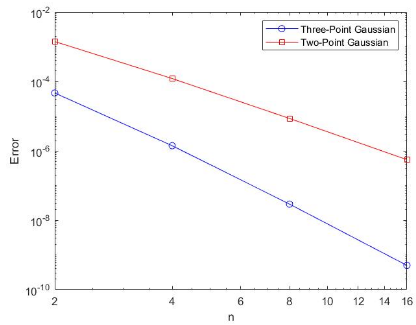
and the output table is

Listing 4: output

```
three_point_Gaussian

T

T =

  4 4 table

    nvec        I_3vec              errorvec             reduc_fac
    ----    ----------------  --------------------  ----------------
```

```
  2     2.54522391036974   4.6465890180869e-05                 0
  4     2.54517884892856  1.40444900109671e-06  33.0847828184467
  8     2.54517747358396  2.91043988909223e-08  48.2555577375198
 16      2.5451774449787  4.99133179232558e-10  58.3098862224943
```

(d) From the above analysis, we found that when we increase the number of mesh, which decrease the interval of each mesh at the same time, increases the accuracy of the approximation, and when we use more point on a mesh, we will get a more accurate approximation as well. To see this more clearly, we can draw the graph of error term:



Beside that, we can see that the three point Gaussian will converge faster, which means generating an error of higher order. To get the numerical order, we can compute the base 2 log of reduction factor of each method, which is the slope of two plotted lines (without negative sign)

Listing 5: error

```
two_point_Gaussian
log2(reduc_fac)

ans =

                -Inf
   3.540005251874583
   3.823803895298594
   3.946391644477345


three_point_Gaussian
log2(reduc_fac)

ans =

                -Inf
   5.048095904054325
   5.592623206358342
   5.865668602584057
```

So, we can see that this example perform an error greater than third order when $n = 1$ (2-point), and greater than fifth order when $n = 2$ (3-point). This is related to the d.o.p of these two method, which is $2n + 1$, and this example performs better on both of methods than expected.

2.

$$
\begin{aligned}
y_{j+1} &= y_j + h\left(\theta f(t_j, y_j) + (1-\theta) f(t_{j+1}, y_{j+1})\right) \\
&= y_j + h\theta f(t_j, y_j) + h(1-\theta) f(t_{j+1}, y_{j+1}) \\
&= y_j + h\theta f(t_j, y_j) + h f(t_{j+1}, y_{j+1}) - h\theta f(t_{j+1}, y_{j+1}) \\
&= y_j + h\theta r y_j + h r y_{j+1} - h\theta r y_{j+1} \\
&= (1 + h\theta r) y_j + (hr - hr\theta) y_{j+1} \\
(1 + hr\theta - hr) y_{j+1} &= (1 + h\theta r) y_j \\
y_{j+1} &= \frac{1 + hr\theta}{1 + hr\theta - hr} y_j \\
\therefore Q(rh, \theta) &= \frac{1 + rh\theta}{1 + rh\theta - rh}
\end{aligned}
$$

$$
\begin{aligned}
\left| \frac{1 + rh\theta}{1 + rh\theta - rh} \right| &\leq 1 \\
-1 \leq \frac{1 + rh\theta}{1 + rh\theta - rh} &\leq 1 \\
-1 \leq 1 + \frac{rh}{1 + rh\theta - rh} &\leq 1 \\
-2 \leq \frac{rh}{1 + rh\theta - rh} &\leq 0 \\
\begin{cases} 1 + rh\theta - rh & \geq 0 \\ \dfrac{-2}{rh} \geq \dfrac{1}{1 + rh\theta - rh} & \end{cases}
\end{aligned}
$$

From the first equation above we can get $-(1 + rh\theta - rh) < 1 + rh\theta \leq 1 + rh\theta - rh$, which lead the same equation of the second equation above, and from the second equation above, we can get $\frac{rh}{-2} \leq 1 + rh\theta - rh$, $rh \geq -2 - 2rh\theta + 2rh$, $rh(1 - 2\theta) \leq 2$,

- $\theta \in [0, 0.5)$:

$$
\begin{aligned}
\because 1 - 2\theta &> 0 \\
rh &\leq \frac{2}{1 - 2\theta} \\
h &\geq \frac{2}{r(1 - 2\theta)} \\
\because r < 0, 1 - 2\theta &> 0 \\
\therefore \frac{2}{r(1 - 2\theta)} &< 0 \\
\text{Together with } h &> 0 \\
\therefore h &> 0
\end{aligned}
$$

- $\theta = 0.5$

$$\begin{aligned} &h \text{ unbounded} \\ &\text{Together with } h > 0 \\ &\therefore h > 0 \end{aligned}$$

- $\theta \in (0.5, 1]$

$$\begin{aligned} &\because r < 0, 1 - 2\theta < 0 \\ &\therefore h \leq \frac{2}{r(1-2\theta)} \\ &\frac{2}{r(1-2\theta)} \geq 0 \\ &\therefore 0 < h \leq \frac{2}{r(1-2\theta)} \end{aligned}$$

In conclusion, when $\theta \in [0, 0.5]$, $h > 0$, and when $\theta \in (0.5, 1]$, $0 < h \leq \dfrac{2}{r(1-2\theta)}$

3. (a)

$$\begin{aligned} y' &= 2(t+1)y \\ \frac{dy}{dt} &= 2(t+1)y \\ \frac{dy}{y} &= 2(t+1)dt \\ \int \frac{dy}{y} &= \int 2(t+1)dt \\ \ln(y) &= t^2 + 2t + c \\ \because y(0) &= 1 \\ \ln(1) &= 0 + 0 + c \\ c &= 0 \\ \therefore \ln(y) &= t^2 + 2t \\ y &= e^{t^2+2t} \end{aligned}$$

$$
\begin{aligned}
\frac{dy}{dt} &= \frac{1}{y^2} \\
y^2 dy &= dt \\
\int y^2 dy &= \int dt \\
\frac{y^3}{3} &= t + c \\
\because y(0) &= 1 \\
\frac{1}{3} &= 0 + c \\
c &= \frac{1}{3} \\
\therefore \frac{y^3}{3} &= t + \frac{1}{3} \\
y^3 &= 3t + 1 \\
y &= \sqrt[3]{3t+1}
\end{aligned}
$$

