1 初始值问题的单步方法数值实验

1.1 实验目的

通过实验，使得学生掌握求解初值问题的四阶、三阶以及二阶的 Runge-Kutta 方法以及 Euler 法的实现，并通过编程实现算法。

1.2 实验内容

求解如下微分方程的数值解

$$\begin{cases} y' = y - \dfrac{2x}{y}, \quad 0 < x < 1 \\ y(0) = 1 \end{cases}$$

其中计算区间为 $[0,1]$。解析解为 $y = \sqrt{1+2x}$.

分别用 Euler 前差，改进的 Euler 公式、四阶 R-K 方法以及四阶 Adams 显格式和预估校正格式求解该问题。

取空间步长为 $h = 1/40, 1/20, 1/10, 1/5$。然后将每一种步长下的数值解分别于解析解相比较（计算绝对误差），给出收敛阶（$RR1 = LOG(ER1/ERROR1)/LOG(2.0)$）并列表显示。分析计算结果。

1.3 实验算法理论

对于求解一阶微分方程问题:

$$\begin{aligned} &y_i' = f_i(x, y_1, \cdots, y_m) \\ &y_i(x_0) = y_{i0} \\ &i = 1, 2 \cdots, m \end{aligned}$$

由初值问题的经典单步法公式可得一阶常微分方程组初值问题各阶公式:

1. Euler 方法

$$y(x_{n+1}) = y(x_n) + hf(x_n, y(x_n))$$

2. 二阶 R-K 方法（改进的 Euler 公式）

$$\begin{aligned} &y_{n+1} = y_n + \frac{h}{2}(k_1 + k_2) \\ &k_1 = f(x_n, y_n) \\ &k_2 = f(x_{n+1}, y_n + hk_1) \\ &y_0 = y(x_0) \end{aligned}$$

3. 四阶 R-K 方法

$$
\begin{aligned}
&y_{n+1}=y_n+\frac{h}{6}\left(k_1+2k_2+2k_3+k_4\right)\\
&k_1=f\left(x_n,y_n\right)\\
&k_2=f\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}hk_1\right)\\
&k_3=f\left(x_n+\frac{1}{2}h,y_n+\frac{1}{2}hk_2\right)\\
&k_4=f\left(x_n+h,y_n+hk_3\right)
\end{aligned}
$$

4. 四阶 Adams 显格式

$$
y_{n+1}=y_n+\frac{h}{24}\left(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}\right)
$$

5. 四阶 Adams 预估校正格式

$$
\begin{aligned}
\bar{y}_{n+1}&=y_n+\frac{h}{24}\left(55f_n-59f_{n-1}+37f_{n-2}-9f_{n-3}\right)\\
y_{n+1}&=y_n+\frac{h}{24}\left(9f(x_{n+1},\bar{y}_{n+1})+19f_n-5f_{n-1}+f_{n-2}\right)
\end{aligned}
$$

