作业：

1.用欧拉方法和龙格—库塔方法求微分方程数值解，画出解的图形，对结果进行分析比较。

$$x^2y''+xy'+(x^2-n^2)y=0, y\left(\frac{\pi}{2}\right)=2, y'\left(\frac{\pi}{2}\right)=-\frac{2}{\pi}$$

（Bessel 方程，令$n=\frac{1}{2}$），精确解$y=\sin x\sqrt{\frac{2\pi}{x}}$。

解：

改进的欧拉法：

改进的欧拉法的公式为$y_{n+1}=y_n+\frac{h}{2}[f(x_n,y_n)+f(x_{n+1},y_n+hf(x_n,y_n))]$

原方程组：$\begin{cases} x^2y''+xy'+(x^2-\frac{1}{4})y=0 \\ y(\frac{\pi}{2})=2 \\ y'(\frac{\pi}{2})=-\frac{2}{\pi} \end{cases}$，即$\begin{cases} y''=-\frac{y'}{x}+(\frac{1}{4x^2}-1)y \\ y(\frac{\pi}{2})=2 \\ y'(\frac{\pi}{2})=-\frac{2}{\pi} \end{cases}$

令$\frac{dy}{dx}=z$，则原方程组改写为：$\begin{cases} \frac{dy}{dx}=z \\ \frac{dz}{dx}=-\frac{z}{x}+(\frac{1}{4x^2}-1)y \\ y(\frac{\pi}{2})=2 \\ z(\frac{\pi}{2})=-\frac{2}{\pi} \end{cases}$

将上式待解 y、z 视作同一个矢量$\vec{y}=\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$的两个分量，

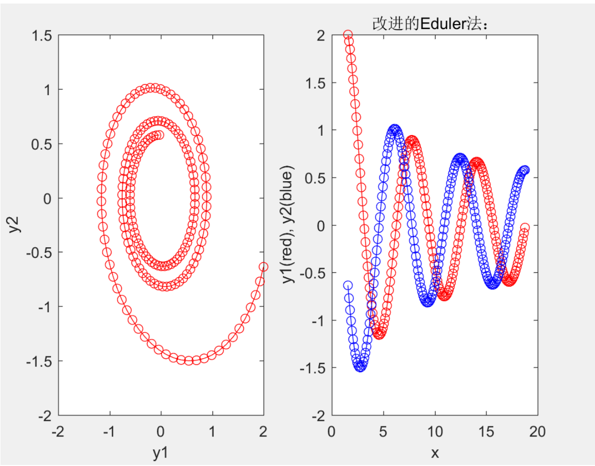
将方程组写成向量形式：$\frac{d}{dx}\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}=\begin{bmatrix} y_2 \\ -\frac{y_2}{x}+(\frac{1}{4x^2}-1)y_1 \end{bmatrix}$，$\begin{cases} y_1(\frac{\pi}{2})=2 \\ y_2(\frac{\pi}{2})=-\frac{2}{\pi} \end{cases}$

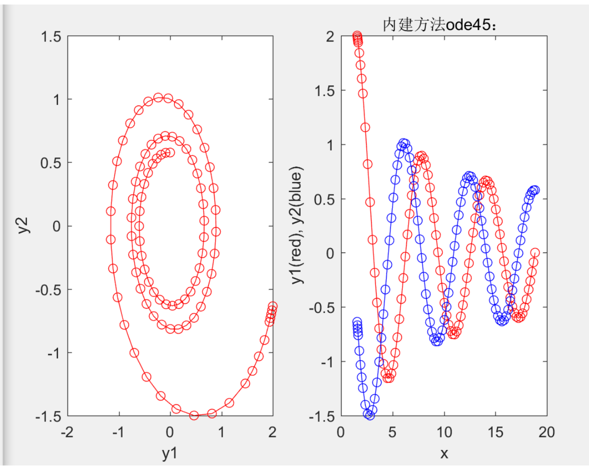
matlab程序：

```
%%%ODE_Euler_Method.m
%%%Euler method to solve ODEs
clearvars; close all; format short; format compact; clc;

xspan = [pi/2, 6*pi];
y0 = [2; -2/pi]; %%%初值
h = 0.1; %%%步长
[x, y] = ODEs_Euler(@odefunc, xspan, y0, h);
figure;
title('Eduler');
subplot(1, 2, 1); plot(y(1, :), y(2, :), 'ro-'); xlabel('y1'); ylabel('y2');
subplot(1, 2, 2); plot(x, y(1, :), 'ro-', x, y(2, :), 'bo-'); xlabel('x');
ylabel('y1(red), y2(blue)');

[xs, ys] = ode45(@odefunc, xspan, y0);
figure;
title('ode45');
subplot(1, 2, 1); plot(ys(:, 1), ys(:, 2), 'ro-'); xlabel('y1'); ylabel('y2');
subplot(1, 2, 2);plot(xs, ys(:, 1), 'ro-', xs, ys(:, 2), 'bo-'); xlabel('x');
ylabel('y1(red), y2(blue)');
```

```
function [x, y] = ODEs_Euler(odefunc, xspan, y0, h)
x0 = xspan(1);
xf = xspan(2);
n = (xf - x0)/h + 1;
n = floor(n);
x = zeros(1, n);
y = zeros(numel(y0), n);
x(1) = x0;
y(:, 1) = y0;

for i = 1: n-1
    x(i+1) = x(i) + h;
    y(:, i+1) = y(:, i) + 0.5 * h * (odefunc(x(i), y(:, i)) + odefunc(x(i+1),
y(:, i) + h * odefunc(x(i), y(:, i))));
end
end
```

```
function [ydot] = odefunc(x, y)
ydot = [y(2); -y(2)/x + (1/(4*x*x) - 1)*y(1)];
end
```