(b) In this function, we just create three vectors and update its value in each iteration of a loop, and the code is:

Listing 6: FE1.m

```
h = 0.1;
y1prime = @(t,y) 2*(t+1)*y;
y2prime = @(t,y) 1/y^2;
y1real = @(t) exp(t^2+2*t);
y2real = @(t) (3*t+1)^(1/3);

num_int = 1/h;
tvec = 0 + (0:num_int)*h;
tvec = tvec';
y1vec = ones(num_int+1,1);
errvec1 = zeros(num_int+1,1);
errvec1(1) = abs(y1vec(1) - y1real(0));
y2vec = ones(num_int+1,1);
errvec2 = zeros(num_int+1,1);
errvec2(1) = abs(y2vec(1) - y2real(0));
for i = 1:num_int
   y1vec(i+1) = y1vec(i) + h * y1prime(tvec(i),y1vec(i));
   errvec1(i+1) = abs(y1vec(i+1) - y1real(tvec(i+1)));
   y2vec(i+1) = y2vec(i) + h * y2prime(tvec(i),y2vec(i));
   errvec2(i+1) = abs(y2vec(i+1) - y2real(tvec(i+1)));
end
T1 = table(tvec,y1vec,errvec1)
T2 = table(tvec,y2vec,errvec2)
```

and we can get the table

Listing 7: output

```
FE1
```

```
T1 =

  11 3  table

    tvec        y1vec              errvec1
    ----    ----------------  ------------------

      0                1                    0
    0.1              1.2    0.0336780599567434
    0.2            1.464     0.088707218511336
    0.3          1.81536     0.178355533243083
    0.4        2.2873536     0.324342873423118
    0.5      2.927812608     0.562530349461841
    0.6     3.8061563904     0.952664854737854
    0.7   5.024126435328      1.59524224571508
    0.8 6.73232942333952      2.66100186410326
    0.9 9.15596801574175      4.44308283608918
      1 12.6352358617236      7.45030106146405


T2 =

  11 3  table

    tvec        y2vec               errvec2
    ----    ----------------  -------------------

      0                1                     0
    0.1              1.1   0.00860711693889415
    0.2 1.18264462809917    0.0130375328140271
    0.3 1.25414222962591    0.0155798999957433
    0.4  1.3177201644182    0.0171287175668173
    0.5 1.37531103138972    0.0181022230922636
    0.6 1.42817967240341    0.0187199259904354
    0.7 1.47720655793524    0.0191068221085242
    0.8 1.52303314833857    0.0193385521335918
    0.9 1.56614347250788    0.0194630987358464
      1 1.60691311569235    0.0195120637241524
```

where T1 is the table of $y' = 2(t+1)y$, and T2 is the table of $y' = \frac{1}{y^2}$

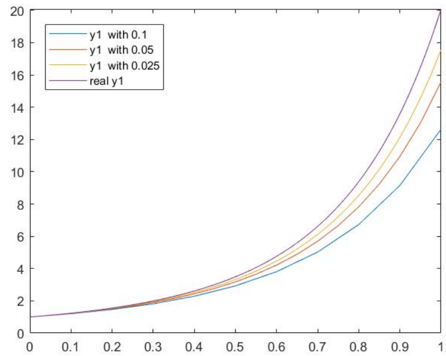
(c) To draw the graph, we can just separate it into two graph, one for each function, and add a loop to the previously code with different h.

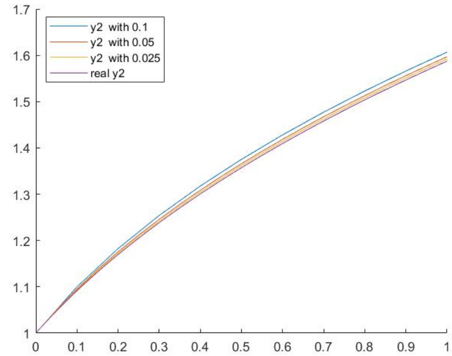
Listing 8: FE2.m

```
hvec = [0.1,0.05,0.025];
y1prime = @(t,y) 2*(t+1)*y;
y2prime = @(t,y) 1/y^2;
y1real = @(t) exp(t^2+2*t);
y2real = @(t) (3*t+1)^(1/3);
t = linspace(0,1,200);
y1realvec = ones(200,1);
for i = 1:200
    y1realvec(i) = y1real(t(i));
end
y2realvec = ones(200,1);
```

```
for i = 1:200
    y2realvec(i) = y2real(t(i));
end
figure(1);
legend1 = [];
for h = hvec
    num_int = 1/h;
    tvec = 0 + (0:num_int)*h;
    tvec = tvec';
    y1vec = ones(num_int+1,1);
    errvec1 = zeros(num_int+1,1);
    errvec1(1) = abs(y1vec(1) - y1real(0));
    for i = 1:num_int
        y1vec(i+1) = y1vec(i) + h * y1prime(tvec(i),y1vec(i));
        errvec1(i+1) = abs(y1vec(i+1) - y1real(tvec(i+1)));
    end
    T1 = table(tvec,y1vec,errvec1);
    plot(tvec,y1vec);
    legend1 = [legend1, "y1 with " + h];
    hold all;
end
 plot(t,y1realvec);
 legend( [legend1, "real y1"] );


figure(2);
legend2 = [];
for h = hvec
    num_int = 1/h;
    tvec = 0 + (0:num_int)*h;
    tvec = tvec';
    y2vec = ones(num_int+1,1);
    errvec2 = zeros(num_int+1,1);
    errvec2(1) = abs(y2vec(1) - y2real(0));
    for i = 1:num_int
        y2vec(i+1) = y2vec(i) + h * y2prime(tvec(i),y2vec(i));
        errvec2(i+1) = abs(y2vec(i+1) - y2real(tvec(i+1)));
    end
    T2 = table(tvec,y2vec,errvec2);
    hold on
    plot(tvec, y2vec);
    legend2 = [legend2, "y2 with " + h];
end
 plot(t,y2realvec);
 legend( [legend2, "real y2"] );
hold off
```