1.4 实验程序

我们编写了以下 MATLAB 函数，实现使用欧拉法、梯形法、四阶 R-K 方法、四阶 Adams 显格式或预估校正格式求解一阶常微分方程：

ode

```
function [x,y] = ode(f,a,b,h,y0,method)
%ODE 通过单步法、线性多步法求解一阶微分方程问题 y  = f(x,y)
%   h必须刚好划分 [a,b]
%   method  1: euler, 2:second_order_R_K, 3: fourth_order_R_K, 4:
    fourth_order_Adams_explicit, 5:
    fourth_order_Adams_predictive_correction
x = a:h:b;
m = size(x,2);
y = zeros(1,m);
y(1) = y0;
for i = 1:m-1
    if method == 1 % euler
        y(i+1) = y(i) + h * f(x(i),y(i));
    elseif method == 2 % second_order_R_K
        k1 = f(x(i),y(i));
        k2 = f(x(i+1),y(i)+h*k1);
```

```
        y(i+1) = y(i) + h * (k1+k2)/2;
    elseif method == 3 % fourth_order_R_K
        k1 = f(x(i),y(i));
        k2 = f(x(i)+h/2,y(i)+h/2*k1);
        k3 = f(x(i)+h/2,y(i)+h/2*k2);
        k4 = f(x(i)+h,y(i)+h*k3);
        y(i+1) = y(i) + h*(k1+2*k2+2*k3+k4)/6;
    elseif method == 4 % fourth_order_Adams_explicit
        % 使用线性多步法时，所需的y2,y3,...,y(k)这k-1个值由同阶R-K方法算出
        if i == 1
            [~,y(1:4)] = ode(f,a,x(4),h,y0,3);
        elseif i >= 4
            y(i+1) = y(i) + h*( 55*f(x(i),y(i)) - 59*f(x(i-1),y(i-1)) + 37*f(x(i-2),y(i-2)) - 9*f(x(i-3),y(i-3)) )/24;
        end
    elseif method == 5 % fourth_order_Adams_predictive_correction
        if i == 1
            [~,y(1:4)] = ode(f,a,x(4),h,y0,3);
        elseif i >= 4
            y(i+1) = y(i) + h*( 55*f(x(i),y(i)) - 59*f(x(i-1),y(i-1)) + 37*f(x(i-2),y(i-2)) - 9*f(x(i-3),y(i-3)) )/24;
            y(i+1) = y(i) + h*( 9*f(x(i+1),y(i+1)) + 19*f(x(i),y(i)) - 5*f(x(i-1),y(i-1)) + f(x(i-2),y(i-2)) )/24;
        end
    else
        '出错，method不正确'
        break
    end
end
end
```

1.5 实验结果及分析

使用以下 MATLAB 脚本对上述算法进行测试：

ode test

```
f = @(x,y)y-2*x/y; a = 0; b = 1; y0 = 1; h_ = [1/5,1/10,1/20,1/40];
methods = {'euler','second_order_R_K','fourth_order_R_K','
```

```
    fourth_order_Adams_explicit','
    fourth_order_Adams_predictive_correction'};
for method=1:5
    solutions = zeros(size(h_,2)+1,size(a:min(h_):b,2))*nan;
    for i = 1:size(h_,2)
        [x,y]=ode(f,a,b,h_(i),y0,method);
        solutions(i,1:2^(size(h_,2)-i):size(solutions,2)) = y;
    end
    % 计算精确解
    solutions(size(h_,2)+1,:) = sqrt(1+2*(a:min(h_):b));
    % 计算绝对误差
    abs_error = solutions;
    for i = 1:size(h_,2)
        abs_error(i,:) = solutions(i,:) - solutions(size(h_,2)+1,:);
    end
    % 计算收敛阶
    rr = [];
    for i = 1:size(h_,2)-1
        rr=[rr,abs(abs_error(i,size(abs_error,2))/abs_error(i+1,size(
            abs_error,2)))];
    end
    % 输出各项数据
    methods(method)
    solutions'
    abs_error'
    rr1 = mean(log(rr)/log(2))
end
```

各迭代方法的运行结果如图 1-10 和表 1-10 所示。由于数据过多，我们仅在 Euler 方法的计算结果表 1-2 中完整展现 40 个节点的数据值，而在表 3-10 中仅保留 5 个数据点。

