**计算方法实验报告**

**姓名：王铭**

**学号：190110509**

**院系：计算机**

**专业：计算机**

**班级：1901105班**

**实验报告二**

|  |
| --- |
| 题目（摘要）  用Romberg积分法列出T-数表，根据T-数表来计算积分，并进行误差比较并确定最终的结果。    前言：（目的和意义）  实验目的：使用龙贝格(Romberg)积分法计算积分。  意义：在实际应用中，除了少数能够使用牛顿莱布尼茨公式解决的简单积分问题，遇到更多的是无法寻找到原函数或是能够找到原函数但原函数较为复杂，导致积分难以计算的情况。这种情况下，我们可以使用近似值来估计积分。Romberg积分法就是其中一种方法，它可以加快收敛速度，且根据其T-数表，可以判断近似值的误差。 |
| 数学原理  计算T-数表：  用表示T-数表中的元素，其中k表示对区间进行次划分，m表示在每个小区间上使用多少个点估计积分值，如当时表示使用复化的梯形公式来近似积分值。  对T数表的第一列元素，用以下两个公式计算：    根据第一列的值向后进行外推，使用下列公式计算：    由于每一列和每一对角线方向上的均收敛，用相邻两项的差来估计误差，当计算出相邻两项的差小于给定的误差时，可以判断结果已经收敛，此时停止计算。 |

|  |
| --- |
| 程序设计流程 |

|  |
| --- |
| 实验结果、结论与讨论  实验结果截图：  1.1  ，取N=6，精度theta=1e-6  输出停机标志，并返回矩阵M    故结果为0.7182818365  1.2  ，取N=6，精度theta=1e-6  输出停机标志，并返回矩阵M    结果为10.95017033  1.3  ,取N=6，精度theta=1e-6  输出停机标志，并返回矩阵M    结果为3.141592651  1.4  ,取N=6，精度theta=1e-6  输出停机标志，并返回矩阵M    结果为0.6931472103  实验结果  实验的四个题目均在给定精度内收敛，可以作为积分的近似值。  思考题  由结果矩阵M知，二分次数越大，精度越高。 |

**实验报告三**

|  |
| --- |
| 题目（摘要）  利用标准四阶Runge-Kutta方法求解给定的微分方程初值方程    前言：（目的和意义）  目的：采用Runge-Kutta方法用离散点上的解值来近似。  意义：Runge-Kutta法不求解微分方程的通项或近似表达式，通过选取步长h，求解一系列离散值来近似真实值。而标准的四阶Runge-Kutta方法的误差为，在一定误差范围的要求下，可以采用近似值来解决现实生活中遇到的难以求解或较为求解过程繁琐的的微分方程问题，且该方法可以通过编程利用计算机实现，求解速度快。 |
| 数学原理  ，其中：  利用该式一直递推到即可得到处的函数近似值。 |

|  |
| --- |
| 程序设计流程 |

|  |
| --- |
| 实验结果、结论与讨论  实验结果  题目1.1      结果为：    N=10时，结果为：    N=20时，结果为：    此微分方程的解析解为，误差较小。  题目1.2      结果为：    N=10时，结果为：    N=20时，结果为：    此微分方程的解析解为,误差较小  题目2.1    结果为    N=10时，结果为：    N=20时，结果为：    该问题的解析解为，误差较小。  题目2.2    结果为：    当N=10时，结果为：    N=20时，结果为：    该问题的解析解为，误差较小  题目3.1    结果为：    当N=10时，结果为：    当N=20时，结果为：    该问题的解析解为，误差较小。  题目3.2    结果为：    当N=10时，结果为：    当N=20时，结果为：    该问题的解析解为，当N=5时误差较大，N=10和N=20时误差较小  题目3.3    结果为：    当N=10时，结果为：    当N=20时，结果为：    该问题的解析解为，当N=5时误差较大，N=10和N=20时误差较小  实验结果  实验结果与微分方程的解析解基本相同，但当N=5时，带sinx和cosx等导数变化剧烈的函数值时，由于步长较长，易导致产生较大误差，当N取10和20时，误差较小。  讨论  ①数值解和解析解基本相同，因为使用了标准四阶Runge-Kutta方法，而该方法的误差为O(h5)。  ②通过对比N为5,10,20，显然N越大越精确。N越大，步长越小，故计算的误差越小。  ③N较小时会导致误差较大，因为N较小时，步长较大,sinx和cosx导数变化较剧烈，导致变化较大，产生较大误差。 |