The graph of the first equation on the left, and that of the second equation on the right:



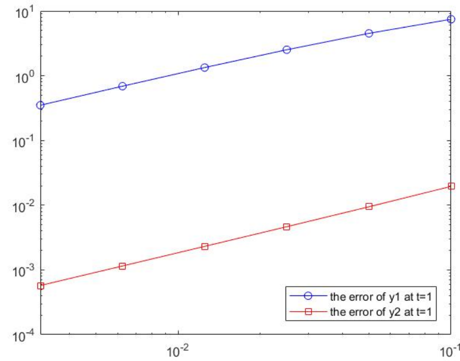


(d) To plot the error of the function value at $t = 1$, we just need to update each y in ecey iteration, instead of store it, and the code is:

Listing 9: FE3_loglog.m

```
hvec = 0.1*2.^(-(0:5));
y1prime = @(t,y) 2*(t+1)*y;
y2prime = @(t,y) 1/y^2;
y1real = @(t) exp(t^2+2*t);
y2real = @(t) (3*t+1)^(1/3);
error1vec = [];
error2vec = [];
red_fac1 = [];
red_fac2 = [];
for h = hvec
    num_int = 1/h;
    tvec = 0 + (0:num_int)*h;
    tvec = tvec';
    y1= 1;
    y2= 1;
    for i = 1:num_int
        y1 = y1 + h * y1prime(tvec(i),y1);
        y2 = y2 + h * y2prime(tvec(i),y2);
    end
    error1 = abs(y1 - y1real(1));
    error2 = abs(y2 - y2real(1));
    if length(error1vec) > 1
        red_fac1 = [red_fac1; error1/error1vec(length(error1vec)) ];
        red_fac2 = [red_fac2; error2/error2vec(length(error1vec)) ];
    end
    error1vec = [error1vec, error1];
    error2vec = [error2vec, error2];

end
 loglog(hvec,error1vec,'bo-');
 hold on
 loglog(hvec,error2vec,'rs-');
 legend( "the error of y1 at t=1", "the error of y2 at t=1" );
```

and we can see the graph:



We can observe that they have the same slope, and this is consistence with that FE has convergence scheme of order 1.

(e) Since for trapzoid method, we have

$$y_{j+1} = y_j + h(\frac{1}{2}f(t_j, y_j) + \frac{1}{2}f(t_{j+1}, y_{j+1}))$$

$$y_{j+1} - \frac{h}{2}f(t_{j+1}, y_{j+1}) = y_j + \frac{h}{2}f(t_j, y_j)$$

So for the first equation:

$$\begin{aligned} y_{j+1} - \frac{h}{2}(2(t_{j+1}+1)y_{j+1}) &= y_j + \frac{h}{2}(2(t_j+1)y_j) \\ y_{j+1} - h(t_{j+1}+1)y_{j+1} &= y_j + h(t_j+1)y_j \\ y_{j+1} &= \frac{1+h(t_j+1)}{1-h(t_{j+1}+1)}y_j \end{aligned}$$

So for the second equation:

$$\begin{aligned} y_{j+1} - \frac{h}{2}(\frac{1}{y_{j+1}^2}) &= y_j + \frac{h}{2}(\frac{1}{y_j^2}) \\ \frac{2y_{j+1}^3 - h}{2y_{j+1}^2} &= \frac{2y_j^3 + h}{2y_j^2} \end{aligned}$$

And for this one, we can use matlab solve function to get the solution, specifically, it's

```
y2 = y2vec(i);
y2vec(i+1) = fzero(@(y)(2*y^3-h)/(2*y^2)-(2*y2^3+h)/(2*y2^2),1);
```

And the entire code is attached in the end of this homework, and the table is:

Listing 10: output table

```
Trap

T1 =

  11 3 table

    tvec         y1vec                  errvec1
```

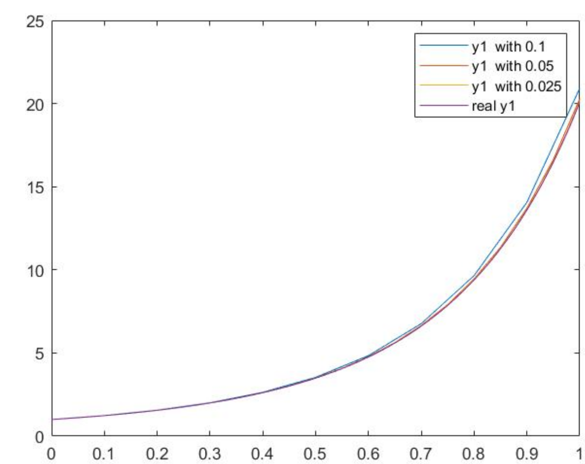
0	1	0
0.1	1.23595505617978	0.00227699622303201
0.2	1.55898876404494	0.00628154553360805
0.3	2.00697404106935	0.0132585078262706
0.4	2.63707054233531	0.0253740689121948
0.5	3.53677696266148	0.0464340051996364
0.6	4.84201607983417	0.0831948346963109
0.7	6.76715500314173	0.14778632209865
0.8	9.65557482155588	0.262243534113095
0.9	14.0661460363407	0.467095184509731
1	20.9233922290567	0.837855305869066

```
T2 =

  11 3 table
```