表 11 列出了通过估计得到的各迭代格式的收敛阶数。

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.0250				1.0250	1.0247
0.0500			1.0500	1.0494	1.0488
0.0750				1.0733	1.0724
0.1000		1.1000	1.0977	1.0966	1.0954
0.1250				1.1195	1.1180
0.1500			1.1435	1.1419	1.1402
0.1750				1.1638	1.1619
0.2000	1.2000	1.1918	1.1876	1.1854	1.1832
0.2250				1.2066	1.2042
0.2500			1.2301	1.2275	1.2247
0.2750				1.2480	1.2450
0.3000		1.2774	1.2713	1.2681	1.2649
0.3250				1.2880	1.2845
0.3500			1.3113	1.3076	1.3038
0.3750				1.3269	1.3229
0.4000	1.3733	1.3582	1.3501	1.3459	1.3416
0.4250				1.3647	1.3601
0.4500			1.3880	1.3833	1.3784
0.4750				1.4016	1.3964
0.5000		1.4351	1.4250	1.4197	1.4142
0.5250				1.4376	1.4318
0.5500			1.4612	1.4553	1.4491
0.5750				1.4727	1.4663
0.6000	1.5315	1.5090	1.4966	1.4900	1.4832
0.6250				1.5071	1.5000
0.6500			1.5313	1.5241	1.5166
0.6750				1.5409	1.5330
0.7000		1.5803	1.5654	1.5575	1.5492
0.7250				1.5740	1.5652
0.7500			1.5990	1.5903	1.5811
0.7750				1.6064	1.5969
0.8000	1.6811	1.6498	1.6320	1.6225	1.6125
0.8250				1.6384	1.6279
0.8500			1.6646	1.6542	1.6432
0.8750				1.6698	1.6583
0.9000		1.7178	1.6968	1.6854	1.6733
0.9250				1.7008	1.6882
0.9500			1.7286	1.7162	1.7029
0.9750				1.7314	1.7176
1.0000	1.8269	1.7848	1.7600	1.7465	1.7321

表 1: Euler 方法计算结果

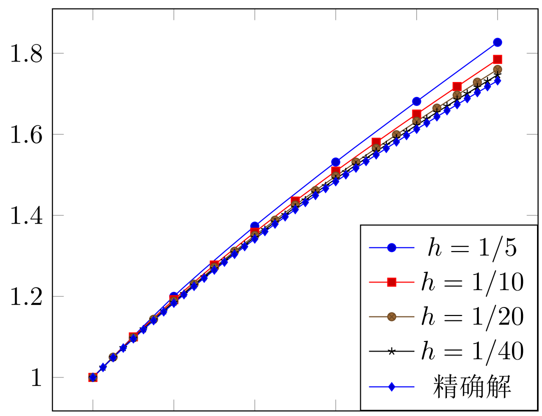

图 1: Euler 方法计算结果

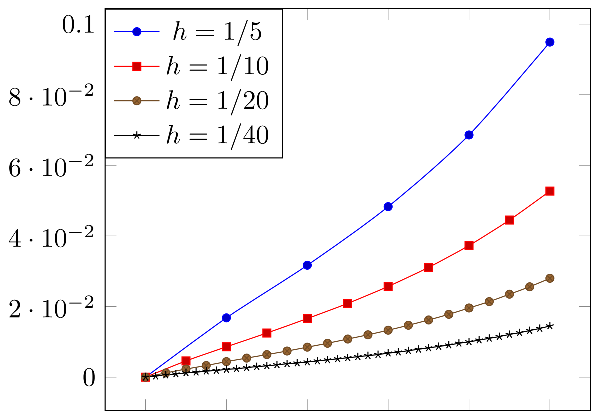

图 2: Euler 方法绝对误差

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	0	0	0	0	1.0000
0.0250				0.0003	1.0247
0.0500			0.0012	0.0006	1.0488
0.0750				0.0009	1.0724
0.1000		0.0046	0.0023	0.0012	1.0954
0.1250				0.0014	1.1180
0.1500			0.0033	0.0017	1.1402
0.1750				0.0019	1.1619
0.2000	0.0168	0.0086	0.0044	0.0022	1.1832
0.2250				0.0024	1.2042
0.2500			0.0054	0.0027	1.2247
0.2750				0.0030	1.2450
0.3000		0.0125	0.0064	0.0032	1.2649
0.3250				0.0035	1.2845
0.3500			0.0074	0.0038	1.3038
0.3750				0.0040	1.3229
0.4000	0.0317	0.0166	0.0085	0.0043	1.3416
0.4250				0.0046	1.3601
0.4500			0.0096	0.0049	1.3784
0.4750				0.0052	1.3964
0.5000		0.0209	0.0108	0.0055	1.4142
0.5250				0.0058	1.4318
0.5500			0.0120	0.0061	1.4491
0.5750				0.0064	1.4663
0.6000	0.0483	0.0257	0.0133	0.0068	1.4832
0.6250				0.0071	1.5000
0.6500			0.0147	0.0075	1.5166
0.6750				0.0079	1.5330
0.7000		0.0311	0.0162	0.0083	1.5492
0.7250				0.0087	1.5652
0.7500			0.0178	0.0091	1.5811
0.7750				0.0096	1.5969
0.8000	0.0686	0.0373	0.0196	0.0100	1.6125
0.8250				0.0105	1.6279
0.8500			0.0214	0.0110	1.6432
0.8750				0.0115	1.6583
0.9000		0.0445	0.0235	0.0121	1.6733
0.9250				0.0126	1.6882
0.9500			0.0256	0.0132	1.7029
0.9750				0.0138	1.7176
1.0000	0.0949	0.0527	0.0280	0.0145	1.7321