**实验报告四**

|  |
| --- |
| 题目（摘要）  利用牛顿迭代法求方程的近似根。  前言：（目的和意义）  目的：利用牛顿迭代法求实际问题中无法得到准确解的方程。  意义：实际生活中遇到的某些方程如超越方程无法求其精确解，可以利用牛顿迭代法在给定的精度下求出其近似解来代替精确解。 |
| 数学原理  牛顿迭代法的计算公式为:  牛顿迭代法有局部收敛性，在其精确解的某一满足收敛性要求的邻域内取一点,根据牛顿迭代公式可得到收敛于的迭代序列，且收敛速度为2阶的；若,且,则对充分小的δ>0,当时，由牛顿迭代法的收敛速度是1阶的。 |

|  |
| --- |
| 程序设计流程  流程图：    每次迭代时先判断是否满足精度要求，满足则直接输出，否则判断是否小于，若是则输出奇异标志，否则进行迭代。迭代结束后根据来估计误差是否满足要求，若满足则返回迭代值，否则进行下一次迭代。 |

|  |
| --- |
| 实验结果、结论与讨论  程序运行结果截图：  问题1：  （1）    （2）    问题2：  （1）    （2）    结论  该实验的四个函数均能够从迭代，并最终收敛到根值，且都输出了在满足所给精度条件下根的近似值。  思考题  ①由于牛顿迭代法的局部收敛性，在选择初值时应使得该值在根的某一足够小的邻域内，使得从该点开始计算出的迭代序列收敛到根。在实际计算中，要根据结果是否收敛，通过判断f(a)\*f(b)是否小于0，并运用二分法找到满足收敛精度的迭代初值。  ②该题所求根为重根，牛顿迭代法在求重根的近似值时收敛速度为一阶，在使用matlab计算时，在一段时间内无法得出结果。原因：由于是符号标记，并未转换为数值，故运算速度较慢，可以用这一函数解决运算速度慢的问题。 |

**实验报告五**

|  |
| --- |
| 题目（摘要）  利用 Gauss 列主元消去法求解线性方程组。    前言：（目的和意义）  目的：通过化简系数矩阵的方法，求解线性方程组。  意义：高斯消元法通过化简系数矩阵为三角阵，可以简化求解线性方程组的步骤。而列选的高斯消元法，可以压制舍入误差对实验结果的影响，即通过每次选取待化简的一列中绝对值最大的数，来控制乘数因子不会出现大数吃小数的情况。 |
| 数学原理 |

|  |
| --- |
| 程序设计流程 |

|  |
| --- |
| 实验结果、结论与讨论  题目1：  （1）    （2）    （3）    （4）    题目二:  (1)    （2）    (3)    (4)    结论  实验中无奇异标志输出，以上线性方程均存在解，符合预期。 |

syms x;  
disp("1.1");  
a=0;  
b=1;  
f=x\*x\*exp(x);  
N=6;  
theta=1e-6;  
vpa(Romberg(a,b,N,theta,f),10)  
  
syms x;  
disp("1.2");  
a=1;  
b=3;  
N=6;  
f=exp(x)\*sin(x);  
theta=1e-6;  
vpa(Romberg(a,b,N,theta,f),10)  
  
syms x;  
disp("1.3");  
a=0;  
b=1;  
f=4/(1+x\*x);  
theta=1e-6;  
N=6;  
vpa(Romberg(a,b,N,theta,f),10)  
  
syms x;  
disp("1.4");  
a=0;  
b=1;  
f=1/(1+x);  
theta=1e-6;  
N=6;  
vpa(Romberg(a,b,N,theta,f),10)  
  
function M = Romberg(a,b,N,theta,f)  
f = matlabFunction(f);  
M=zeros(N);  
M(1,1)=(b-a)/2\*(f(a)+f(b));  
for i = 2:(N)  
 m=power(2,i-1);  
 h=(b-a)/m;  
 plus=0;  
 for j = (a+h):(2\*h):(b)  
 plus = plus + 2 \* h \* f(j);  
 end  
 M(i,1)=0.5\*M(i-1,1)+0.5\*plus;  
 if(abs(M(i,1)-M(i-1,1)) < theta)  
 disp("停机");  
 return;  
 end  
end  
for i =2:(N)  
 m=power(4,i-1);  
 M(i,i)=(m\*M(i,i-1)-M(i-1,i-1))/(m-1);  
 if(abs(M(i,i)-M(i-1,i-1))<theta)  
 disp("停机");  
 return;  
 end  
 for j = (i+1):(N)  
 M(j,i)=(m\*M(j,i-1)-M(j-1,i-1))/(m-1);  
 if(abs(M(j,i)-M(j-1,i))<theta)  
 disp("停机");  
 return;  
 end  
 end  
end  
end