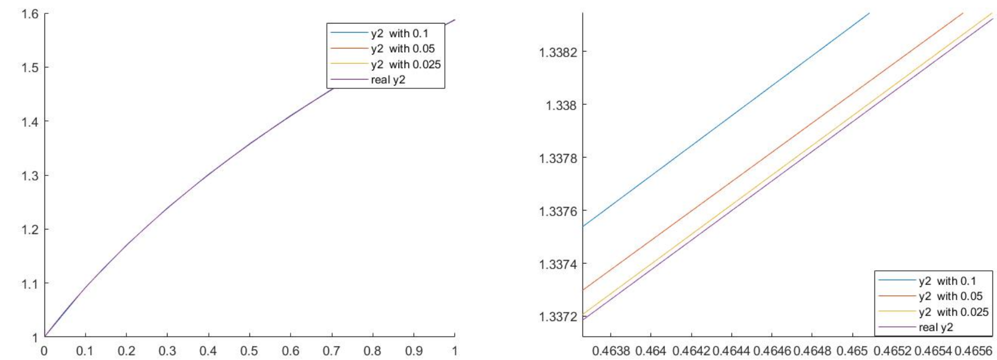
tvec	y2vec	errvec2
0	1	0
0.1	1.09193498058686	0.0005420975257584
0.2	1.170372398364	0.000765303078849033
0.3	1.23942331334555	0.000860983715383457
0.4	1.30148989153893	0.00089844468754019
0.5	1.35811591998404	0.000907111686584861
0.6	1.41036062852983	0.000900882116847468
0.7	1.45898656538004	0.000886829553327972
0.8	1.50456334815252	0.000868751947549429
0.9	1.54752913995206	0.000848766180020943
1	1.588229132647	0.000828080678797027

where T1 is the table of the first equation, and T2 is that of the second equation.

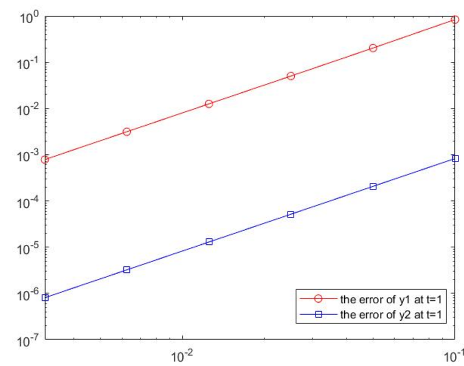
And now, we can draw the graph of two functions with approximation with different h and the real plot, that is



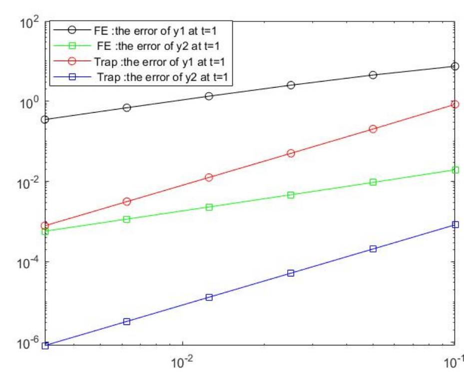
and for y2, we can hardly see the difference of the plots, so I zoom in to get the second graph:



and the error at $t = 1$ for different h is:



(f) We can observe that Trapezoidal method have a higher accuracy than the forward Euler method, and have a higher order of error term, to see this clearly, we can plot two error loglog graph onto the same picture:



Obviously, the slope of Trapezoidal method is steeper than that of forward Euler method, which means Trapezoidal method will converge faster than forward Euler Method as h decrease. To be more specifically, forward Euler generate an error of order 1, and Trapezoidal method generate an error of order 2, and this conclusion consists of the graph we draw earlier that we can directly distinguish the lines and see the convergence of forward Euler method, but for Trapezoid method, we can hardly see the differences of the graph directly. And we can compute the slope of two method, four plotted lines, which are approximately -1 and -2, in the sense of first and second order error

Listing 11: slope

```
FE3_loglog
log2(red_fac1)

ans =

  -0.841391423219877
  -0.914602037851247
  -0.955601752331824
  -0.977351074346422

log2(red_fac2)

ans =

  -1.022029497651785
  -1.010779758256592
  -1.005333131388889
  -1.002652629188419

trap_loglog
log2(red_fac1)

ans =

  -2.011194222139940
  -2.002785465832618
  -2.000695553677057
  -2.000173837648613

log2(red_fac2)

ans =

  -2.000432034317968
  -2.000108363413450
  -2.000027120217569
  -2.000006814943601
```

Besides that, we can observe that $y2$ (the second function) always generate smaller error than $y1$ (the first function) during the approximation process, and when we observe the figures, we can find that the graph of $y2$ is smoother than that of $y1$ in the interval $[0, 1]$, this make sense that we can always approximate the function with smaller local change better than the function with local change dramatically.

4. • **Forward Euler Method**:

$$
\begin{aligned}
x_{j+1} &= x_j + h \cdot f_x(\phi_j, x(\phi_j), y(\phi_j)) \\
&= x_j - h \cdot y(\phi_j) \\
&= x_j - h \cdot y_j \\
y_{j+1} &= y_j + h \cdot f_y(\phi_j, x(\phi_j), y(\phi_j)) \\
&= y_j + h \cdot x(\phi_j) \\
&= y_j + h \cdot x_j
\end{aligned}
$$

- **Backward Euler Method**:

$$
\begin{aligned}
x_{j+1} &= x_j + h \cdot f_x(\phi_{j+1}, x(\phi_{j+1}), y(\phi_{j+1})) \\
&= x_j - h \cdot y(\phi_{j+1}) \\
&= x_j - h \cdot y_{j+1} \\
y_{j+1} &= y_j + h \cdot f_y(\phi_{j+1}, x(\phi_{j+1}), y(\phi_{j+1})) \\
&= y_j + h \cdot x(\phi_{j+1}) \\
&= y_j + h \cdot x_{j+1} \\
x_{j+1} &= x_j - h \cdot y_j - h^2 \cdot x_{j+1} \\
&= \frac{x_j - h \cdot y_j}{1 + h^2}
\end{aligned}
$$