龙格-库塔方法：

方程组的分析同上

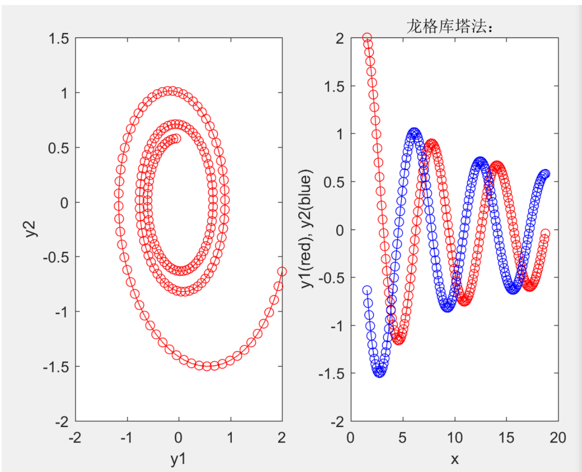
matlab程序：

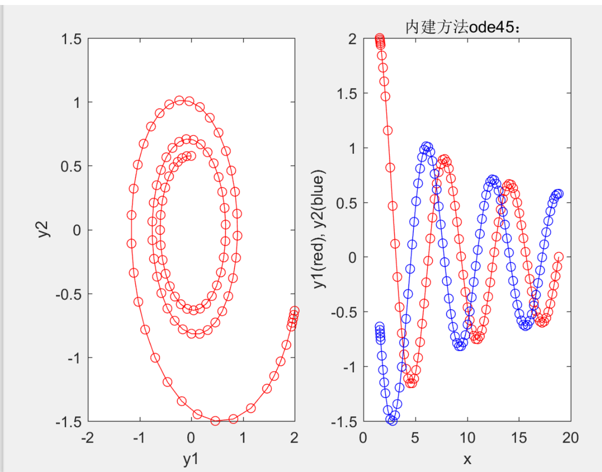
```
%%%ODE_RK_Method.m
%%%RK method to solve ODEs
clearvars; close all; format short; format compact; clc;
%调用龙格库塔方法求解微分方程
%函数原型：function [x, y] = my_rk4(odefunc, xspan, y0, h)

xspan = [pi/2, 6*pi];
y0 = [2; -2/pi];
h = 0.1;
[x, y] = my_rk4(@odefunc, xspan, y0, h);
figure;
subplot(1, 2, 1); plot(y(1, :), y(2, :), 'ro-'); xlabel('y1'); ylabel('y2');
subplot(1, 2, 2); plot(x, y(1, :), 'ro-', x, y(2, :), 'bo-'); xlabel('x');
ylabel('y1(red), y2(blue)');
title('龙格库塔法：');
```

```
[xs, ys] = ode45(@odefunc, xspan, y0);
figure;
subplot(1, 2, 1); plot(ys(:, 1), ys(:, 2), 'ro-'); xlabel('y1'); ylabel('y2');
subplot(1, 2, 2);plot(xs, ys(:, 1), 'ro-', xs, ys(:, 2), 'bo-'); xlabel('x');
ylabel('y1(red), y2(blue)');
title('内建方法ode45：');
```

```
%%%龙格库塔方法求解微分方程
function [x, y] = my_rk4(odefunc, xspan, y0, h)
x0 = xspan(1);
xf = xspan(2);
n = (xf - x0)/h + 1;
n = floor(n);
x = zeros(1, n);
y = zeros(numel(y0), n);
x(1) = x0;
y(:, 1) = y0;

for i = 1: n-1
    x(i+1) = x(i) + h;
    k1 = odefunc(x(:, i), y(:, i));
    k2 = odefunc(x(:, i) + h/2, y(:, i) + (h*k1)/2);
    k3 = odefunc(x(:, i) + h/2, y(:, i) + (h*k2)/2);
    k4 = odefunc(x(:, i) + h, y(:, i) + h*k3);
    y(:, i+1) = y(:, i) + h/6*(k1+2*k2+2*k3+k4);
end
end
```

```
function [ydot] = odefunc(x, y)
ydot = [y(2); -y(2)/x + (1/(4*x*x) - 1)*y(1)];
end
```