1.1  
停机  
   
ans =  
   
[ 1.359140914, 0, 0, 0, 0, 0]  
[ 0.885660616, 0.7278338499, 0, 0, 0, 0]  
[0.7605963324, 0.7189082379, 0.7183131972, 0, 0, 0]  
[ 0.728890177, 0.7183214585, 0.7182823399, 0, 0, 0]  
[0.7209357789, 0.7182843129, 0.7182818365, 0, 0, 0]  
[0.7189454326, 0.7182819839, 0, 0, 0, 0]  
   
1.2  
停机  
   
ans =  
   
[ 5.12182642, 0, 0, 0, 0, 0]  
[9.279762907, 10.66574174, 0, 0, 0, 0]  
[10.52055428, 10.93415141, 10.95204539, 0, 0, 0]  
[10.84204347, 10.94920653, 10.9502102, 0, 0, 0]  
[10.92309389, 10.9501107, 10.95017097, 0, 0, 0]  
[10.94339842, 10.9501666, 10.95017033, 0, 0, 0]  
   
1.3  
停机  
   
ans =  
   
[ 3.0, 0, 0, 0, 0, 0]  
[ 3.1, 3.133333333, 0, 0, 0, 0]  
[3.131176471, 3.141568627, 0, 0, 0, 0]  
[3.138988494, 3.141592502, 0, 0, 0, 0]  
[3.140941612, 3.141592651, 0, 0, 0, 0]  
[3.141429893, 0, 0, 0, 0, 0]  
   
1.4  
停机  
   
ans =  
   
[ 0.75, 0, 0, 0, 0, 0]  
[0.7083333333, 0.6944444444, 0, 0, 0, 0]  
[0.6970238095, 0.6932539683, 0, 0, 0, 0]  
[0.6941218504, 0.6931545307, 0, 0, 0, 0]  
[0.6933912022, 0.6931476528, 0, 0, 0, 0]  
[0.6932082083, 0.6931472103, 0, 0, 0, 0]

[*Published with MATLAB® R2020b*](https://www.mathworks.com/products/matlab)

syms x;  
syms y;  
disp("1.1");  
a=0;  
b=1;  
alpha=-1;  
N=5;  
f(x,y)=x+y;  
Runge\_Kutta(f,a,b,alpha,N)  
N=10;  
Runge\_Kutta(f,a,b,alpha,N)  
N=20;  
Runge\_Kutta(f,a,b,alpha,N)  
  
syms x;  
syms y;  
disp("1.2");  
a=0;  
b=1;  
alpha=1;  
N=5;  
f(x,y)=-y\*y;  
Runge\_Kutta(f,a,b,alpha,N)  
N=10;  
Runge\_Kutta(f,a,b,alpha,N)  
N=20;  
Runge\_Kutta(f,a,b,alpha,N)  
  
syms x;  
syms y;  
disp("2.1");  
a=1;  
b=3;  
alpha=0;  
N=5;  
f(x,y)=2\*y/x+x\*x\*exp(x);  
Runge\_Kutta(f,a,b,alpha,N)  
N=10;  
Runge\_Kutta(f,a,b,alpha,N)  
N=20;  
Runge\_Kutta(f,a,b,alpha,N)  
  
syms x;  
syms y;  
disp("2.2");  
a=1;  
b=3;  
alpha=-2;  
N=5;  
f(x,y)=(y+y\*y)/x;  
Runge\_Kutta(f,a,b,alpha,N)  
N=10;  
Runge\_Kutta(f,a,b,alpha,N)  
N=20;  
Runge\_Kutta(f,a,b,alpha,N)  
  