- **Trapezoidal Method**:

$$
\begin{aligned}
x_{j+1} &= x_j + \frac{h}{2}\left(f_x(\phi_j, x(\phi_j), y(\phi_j)) + f_x(\phi_{j+1}, x(\phi_{j+1}), y(\phi_{j+1}))\right) \\
&= x_j - \frac{h}{2}\left(y(\phi_j) + y(\phi_{j+1})\right) \\
&= x_j - \frac{h}{2}\left(y_j + y_{j+1}\right) \\
y_{j+1} &= y_j + \frac{2}{h}\left(f_y(\phi_j, x(\phi_j), y(\phi_j)) + f_y(\phi_{j+1}, x(\phi_{j+1}), y(\phi_{j+1}))\right) \\
&= y_j + \frac{h}{2}\left(x(\phi_j) + x(\phi_{j+1})\right) \\
&= y_j + \frac{h}{2}\left(x_j + x_{j+1}\right) \\
x_{j+1} &= x_j - \frac{h}{2}\left(y_j + y_{j+1}\right) \\
&= x_j - \frac{h}{2}\left(y_j + y_j + \frac{h}{2}\left(x_j + x_{j+1}\right)\right) \\
&= x_j - h \cdot y_j - \frac{h^2}{4}\left(x_j + x_{j+1}\right)
\end{aligned}
$$

$$\left(1+\frac{h^2}{4}\right)x_{j+1} = \left(1-\frac{h^2}{4}\right)x_j - h\cdot y_j$$
$$\therefore x_{j+1} = \frac{\left(1-\frac{h^2}{4}\right)x_j - h\cdot y_j}{1+\frac{h^2}{4}}$$

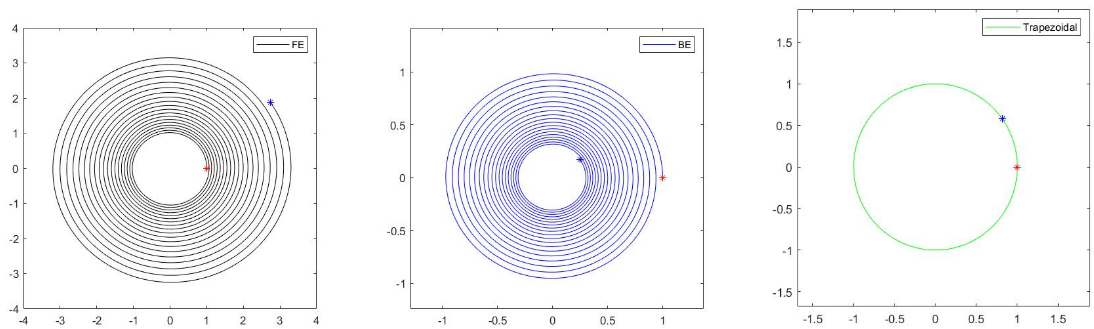
And the code is:

Listing 12: circle.m

```
h = 0.02;
r = 1;
xprime = @(t,y) -y;
yprime = @(t,x) x;
num_int = 120/h;
tvec = 0 + (0:num_int)*h;
tvec = tvec';
yvec = zeros(num_int+1,1);
xvec = ones(num_int+1,1);
xvec(1) = r;
for i = 1:num_int
   yvec(i+1) = yvec(i) + h * yprime(tvec(i),xvec(i));
   xvec(i+1) = xvec(i) + h * xprime(tvec(i),yvec(i));
end
figure(1)
plot(xvec, yvec,'k-');
hold on
plot(xvec(1),yvec(1),'r*')
plot(xvec(num_int+1),yvec(num_int+1),'b*')
legend ("FE")


yvec2 = zeros(num_int+1,1);
xvec2 = ones(num_int+1,1);
xvec2(1) = r;
for i = 1:num_int
   x1 = xvec2(i); y1 = yvec2(i);
   xvec2(i+1) = (x1 - h* y1)/(1+h^2);
   yvec2(i+1) = y1 + h * xvec2(i+1);
end
figure(2)
plot(xvec2, yvec2,'b-');
hold on
plot(xvec2(1),yvec2(1),'r*')
hold on
plot(xvec2(num_int+1),yvec2(num_int+1),'b*')
legend ("BE")

yvec3 = zeros(num_int+1,1);
xvec3 = ones(num_int+1,1);
xvec3(1) = r;
for i = 1:num_int
   x1 = xvec3(i); y1 = yvec3(i);
   xvec3(i+1) = ((1-h^2/4)*x1-h*y1)/(1+h^2/4);
   yvec3(i+1) = y1 + h/2 * (x1 +xvec3(i+1));
end
```

```
figure(3)
plot(xvec3, yvec3,'g-');
hold on
plot(xvec3(1),yvec3(1),'r*')
hold on
plot(xvec3(num_int+1),yvec3(num_int+1),'b*')
legend ("Trapezoidal")
```

with the graph of Forward Euler, Backward Euler and Trapezoidal Method:



And since I make the red point to be the beginning, and the blue one to be the ending, so we can see that FE spirals out, BE spirals in, and Trapezoidal method gives us the circle we desired.

For Forward Euler Method, we can examine that

$$\begin{aligned}
x_{j+1}^2 + y_{j+1}^2 &= (x_j - h \cdot y_j)^2 + (y_j + hx_j)^2 \\
&= x_j^2 - 2hx_jy_j + h^2y_j^2 + y_j^2 + 2hx_jy_j + h^2x_j^2 \\
&= x_j^2 + h^2y_j^2 + y_j^2 + h^2x_j^2 \\
&= (1+h^2)(x_j^2 + y_j^2) \\
&> x_j^2 + y_j^2
\end{aligned}$$

Therefore, the radius increase in each step, so it spirals out.