2. 一只小船渡过宽为 的河流，目标是起点 正对着的另一岸点。已知河水流速 与船在静水中的速度 之比为 。

（i）建立小船航线的方程，求其解析解。

（ii）设 m， m/s， m/s，用数值解法求渡河所需时间、任意时刻小船的位置及航行曲线，作图，并与解析解比较。

题目缺参数：河流宽度、河水流速与船静水速度之比

3. 求二维拉普拉斯方程

$$\Delta u=\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0$$

在边界条件$u(x,y)|_{x=0}=u(x,y)|_{x=6}=u(x,y)|_{y=0}=0$，$u(x,y)|_{y=2}=10$下的数值解。

解：

对拉普拉斯方程采用正方形网格剖分，步长为1，划分为12个正方形网格，对其中5个节点按中心差分格式列方程组，将问题转化为AX=B的形式，内节点编号与坐标关系：

$\text{节点编号}=\text{横坐标}$

对于每一个$(x_i,y_i)\in D_0$，利用数值微分方程：

$$\frac{\partial^2 u(x_i,y_j)}{\partial x^2}=\frac{1}{h_1^2}[u(x_{i+1},y_j)-2u(x_i,y_j)+u(x_{i-1},y_j)]-\frac{1}{12}h_1^2\frac{\partial^4 u(\xi_i,y_j)}{\partial x^4},\xi_i\in(x_{i-1},x_{i+1})$$

$$\frac{\partial^2 u(x_i,y_j)}{\partial y^2}=\frac{1}{h_2^2}[u(x_i,y_{j+1})-2u(x_i,y_j)+u(x_i,y_{j-1})]-\frac{1}{12}h_2^2\frac{\partial^4 u(x_i,\eta_j)}{\partial y^4},\eta_j\in(y_{j-1},y_{j+1})$$

式中$h1$、$h2$分别是沿x、y轴方向的步长

将上述公式带入Laplace方程第一边值问题的条件中，取步长$h1$=$h2$=1,即取正方形网格时，每个节点差分方程为：

$$4u_{ij}-(u_{(i+1)j}+u_{(i-1)j}+u_{i(j+1)}+u_{i(j-1)})=0$$

再依次带入i=1,2,3,4,5,j=1可得到差分方程组：$\begin{cases}4u_1-u_2=10\\4u_2-u_1-u_3=10\\4u_3-u_4-u_2=10\\4u_4-u_5-u_3=10\\4u_5-u_4=10\end{cases}$

则：$A=\begin{bmatrix}4 & -1 & 0 & 0 & 0\\ -1 & 4 & -1 & 0 & 0\\ 0 & -1 & 4 & -1 & 0\\ 0 & 0 & -1 & 4 & -1\\ 0 & 0 & 0 & -1 & 4\end{bmatrix}$,b= $\begin{bmatrix}10\\10\\10\\10\\10\end{bmatrix}$