syms x;  
syms y;  
disp("3.1");  
a=0;  
b=1;  
alpha=1/3;  
N=5;  
f(x,y)=-20\*(y-x\*x)+2\*x;  
Runge\_Kutta(f,a,b,alpha,N)  
N=10;  
Runge\_Kutta(f,a,b,alpha,N)  
N=20;  
Runge\_Kutta(f,a,b,alpha,N)  
  
syms x;  
syms y;  
disp("3.2");  
a=0;  
b=1;  
alpha=1;  
N=5;  
f(x,y)=-20\*y+20\*sin(x)+cos(x);  
Runge\_Kutta(f,a,b,alpha,N)  
N=10;  
Runge\_Kutta(f,a,b,alpha,N)  
N=20;  
Runge\_Kutta(f,a,b,alpha,N)  
  
syms x;  
syms y;  
disp("3.3");  
a=0;  
b=1;  
alpha=0;  
N=5;  
f(x,y)=-20\*(y-exp(x)\*sin(x))+exp(x)\*(sin(x)+cos(x));  
Runge\_Kutta(f,a,b,alpha,N)  
N=10;  
Runge\_Kutta(f,a,b,alpha,N)  
N=20;  
Runge\_Kutta(f,a,b,alpha,N)  
  
function M = Runge\_Kutta(f,a,b,alpha,N)  
 f = matlabFunction(f);  
 h = (b - a) / N;  
 M=zeros(N+1,2);  
 M(1,1)=a;  
 M(1,2)=alpha;  
 for i=2:(N+1)  
 M(i,1)=M(i-1,1)+h;  
 K1=f(M(i-1,1),M(i-1,2));  
 K2=f((M(i-1,1)+h/2),(M(i-1,2)+h\*K1/2));  
 K3=f((M(i-1,1)+h/2),(M(i-1,2)+h\*K2/2));  
 K4=f((M(i-1,1)+h),(M(i-1,2)+h\*K3));  
 M(i,2)=M(i-1,2)+(K1+2\*K2+2\*K3+K4)\*h/6;  
 end  
end

1.1  
  
ans =  
  
 0 -1.0000  
 0.2000 -1.2000  
 0.4000 -1.4000  
 0.6000 -1.6000  
 0.8000 -1.8000  
 1.0000 -2.0000  
  
  
ans =  
  
 0 -1.0000  
 0.1000 -1.1000  
 0.2000 -1.2000  
 0.3000 -1.3000  
 0.4000 -1.4000  
 0.5000 -1.5000  
 0.6000 -1.6000  
 0.7000 -1.7000  
 0.8000 -1.8000  
 0.9000 -1.9000  
 1.0000 -2.0000  
  
  
ans =  
  
 0 -1.0000  
 0.0500 -1.0500  
 0.1000 -1.1000  
 0.1500 -1.1500  
 0.2000 -1.2000  
 0.2500 -1.2500  
 0.3000 -1.3000  
 0.3500 -1.3500  
 0.4000 -1.4000  
 0.4500 -1.4500  
 0.5000 -1.5000  
 0.5500 -1.5500  
 0.6000 -1.6000  
 0.6500 -1.6500  
 0.7000 -1.7000  
 0.7500 -1.7500  
 0.8000 -1.8000  
 0.8500 -1.8500  
 0.9000 -1.9000  
 0.9500 -1.9500  
 1.0000 -2.0000  
  
1.2  
  
ans =  
  
 0 1.0000  
 0.2000 0.8333  
 0.4000 0.7143  
 0.6000 0.6250  
 0.8000 0.5556  
 1.0000 0.5000  
  
  
ans =  
  
 0 1.0000  
 0.1000 0.9091  
 0.2000 0.8333  
 0.3000 0.7692  
 0.4000 0.7143  
 0.5000 0.6667  
 0.6000 0.6250  
 0.7000 0.5882  
 0.8000 0.5556  
 0.9000 0.5263  
 1.0000 0.5000  
  
  
ans =  
  
 0 1.0000  
 0.0500 0.9524  
 0.1000 0.9091  
 0.1500 0.8696  
 0.2000 0.8333  
 0.2500 0.8000  
 0.3000 0.7692  
 0.3500 0.7407  
 0.4000 0.7143  
 0.4500 0.6897  
 0.5000 0.6667  
 0.5500 0.6452  
 0.6000 0.6250  
 0.6500 0.6061  
 0.7000 0.5882  
 0.7500 0.5714  
 0.8000 0.5556  
 0.8500 0.5405  
 0.9000 0.5263  
 0.9500 0.5128  
 1.0000 0.5000  
  