For Backward Euler Method, we can examine that

$$\begin{aligned}
x_{j+1}^2 + y_{j+1}^2 &= x_{j+1}^2 + (y_j + h \cdot x_{j+1})^2 \\
&= x_{j+1}^2 + y_j^2 + 2hx_{j+1}y_j + h^2x_{j+1}^2 \\
&= (1+h^2)x_{j+1}^2 + y_j^2 + 2hx_{j+1}y_j \\
&= \frac{(1+h^2)(x_j - hy_j)^2}{(1+h^2)^2} + y_j^2 + \frac{2hy_j(x_j - hy_j)}{1+h^2} \\
&= \frac{(1+h^2)(x_j - hy_j)^2}{(1+h^2)^2} + y_j^2 + \frac{2hy_j(x_j - hy_j)}{1+h^2} \\
&= \frac{(x_j - hy_j)^2 + 2hy_j(x_j - hy_j)}{1+h^2} + y_j^2
\end{aligned}$$

$$
\begin{aligned}
&= \frac{x_j^2 - 2hx_jy_j + h^2y_j^2 + 2hx_jy_j - 2h^2y_j^2}{1+h^2} + y_j^2 \\
&= \frac{x_j^2 - h^2y_j^2}{1+h^2} + y_j^2 \\
&= x_j^2 + y_j^2 + \frac{-h^2x_j^2 - h^2y_j^2}{1+h^2} \\
&< x_j^2 + y_j^2
\end{aligned}
$$

Therefore, the radius decrease in each step, so it spirals in.

For Trapezoidal method, we can examine that

$$
\begin{aligned}
x_{j+1}^2 + y_{j+1}^2 &= x_{j+1}^2 + \left(y_j + \frac{h}{2}(x_j + x_{j+1})\right)^2 \\
&= x_{j+1}^2 + \left(y_j + \frac{h}{2}\frac{(1+\frac{h^2}{4}+1-\frac{h^2}{4})x_j - hy_j}{1+\frac{h^2}{4}}\right)^2 \\
&= x_{j+1}^2 + \left(y_j + \frac{h}{2}\frac{2x_j - hy_j}{1+\frac{h^2}{4}}\right)^2 \\
&= x_{j+1}^2 + \left(\frac{hx_j}{1+\frac{h^2}{4}} + \frac{(2-\frac{h^2}{2})y_j}{2+\frac{h^2}{2}}\right)^2 \\
&= x_{j+1}^2 + \left(\frac{2hx_j + (2-\frac{h^2}{2})y_j}{2+\frac{h^2}{2}}\right)^2 \\
&= \frac{(1-\frac{h^2}{4})^2 - 2hx_jy_j(1-\frac{h^2}{4}) + h^2y_j^2}{(1+\frac{h^2}{4})^2} + \left(\frac{2hx_j + (2-\frac{h^2}{2})y_j}{2+\frac{h^2}{2}}\right)^2 \\
&= \frac{(1-\frac{h^2}{4})^2x_j^2 - 2hx_jy_j(1-\frac{h^2}{4}) + h^2y_j^2}{(1+\frac{h^2}{4})^2} + \left(\frac{hx_j + (1-\frac{h^2}{4})y_j}{1+\frac{h^2}{4}}\right)^2 \\
&= \frac{(1-\frac{h^2}{4})^2x_j^2 - 2hx_jy_j(1-\frac{h^2}{4}) + h^2y_j^2 + h^2x_j^2 + 2hx_jy_j(1-\frac{h^2}{4}) + (1-\frac{h^2}{4})^2y_j^2}{(1+\frac{h^2}{4})^2} \\
&= \frac{(1-\frac{h^2}{4})^2x_j^2 + h^2y_j^2 + h^2x_j^2 + (1-\frac{h^2}{4})^2y_j^2}{(1+\frac{h^2}{4})^2} \\
&= \frac{(1-\frac{h^2}{2}+\frac{h^4}{16})x_j^2 + h^2y_j^2 + h^2x_j^2 + (1-\frac{h^2}{2}+\frac{h^4}{16})y_j^2}{(1+\frac{h^2}{2}+\frac{h^4}{16})} \\
&= \frac{(1+\frac{h^2}{2}+\frac{h^4}{16})x_j^2 + (1+\frac{h^2}{2}+\frac{h^4}{16})y_j^2}{(1+\frac{h^2}{2}+\frac{h^4}{16})} \\
&= x_j^2 + y_j^2
\end{aligned}
$$

Therefore, the radius does not change, so it's a circle we deserved.

5. For four-point Gaussian, we can firstly derive the Legendre Polynomial of degree 4, which is

$$
\begin{aligned}
P_2(x) &= \frac{1}{2}(3x^2 - 1) \\
P_3(x) &= \frac{1}{2}(5x^3 - 3x) \\
4P_4(x) &= 7xP_3(x) - 3P_2(x) \\
&= \frac{7}{2}(5x^4 - 3x^2) - \frac{3}{2}(3x^2 - 1) \\
&= \frac{35x^4}{2} - \frac{30x^2}{2} + \frac{3}{2} \\
P_4(x) &= \frac{35x^4}{8} - \frac{15x^2}{4} + \frac{3}{8}
\end{aligned}
$$

And we can solve $P_4(x) = 0$ to get the nodes:

$$
\begin{aligned}
\frac{35x^4}{8} - \frac{15x^2}{4} + \frac{3}{8} &= 0 \\
35x^4 - 30x^2 + 3 &= 0 \\
x^2 &= \frac{30 \pm \sqrt{900 - 4 \cdot 35 \cdot 3}}{70} \\
&= \frac{30 \pm \sqrt{480}}{70} \\
&= \frac{15 \pm 2\sqrt{30}}{35} \\
x &= \pm\sqrt{\frac{15 \pm 2\sqrt{30}}{35}} \\
x_1 &= -\sqrt{\frac{15 + 2\sqrt{30}}{35}} \\
x_2 &= -\sqrt{\frac{15 - 2\sqrt{30}}{35}} \\
x_3 &= \sqrt{\frac{15 - 2\sqrt{30}}{35}} \\
x_4 &= \sqrt{\frac{15 + 2\sqrt{30}}{35}}
\end{aligned}
$$

For the weight, since

$$\begin{aligned}
\int_{-1}^{1} f(x)dx \approx \int_{-1}^{1} Q(x)dx \\
&= \int_{-1}^{1} \sum_{j=1}^{n+1} L_j(x) f(x_j) dx \\
&= \int_{-1}^{1} \sum_{j=1}^{4} L_j(x) dx f(x_j)
\end{aligned}$$