表 2: Euler 方法绝对误差

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.2000	1.1867	1.1841	1.1834	1.1833	1.1832
0.4000	1.3483	1.3434	1.3421	1.3417	1.3416
0.6000	1.4937	1.4860	1.4839	1.4834	1.4832
0.8000	1.6279	1.6165	1.6135	1.6127	1.6125
1.0000	1.7542	1.7379	1.7335	1.7324	1.7321

表 3: 改进的 Euler 方法计算结果

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	0	0	0	0	1.0000
0.2000	0.0035	0.0009	0.0002	0.0001	1.1832
0.4000	0.0067	0.0017	0.0004	0.0001	1.3416
0.6000	0.0105	0.0027	0.0007	0.0002	1.4832
0.8000	0.0154	0.0040	0.0010	0.0003	1.6125
1.0000	0.0222	0.0058	0.0015	0.0004	1.7321

表 4: 改进的 Euler 方法绝对误差

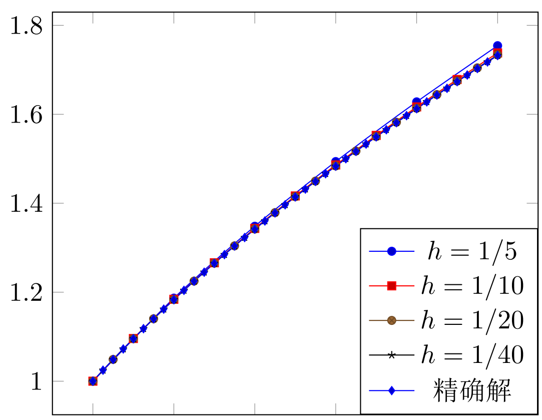

图 3: 改进的 Euler 方法计算结果

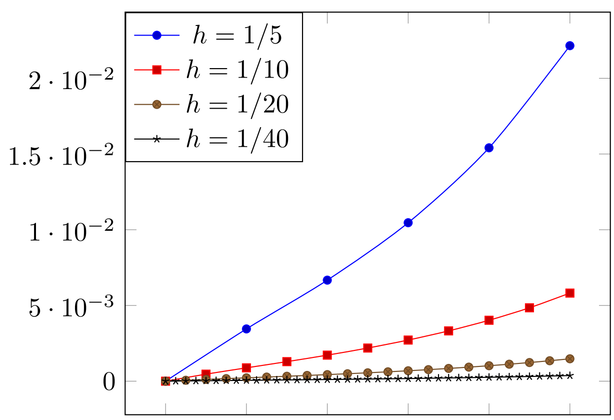

图 4: 改进的 Euler 方法绝对误差

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.2000	1.1832	1.1832	1.1832	1.1832	1.1832
0.4000	1.3417	1.3416	1.3416	1.3416	1.3416
0.6000	1.4833	1.4832	1.4832	1.4832	1.4832
0.8000	1.6125	1.6125	1.6125	1.6125	1.6125
1.0000	1.7321	1.7321	1.7321	1.7321	1.7321

表 5: 四阶 R-K 方法计算结果

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	0	0	0	0	1.0000
0.2000	0.0000	0.0000	0.0000	0.0000	1.1832
0.4000	0.0000	0.0000	0.0000	0.0000	1.3416
0.6000	0.0000	0.0000	0.0000	0.0000	1.4832
0.8000	0.0001	0.0000	0.0000	0.0000	1.6125
1.0000	0.0001	0.0000	0.0000	0.0000	1.7321