2.1  
  
ans =  
  
 1.0000 0  
 1.4000 2.6139  
 1.8000 10.7763  
 2.2000 30.4917  
 2.6000 72.5856  
 3.0000 156.2252  
  
  
ans =  
  
 1.0000 0  
 1.2000 0.8664  
 1.4000 2.6197  
 1.6000 5.7199  
 1.8000 10.7920  
 2.0000 18.6809  
 2.2000 30.5216  
 2.4000 47.8324  
 2.6000 72.6345  
 2.8000 107.6089  
 3.0000 156.2983  
  
  
ans =  
  
 1.0000 0  
 1.1000 0.3459  
 1.2000 0.8666  
 1.3000 1.6072  
 1.4000 2.6203  
 1.5000 3.9676  
 1.6000 5.7209  
 1.7000 7.9638  
 1.8000 10.7935  
 1.9000 14.3229  
 2.0000 18.6829  
 2.1000 24.0250  
 2.2000 30.5244  
 2.3000 38.3835  
 2.4000 47.8359  
 2.5000 59.1510  
 2.6000 72.6389  
 2.7000 88.6566  
 2.8000 107.6143  
 2.9000 129.9833  
 3.0000 156.3048  
  
2.2  
  
ans =  
  
 1.0000 -2.0000  
 1.4000 -1.5540  
 1.8000 -1.3836  
 2.2000 -1.2934  
 2.6000 -1.2375  
 3.0000 -1.1995  
  
  
ans =  
  
 1.0000 -2.0000  
 1.2000 -1.7142  
 1.4000 -1.5555  
 1.6000 -1.4545  
 1.8000 -1.3846  
 2.0000 -1.3333  
 2.2000 -1.2941  
 2.4000 -1.2631  
 2.6000 -1.2381  
 2.8000 -1.2174  
 3.0000 -1.2000  
  
  
ans =  
  
 1.0000 -2.0000  
 1.1000 -1.8333  
 1.2000 -1.7143  
 1.3000 -1.6250  
 1.4000 -1.5556  
 1.5000 -1.5000  
 1.6000 -1.4545  
 1.7000 -1.4167  
 1.8000 -1.3846  
 1.9000 -1.3571  
 2.0000 -1.3333  
 2.1000 -1.3125  
 2.2000 -1.2941  
 2.3000 -1.2778  
 2.4000 -1.2632  
 2.5000 -1.2500  
 2.6000 -1.2381  
 2.7000 -1.2273  
 2.8000 -1.2174  
 2.9000 -1.2083  
 3.0000 -1.2000  
  
3.1  
  
ans =  
  
 1.0e+03 \*  
  
 0 0.0003  
 0.0002 0.0018  
 0.0004 0.0088  
 0.0006 0.0437  
 0.0008 0.2173  
 0.0010 1.0843  
  
  
ans =  
  
 0 0.3333  
 0.1000 0.1228  
 0.2000 0.0793  
 0.3000 0.1048  
 0.4000 0.1666  
 0.5000 0.2539  
 0.6000 0.3630  
 0.7000 0.4927  
 0.8000 0.6426  
 0.9000 0.8125  
 1.0000 1.0025  
  
  
ans =  
  
 0 0.3333  
 0.0500 0.1276  
 0.1000 0.0569  
 0.1500 0.0402  
 0.2000 0.0467  
 0.2500 0.0651  
 0.3000 0.0910  
 0.3500 0.1229  
 0.4000 0.1602  
 0.4500 0.2026  
 0.5000 0.2501  
 0.5500 0.3026  
 0.6000 0.3601  
 0.6500 0.4226  
 0.7000 0.4901  
 0.7500 0.5626  
 0.8000 0.6401  
 0.8500 0.7226  
 0.9000 0.8101  
 0.9500 0.9026  
 1.0000 1.0001  
  