Based on four points,

$$\begin{aligned}
L_1(x) &= \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} \\
L_2(x) &= \frac{(x-x_1)(x-x_3)(x-x_4)}{(x_2-x_1)(x_2-x_3)(x_2-x_4)} \\
L_3(x) &= \frac{(x-x_1)(x-x_2)(x-x_4)}{(x_3-x_1)(x_3-x_2)(x_3-x_4)} \\
L_4(x) &= \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_4-x_1)(x_4-x_2)(x_4-x_3)}
\end{aligned}$$

And we can get

$$\begin{aligned}
\omega_1 &= \int_{-1}^{1} L_1(x)dx \\
&= \int_{-1}^{1} \frac{(x-x_2)(x-x_3)(x-x_4)}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} dx \\
&= \frac{1}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} \int_{-1}^{1} (x-x_2)(x-x_3)(x-x_4)dx \\
&= \frac{1}{(x_1-x_2)(x_1-x_3)(x_1-x_4)} \int_{-1}^{1} x^3 - (x_2+x_3+x_4)x^2 + (x_2x_3+x_2x_4+x_3x_4)x - x_2x_3x_4 dx \\
&= -\frac{2}{3} \frac{3x_2x_3x_4 + x_2 + x_3 + x_4}{(x_1-x_2)(x_1-x_3)(x_1-x_4)}
\end{aligned}$$

$$
\begin{aligned}
\omega_2 &= \int_{-1}^{1} L_2(x)dx \\
&= -\frac{2}{3}\frac{3x_1x_3x_4 + x_1 + x_3 + x_4}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)} \\
\omega_3 &= \int_{-1}^{1} L_3(x)dx \\
&= -\frac{2}{3}\frac{3x_1x_2x_4 + x_1 + x_2 + x_4}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)} \\
\omega_4 &= \int_{-1}^{1} L_4(x)dx \\
&= -\frac{2}{3}\frac{3x_1x_2x_3 + x_1 + x_2 + x_3}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)}
\end{aligned}
$$

Since

$$
\begin{aligned}
x_1 - x_2 &= \sqrt{\frac{15 - 2\sqrt{30}}{35}} - \sqrt{\frac{15 + 2\sqrt{30}}{35}} \\
x_1 - x_3 &= -\sqrt{\frac{15 + 2\sqrt{30}}{35}} - \sqrt{\frac{15 - 2\sqrt{30}}{35}} \\
x_1 - x_4 &= -2\sqrt{\frac{15 + 2\sqrt{30}}{35}} \\
x_2 - x_3 &= -2\sqrt{\frac{15 - 2\sqrt{30}}{35}} \\
x_2 - x_4 &= -\sqrt{\frac{15 - 2\sqrt{30}}{35}} - \sqrt{\frac{15 + 2\sqrt{30}}{35}} \\
x_3 - x_4 &= \sqrt{\frac{15 - 2\sqrt{30}}{35}} - \sqrt{\frac{15 + 2\sqrt{30}}{35}}
\end{aligned}
$$

Therefore

$$
\begin{aligned}
\omega_1 &= -\frac{2}{3}\frac{-3 \cdot \frac{15 - 2\sqrt{30}}{35}\sqrt{\frac{15 + 2\sqrt{30}}{35}} + \sqrt{\frac{15 + 2\sqrt{30}}{35}}}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)} \\
&= \frac{2}{3}\frac{-3 \cdot \frac{15 - 2\sqrt{30}}{35}\sqrt{\frac{15 + 2\sqrt{30}}{35}} + \sqrt{\frac{15 + 2\sqrt{30}}{35}}}{2(\frac{15 + 2\sqrt{30}}{35} - \frac{15 - 2\sqrt{30}}{35})\sqrt{\frac{15 + 2\sqrt{30}}{35}}}
\end{aligned}
$$

$$= \frac{-\frac{15-2\sqrt{30}}{35}\sqrt{\frac{15+2\sqrt{30}}{35}} + \sqrt{\frac{15+2\sqrt{30}}{315}}}{(\frac{4\sqrt{30}}{35})\sqrt{\frac{15+2\sqrt{30}}{35}}}$$

$$= \frac{-\frac{15-2\sqrt{30}}{35} + \frac{1}{3}}{(\frac{4\sqrt{30}}{35})}$$

$$= \frac{-15+2\sqrt{30}+\frac{35}{3}}{4\sqrt{30}}$$

$$= -\frac{\sqrt{30}}{36} + \frac{1}{2}$$

$$\omega_2 = -\frac{2}{3}\frac{-3\cdot\frac{15+2\sqrt{30}}{35}\sqrt{\frac{15-2\sqrt{30}}{35}} + \sqrt{\frac{15-2\sqrt{30}}{35}}}{(x_2-x_1)(x_2-x_3)(x_2-x_4)}$$

$$= \frac{2}{3}\frac{-3\cdot\frac{15+2\sqrt{30}}{35}\sqrt{\frac{15-2\sqrt{30}}{35}} + \sqrt{\frac{15-2\sqrt{30}}{35}}}{2(\frac{15-2\sqrt{30}}{35} - \frac{15+2\sqrt{30}}{35})\sqrt{\frac{15-2\sqrt{30}}{35}}}$$

$$= \frac{2}{3}\frac{-3\cdot\frac{15+2\sqrt{30}}{35}+1}{2(\frac{15-2\sqrt{30}}{35} - \frac{15+2\sqrt{30}}{35})}$$

$$= -\frac{-(15+2\sqrt{30})+\frac{35}{3}}{4\sqrt{30}}$$

$$= \frac{\sqrt{30}}{36} + \frac{1}{2}$$

$$\omega_3 = -\frac{2}{3}\frac{-3\cdot\frac{15+2\sqrt{30}}{35}\sqrt{\frac{15-2\sqrt{30}}{35}} + \sqrt{\frac{15-2\sqrt{30}}{35}}}{(x_3-x_1)(x_3-x_2)(x_3-x_4)}$$