$AX=b, X=[u1 \quad u2 \quad u3 \quad u4 \quad u5]'$

用matlab解方程：

matlab代码：

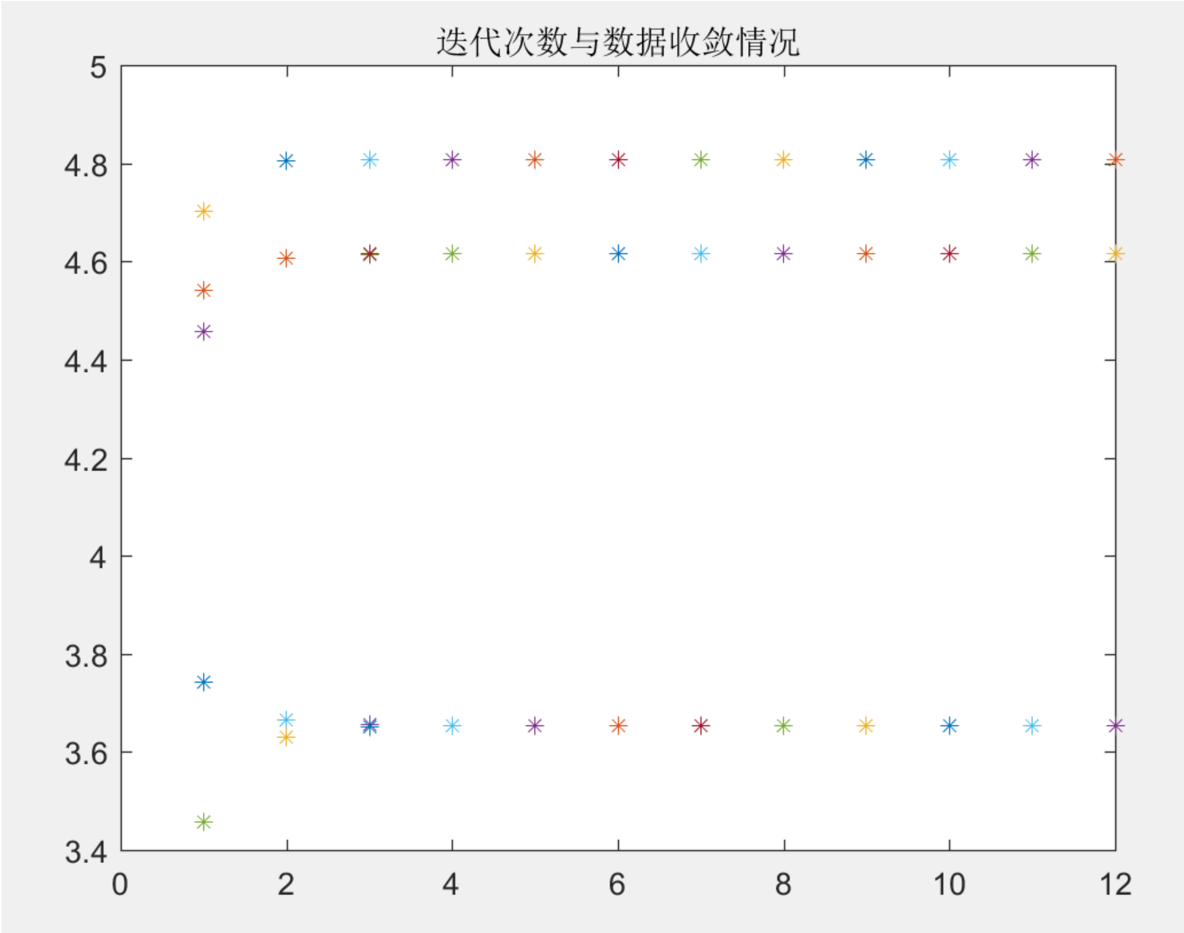
```
%Laplace_Method.m
clearvars; close all; clc;
A=[4 -1 0 0 0; -1 4 -1 0 0; 0 -1 4 -1 0; 0 0 -1 4 -1; 0 0 0 -1 4];
B=[10 10 10 10 10]';
X = A\B
disp('-----');
n=5;
x0=ones(n,1);
esp=1e-8;
m=500;
w=1.3;
[x,k]= LinearEquations_SSOR(A,B,x0,m,esp,w);%超松弛迭代法求解线性方程组
k
disp('-----');
x
```

```
%超松弛迭代法函数
function [x, k] = LinearEquations_SSOR(A, b, x0, m, eps, w)
%输入：
% A：系数矩阵
% b：常数矩阵；
% x0：初始解；
% m：最大迭代次数；
% eps：精度阈值；
% w：松弛因子；
% 输出：
% x：近似解；
% k：迭代次数；
n = length(x0);
x1 = x0;
x2 = zeros(n, 1);
x3 = zeros(n, 1);
r = max(abs(b - A*x1));
k = 0;
while r > eps
    for i = 1 : n
        sum = 0;
        for j = 1 : n
            if j > i
                sum = sum + A(i, j) * x1(j);
            elseif j < i
                sum = sum + A(i, j) * x2(j);
            end
        end
        x2(i) = (1 - w)*x1(i) + w*(b(i) - sum) / (A(i, i) + eps);
```

```
    end
    for i = n : -1 : 1
        sum = 0;
        for j = 1 : n
            if j > i
                sum = sum + A(i, j) * x3(j);
            elseif j < i
                sum = sum + A(i, j) * x2(j);
            end
        end
        x3(i) = (1 - w) * x2(i) + w * (b(i) - sum) / A(i, i);
    end
    r = max(abs(x3 - x1));
    x1 = x3;
    k = k + 1;
    if k > m
        x = [];
        return;
    end
   for i=1:n
    plot(k,x1(i),'*');
    title('迭代次数与数据收敛情况')
    hold on;
   end
end
x = x1;
end
```