表 6: 四阶 R-K 方法绝对误差

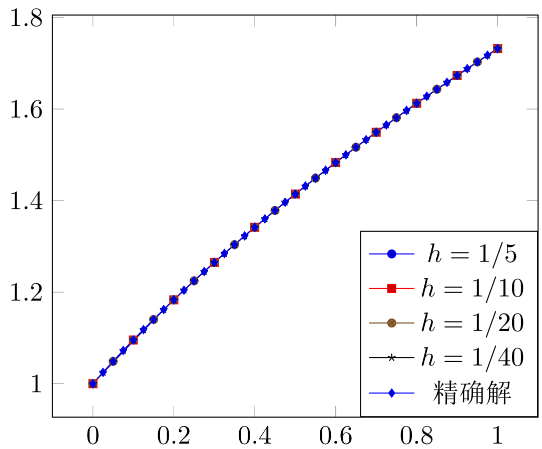

图 5: 四阶 R-K 方法计算结果

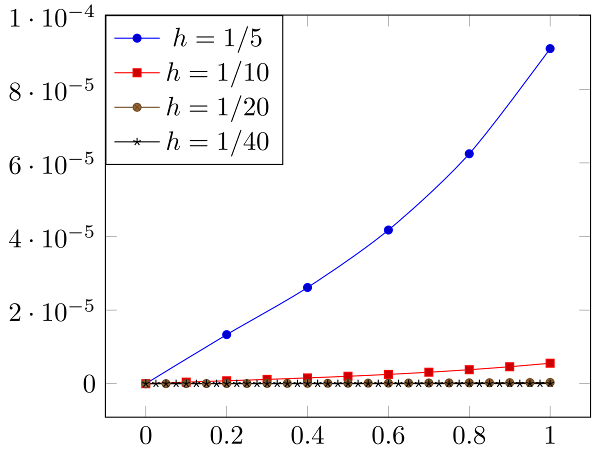

图 6: 四阶 R-K 方法绝对误差

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.2000	1.1832	1.1832	1.1832	1.1832	1.1832
0.4000	1.3417	1.3416	1.3416	1.3416	1.3416
0.6000	1.4833	1.4830	1.4832	1.4832	1.4832
0.8000	1.6114	1.6121	1.6124	1.6124	1.6125
1.0000	1.7298	1.7316	1.7320	1.7320	1.7321

表 7: 四阶 Adams 显格式计算结果

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	0	0	0	0	1.0000
0.2000	0.0000	0.0000	-0.0000	-0.0000	1.1832
0.4000	0.0000	-0.0001	-0.0000	-0.0000	1.3416
0.6000	0.0000	-0.0002	-0.0000	-0.0000	1.4832
0.8000	-0.0010	-0.0003	-0.0000	-0.0000	1.6125
1.0000	-0.0022	-0.0005	-0.0001	-0.0000	1.7321