$$= \frac{2}{3}\frac{-3\cdot\frac{15+2\sqrt{30}}{35}\sqrt{\frac{15-2\sqrt{30}}{35}} + \sqrt{\frac{15-2\sqrt{30}}{35}}}{2(\frac{15-2\sqrt{30}}{35} - \frac{15+2\sqrt{30}}{35})\sqrt{\frac{15-2\sqrt{30}}{35}}}$$

$$= \frac{\sqrt{30}}{36} + \frac{1}{2}$$

$$
\begin{aligned}
\omega_4 &= -\frac{2}{3}\frac{-3\cdot\frac{15-2\sqrt{30}}{35}\sqrt{\frac{15+2\sqrt{30}}{35}}+\sqrt{\frac{15+2\sqrt{30}}{35}}}{(x_4-x_1)(x_4-x_2)(x_4-x_3)} \\
&= \frac{2}{3}\frac{-3\cdot\frac{15-2\sqrt{30}}{35}\sqrt{\frac{15+2\sqrt{30}}{35}}+\sqrt{\frac{15+2\sqrt{30}}{35}}}{2(\frac{15+2\sqrt{30}}{35}-\frac{15-2\sqrt{30}}{35})\sqrt{\frac{15+2\sqrt{30}}{35}}} \\
&= -\frac{\sqrt{30}}{36}+\frac{1}{2}
\end{aligned}
$$

Appendix

Code for Question 3f:

Listing 13: Trap.m

```
h = 0.1;
y1real = @(t) exp(t^2+2*t);
y2real = @(t) (3*t+1)^(1/3);

num_int = 1/h;
tvec = 0 + (0:num_int)*h;
tvec = tvec';
y1vec = ones(num_int+1,1);
errvec1 = zeros(num_int+1,1);
errvec1(1) = abs(y1vec(1) - y1real(0));
y2vec = ones(num_int+1,1);
errvec2 = zeros(num_int+1,1);
errvec2(1) = abs(y2vec(1) - y2real(0));
for i = 1:num_int
   y1vec(i+1) = y1vec(i) *(1+h*(tvec(i)+1))/(1-h*(tvec(i+1)+1));
   errvec1(i+1) = abs(y1vec(i+1) - y1real(tvec(i+1)));
   y2 = y2vec(i);
   y2vec(i+1) = fzero(@(y)(2*y^3-h)/(2*y^2)-(2*y2^3+h)/(2*y2^2),1);
   errvec2(i+1) = abs(y2vec(i+1) - y2real(tvec(i+1)));
end
T1 = table(tvec,y1vec,errvec1)
T2 = table(tvec,y2vec,errvec2)
```

Listing 14: trap2_graph.m

```
hvec = [0.1,0.05,0.025];
y1real = @(t) exp(t^2+2*t);
y2real = @(t) (3*t+1)^(1/3);
t = linspace(0,1,200);
y1realvec = ones(200,1);
for i = 1:200
    y1realvec(i) = y1real(t(i));
end
y2realvec = ones(200,1);
for i = 1:200
    y2realvec(i) = y2real(t(i));
```

```
end
figure(1);
legend1 = [];
for h = hvec
    num_int = 1/h;
    tvec = 0 + (0:num_int)*h;
    tvec = tvec';
    y1vec = ones(num_int+1,1);
    errvec1 = zeros(num_int+1,1);
    errvec1(1) = abs(y1vec(1) - y1real(0));
    for i = 1:num_int
        y1vec(i+1) = y1vec(i) *(1+h*(tvec(i)+1))/(1-h*(tvec(i+1)+1));
        errvec1(i+1) = abs(y1vec(i+1) - y1real(tvec(i+1)));
    end
    T1 = table(tvec,y1vec,errvec1);
    plot(tvec,y1vec);
    legend1 = [legend1, "y1 with " + h];
    hold all;
end
 plot(t,y1realvec);
 legend( [legend1, "real y1"] );


figure(2);
legend2 = [];
for h = hvec
    num_int = 1/h;
    tvec = 0 + (0:num_int)*h;
    tvec = tvec';
    y2vec = ones(num_int+1,1);
    errvec2 = zeros(num_int+1,1);
    errvec2(1) = abs(y2vec(1) - y2real(0));
    for i = 1:num_int
      y2 = y2vec(i);
      y2vec(i+1) = fzero(@(y)(2*y^3-h)/(2*y^2)-(2*y2^3+h)/(2*y2^2),1);
      errvec2(i+1) = abs(y2vec(i+1) - y2real(tvec(i+1)));
    end
    T2 = table(tvec,y2vec,errvec2);
    hold on
    plot(tvec, y2vec);
    legend2 = [legend2, "y2 with " + h];
end
 plot(t,y2realvec);
 legend( [legend2, "real y2"] );
hold off
```

Listing 15: trap_loglog.m

```
hvec = 0.1*2.^(-(0:5));
y1real = @(t) exp(t^2+2*t);
y2real = @(t) (3*t+1)^(1/3);
error1vec = [];
error2vec = [];
red_fac1 = [];
red_fac2 = [];
for h = hvec
```

```
    num_int = 1/h;
    tvec = 0 + (0:num_int)*h;
    tvec = tvec';
    y1= 1;
    y2vec = ones(num_int+1,1);
    for i = 1:num_int
        y1 = y1 *(1+h*(tvec(i)+1))/(1-h*(tvec(i+1)+1));
        y2 = y2vec(i);
        y2vec(i+1) = fzero(@(y)(2*y^3-h)/(2*y^2)- (2*y2^3+h)/(2*y2^2),1);
    end
    y2 = y2vec(num_int+1);
    error1 = abs(y1 - y1real(1));
    error2 = abs(y2 - y2real(1));

    if length(error1vec) > 1
        red_fac1 = [red_fac1; error1/error1vec(length(error1vec)) ];
        red_fac2 = [red_fac2; error2/error2vec(length(error1vec)) ];
    end
    error1vec = [error1vec; error1];
    error2vec = [error2vec; error2];

end
 loglog(hvec,error1vec,'ro-');
 hold on
 loglog(hvec,error2vec,'bs-');
legend( "the error of y1 at t=1", "the error of y2 at t=1" );
```