3.2  
  
ans =  
  
 1.0e+03 \*  
  
 0 0.0010  
 0.0002 0.0052  
 0.0004 0.0254  
 0.0006 0.1255  
 0.0008 0.6253  
 0.0010 3.1238  
  
  
ans =  
  
 0 1.0000  
 0.1000 0.4331  
 0.2000 0.3097  
 0.3000 0.3323  
 0.4000 0.4014  
 0.5000 0.4831  
 0.6000 0.5654  
 0.7000 0.6440  
 0.8000 0.7167  
 0.9000 0.7825  
 1.0000 0.8405  
  
  
ans =  
  
 0 1.0000  
 0.0500 0.4250  
 0.1000 0.2405  
 0.1500 0.2022  
 0.2000 0.2184  
 0.2500 0.2548  
 0.3000 0.2983  
 0.3500 0.3439  
 0.4000 0.3898  
 0.4500 0.4351  
 0.5000 0.4795  
 0.5500 0.5227  
 0.6000 0.5646  
 0.6500 0.6052  
 0.7000 0.6442  
 0.7500 0.6816  
 0.8000 0.7173  
 0.8500 0.7513  
 0.9000 0.7833  
 0.9500 0.8134  
 1.0000 0.8414  
  
3.3  
  
ans =  
  
 0 0  
 0.2000 0.2986  
 0.4000 0.9272  
 0.6000 2.8355  
 0.8000 10.7109  
 1.0000 47.9414  
  
  
ans =  
  
 0 0  
 0.1000 0.1121  
 0.2000 0.2451  
 0.3000 0.4018  
 0.4000 0.5841  
 0.5000 0.7938  
 0.6000 1.0324  
 0.7000 1.3010  
 0.8000 1.6003  
 0.9000 1.9305  
 1.0000 2.2912  
  
  
ans =  
  
 0 0  
 0.0500 0.0526  
 0.1000 0.1104  
 0.1500 0.1737  
 0.2000 0.2427  
 0.2500 0.3178  
 0.3000 0.3990  
 0.3500 0.4867  
 0.4000 0.5811  
 0.4500 0.6823  
 0.5000 0.7906  
 0.5500 0.9061  
 0.6000 1.0290  
 0.6500 1.1594  
 0.7000 1.2974  
 0.7500 1.4432  
 0.8000 1.5966  
 0.8500 1.7579  
 0.9000 1.9268  
 0.9500 2.1034  
 1.0000 2.2875

[*Published with MATLAB® R2020b*](https://www.mathworks.com/products/matlab)

syms x;  
disp("1.1");  
e1 = 1e-6;  
e2 = 1e-4;  
N = 10;  
x0 = pi/4;  
f = cos(x)-x;  
vpa(Newton(f,x0,e1,e2,N),10)  
  
syms x;  
disp("1.2");  
e1 = 1e-6;  
e2 = 1e-4;  
N = 10;  
f=exp(-x)-sin(x);  
x0=0.6;  
vpa(Newton(f,x0,e1,e2,N),10)  
  
syms x;  
disp("1.3");  
e1 = 1e-6;  
e2 = 1e-4;  
N = 10;  
f=x-exp(-x);  
x0=0.5;  
vpa(Newton(f,x0,e1,e2,N),10)  
  
  
syms x;  
disp("1.4");  
e1 = 1e-6;  
e2 = 1e-4;  
f=x\*x-2\*x\*exp(-x)+exp(-2\*x);  
N=20;  
x0=0.5;  
vpa(Newton(f,x0,e1,e2,N),10)  
  
function x1 = Newton(f,x0,e1,e2,N)  
 df = diff(f);  
 f = matlabFunction(f);  
 df = matlabFunction(df);  
 for i = 1:N  
 F = f(x0);  
 DF = df(x0);  
 if(abs(F) < e1)  
 x1 = x0;  
 return;  
 end  
 if(abs(DF)< e2)  
 x1 = -1;  
 disp("失败");  
 return;  
 end  
 x1 = x0 - F/DF;  
 if(abs(x0-x1)<e1)  
 return;  
 end  
 x0 = x1;  
 end  
 x1 = -1;  
 disp("失败");  
end