表 8: 四阶 Adams 显格式绝对误差

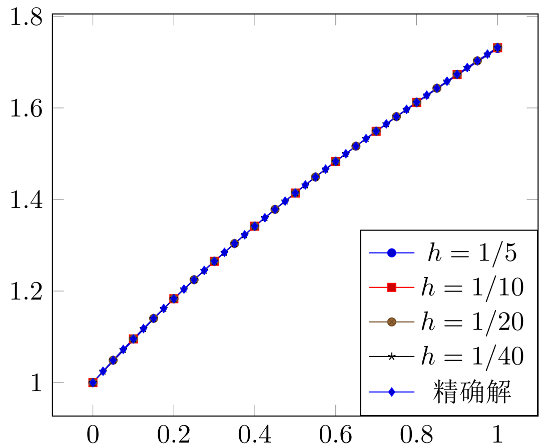

图 7: 四阶 Adams 显格式计算结果

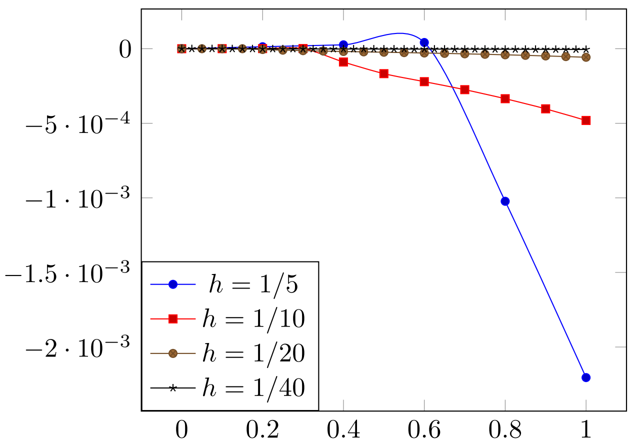

图 8: 四阶 Adams 显格式绝对误差

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	1.0000	1.0000	1.0000	1.0000	1.0000
0.2000	1.1832	1.1832	1.1832	1.1832	1.1832
0.4000	1.3417	1.3416	1.3416	1.3416	1.3416
0.6000	1.4833	1.4832	1.4832	1.4832	1.4832
0.8000	1.6124	1.6125	1.6125	1.6125	1.6125
1.0000	1.7320	1.7321	1.7321	1.7321	1.7321

表 9: 四阶 Adams 预估校正格式计算结果

x_i	$h=\frac{1}{5}$	$h=\frac{1}{10}$	$h=\frac{1}{20}$	$h=\frac{1}{40}$	精确解
0	0	0	0	0	1.0000
0.2000	0.0000	0.0000	0.0000	0.0000	1.1832
0.4000	0.0000	0.0000	0.0000	0.0000	1.3416
0.6000	0.0000	0.0000	0.0000	0.0000	1.4832
0.8000	-0.0000	-0.0000	0.0000	0.0000	1.6125
1.0000	-0.0001	-0.0000	0.0000	0.0000	1.7321

表 10: 四阶 Adams 预估校正格式绝对误差

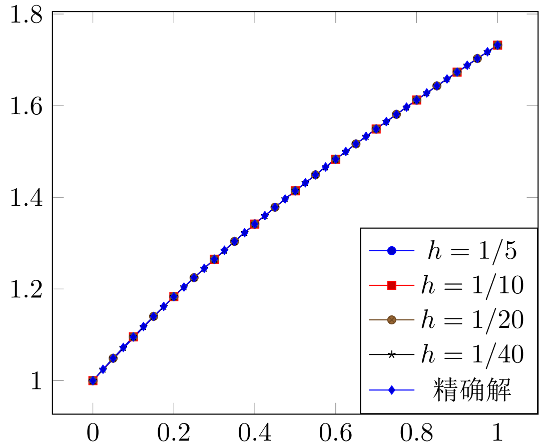

图 9: 四阶 Adams 预估校正格式计算结果

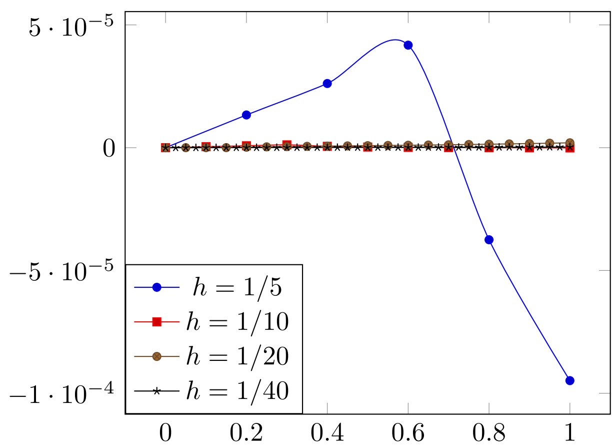

图 10: 四阶 Adams 预估校正格式绝对误差

算法	Euler 法	改进的 Euler 法	四阶 R-K 法	四阶 Adams 显格式	四阶 Adams 预估校正格式
收敛阶	0.9050	1.9653	4.0267	2.9268	2.8125

表 11: 各迭代格式的收敛阶