运行结果:

```
X =
     3.6538
     4.6154
     4.8077
     4.6154
     3.6538
-----
k =
     12
-----
x =
     3.6538
     4.6154
     4.8077
     4.6154
     3.6538
```



4. 求初边值问题

$$
\begin{cases}
\dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2}, & t > 0,\ 0 < x < 1 \\
u(x,0) = 1, & 0 \le x \le 1 \\
u'_x(0,t) - u(0,t) = 0 & \\
u'_x(1,t) + u(1,t) = 0 & t \ge 0
\end{cases}
$$

在$0 \le t \le 3$范围内的数值解。

解:

由$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$得到迭代格式: $\frac{u_{i,j+1} - u_{i,j}}{h_t} - \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h_x^2} = 0$,令$r = \frac{h_t}{h_x^2}$,整理得:

$u_{i,j+1} + (2r-1)u_{k,j} - ru_{k+1,j} - ru_{k-1,j} = 0 \ldots\ldots (1)$

由$u(x,0) = 1$得到:

$u_{i,0}=1\ldots\ldots(2)$

由$u_x'(0,t)-u(0,t)=0$得到迭代格式:

$u_{1,j}=(h_x+1)u_{0,j}\ldots\ldots(3)$

由$u_x'(1,t)+u(1,t)=0$得到迭代格式:

$u_{n_x,j}=u_{n_x-1,j}/(h_x+1)\ldots\ldots(4)$

由以上四个式子列出并解方程可得到u的矩阵，下为matlab求解过程:

```
%第四道题的求解
clear all;close all;clc;

tf = 3;
xf = 1;
ht = 0.03;
hx = 0.01;
nt = floor(tf/ht+1);
nx = floor(xf/hx+1);
%t = 0:ht:tf;
%x = 0:hx:xf;
u = zeros(nx, nt);
b = zeros(nx*nt+2,1);
A = zeros(nx*nt+2, nx*nt); %系数矩阵
r = ht/(hx*hx);

%u(i,1)=1 共nx个方程
for i = 1:nx
    A(i, i) = 1; %前面的标号是第几个方程 后面的标号是变量的下标=i+nx*(j-1)
    b(i) = 1; %标号是第几个方程
end
n = nx; %n表示方程数

%u(2,j)=(hx+1)u(1,j) 共nt个方程
for j = 1:nt
    n = n+1;
    A(n, 2+nx*(j-1))=1;
    A(n, 1+nx*(j-1))=-1-hx;
end

%u(nx,j)=u(nx-1,j)/(1+hx) 共nt个方程
for j = 1:nt
    n = n+1;
    A(n, nx+nx*(j-1))=1;
    A(n, nx-1+nx*(j-1))=-1/(1+hx);
    b(n)=0;
end

%u(i,j+1)+(2r-1)u(i,j)-r(i+1,j)-ru(i-1,j)=0 共(nx-2)*(nt-1)个方程
for i = 2:nx-1
    for j = 1:nt-1
        n = n+1;
        A(n, i+nx*j)=1;
        A(n, i+nx*(j-1))=2*r-1;
        A(n, i+1+nx*(j-1))=-r;
        A(n, i-1+nx*(j-1))=-r;
        b(n)=0;
```

```
    end
end
%A
%b

%转换为原本的u
U = A\b;
for i = 1:nx
    for j = 1:nt
        u(i,j) = U(i+nx*(j-1));
    end
end
%u

%画出数值解
[x,t] = meshgrid(0:hx:xf,0:ht:tf);
mesh(x,t,u)
```