1.1  
   
ans =  
   
0.7390851781  
   
1.2  
   
ans =  
   
0.5885327428  
   
1.3  
   
ans =  
   
0.567143165  
   
1.4  
   
ans =  
   
0.5666057041

[*Published with MATLAB® R2020b*](https://www.mathworks.com/products/matlab)

n=4;  
disp("1.1");  
A=[0.4096,0.1234,0.3678,0.2943;0.2246,0.3872,0.4015,0.1129;0.3645,0.1920,0.3781,0.0643;0.1784,0.4002,0.2786,0.3927];  
B=[1.1951;1.1262;0.9989;1.2499];  
vpa(Gauss(A,B,n),10)  
  
n=4;  
disp("1.2");  
A=[136.01,90.86,0,0;90.86,98.81,-67.59,0;0,-67.59,132.01,46.26;0,0,46.26,177.17];  
B=[226.87;122.08;110.68;223.43];  
vpa(Gauss(A,B,n),10)  
  
n=4;  
disp("1.3");  
A=[1,1/2,1/3,1/4;1/2,1/3,1/4,1/5;1/3,1/4,1/5,1/6;1/4,1/5,1/6,1/7];  
B=[25/12;77/60;57/60;319/420];  
vpa(Gauss(A,B,n),10)  
  
n=4;  
disp("1.4");  
A=[10,7,8,7;7,5,6,5;8,6,10,9;7,5,9,10];  
B=[32;23;33;31];  
vpa(Gauss(A,B,n),10)  
  
n=4;  
disp("2.1");  
A=[197,305,-206,-804;46.8,71.3,-47.4,52;88.6,76.4,-10.8,802;1.45,5.9,6.13,36.5];  
B=[136;11.7;25.1;6.6];  
vpa(Gauss(A,B,n),10)  
  
n=4;  
disp("2.2");  
A=[0.5398,0.7161,-0.5554,-0.2982;0.5257,0.6924,0.3565,-0.6255;0.6465,-0.8187,-0.1872,0.1291;0.5814,0.94,-0.7779,-0.4042];  
B=[0.2058;-0.0503;0.107;0.1859];  
vpa(Gauss(A,B,n),10)  
  
n=3;  
disp("2.3");  
A=[10,1,2;1,10,2;1,1,5];  
B=[13;13;7];  
vpa(Gauss(A,B,n),10)  
  
n=3;  
disp("2.4");  
A=[4,-2,-4;-2,17,10;-4,10,9];  
B=[-2;25;15];  
vpa(Gauss(A,B,n),10)  
  
function X = Gauss(A,B,n)  
X = linspace(0,0,n);  
for i = 1 : n  
 max = abs(A(i,i));  
 maxi = i;  
 %找max a[k][i]  
 for j = (i + 1) : n  
 if(abs(A(j,i)) > abs(max))  
 maxi = j;  
 max = A(j,i);  
 end  
 end  
 if(max == 0)  
 disp("奇异标志");  
 return;  
 end  
 if(maxi ~= i)  
 %交换B  
 temp = B(i);  
 B(i) = B(maxi);  
 B(maxi) = temp;  
 for j = i : n  
 temp = A(i,j);  
 A(i,j) = A(maxi,j);  
 A(maxi,j) = temp;  
 end  
 end  
 %化简  
 for j = i+1 : n  
 m = A(j,i)/A(i,i);  
 B(j) = B(j) - m\*B(i);  
 for k = i : n  
 A(j,k) = A(j,k) - m\*A(i,k);  
 end  
 end  
 if(A(n,n) == 0)  
 disp("奇异标志")  
 return;  
 end  
end  
 %回代  
 X(n) = B(n) / A(n,n);  
 for i = n-1 :-1 : 1  
 X(i) = B(i);  
 for j = i+1 :n  
 X(i) = X(i) - X(j) \* A(i,j);  
 end  
 X(i) = X(i)/A(i,i);  
 end  
end

1.1  
   
ans =  
   
[1.0, 1.0, 1.0, 1.0]  
   
1.2  
   
ans =  
   
[1.0, 1.0, 1.0, 1.0]  
   
1.3  
   
ans =  
   
[1.0, 1.0, 1.0, 1.0]  
   
1.4  
   
ans =  
   
[1.0, 1.0, 1.0, 1.0]  
   
2.1  
   
ans =  
   
[0.9536791069, 0.3209568455, 1.078708076, -0.09010850954]  
   
2.2  
   
ans =  
   
[0.516177298, 0.4152194728, 0.1099661029, 1.036539223]  
   
2.3  
   
ans =  
   
[1.0, 1.0, 1.0]  
   
2.4  
   
ans =  
   
[1.0, 1.0, 1.0]

[*Published with MATLAB® R2020b*](https://www.mathworks.com/products/matlab)