**上海电力学院**

**数值计算上机报告**



课 程： 现代数值计算

题 目： 数值计算方法上机实习题报告

院 系： 计算机科学与技术学院

专业年级： 19电力信息技术

学生姓名： 崔荣成 学号： 19108002

指导教师： 黄建雄

2019年11月27日

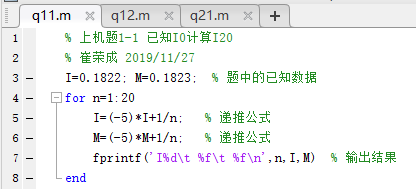
数值计算方法上机实习题

1. 设，
2. 由递推公式，从, 出发，计算；
3. ，，用，计算；
4. 分析结果的可靠性及产生此现象的原因(重点分析原因)。

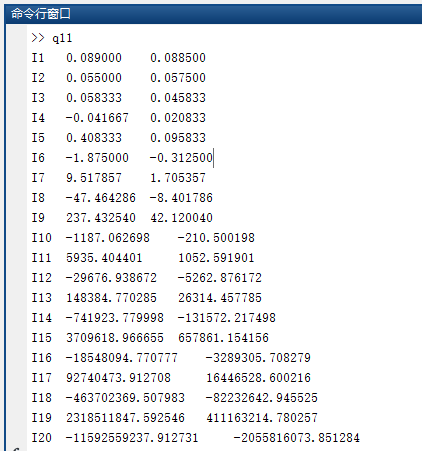
**解：（1）**

**解题思路：**给赋一个初值，然后通过for循环实现递推公式的递推计算，通过20次的循环最终计算出的值。

**程序代码：**



**运行结果：**

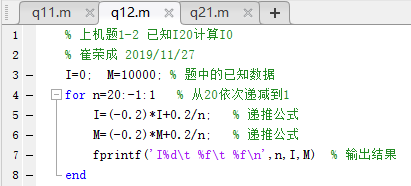


根据最后运行结果，当时，；而当时，。

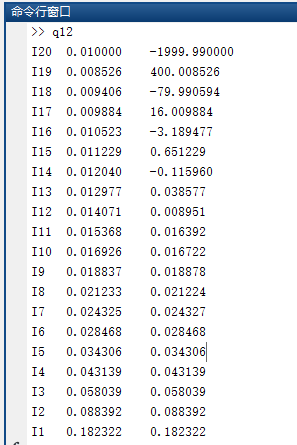
**（2）**

**解题思路：**先给赋初值，然后同样的通过for循环实现递推公式的递推计算，通过20次的（反向）循环最终计算出的值。

**程序代码：**



**运行结果：**



根据最后运行结果，当时，，并且当时，。

**（3）分析结果的可靠性及产生此现象的原因(重点分析原因)。**

从（1）的程序运行结果来看，的初值虽然相差不大，但是通过20次的递推迭代计算，最后的结果相差很大，结果不可靠。

从（2）的程序运行结果来看，即使的初值相差较大，但是通过20次递推迭代计算，得到的最后结果基本一致，结果是可靠的。

原因分析：当初值不同的时候，随着迭代步数的增加，采用正迭代时候初值误差会被放大，所推出的值误差就会越来越大。而当使用倒迭代的时候，误差是在不断缩小的，最后计算的结果误差也很小。

假设两个初值分别为，对于（1）中的算法，由迭代公式，有

因此，每迭代一步，误差就会放大5倍，算法不稳定，结果不可靠。

但是，对于（2）中的算法，当初值不同的时候，随着迭代步数的增加，采用倒迭代时候两初值所推出的

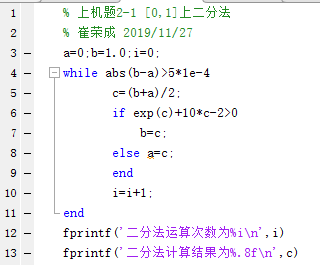
，以此类推，每迭代一步误差就会缩小5倍，虽然初值误差很大，但是随着n的增大，误差反而越来越小，算法稳定，结果可靠。

1. 求方程的近似根，要求，并比较计算量。
2. 在[0，1]上用二分法；
3. 取初值，并用迭代；
4. 加速迭代的结果；
5. 取初值，并用牛顿迭代法；
6. 分析绝对误差。

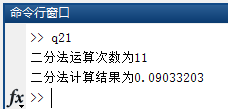
**解：**

**（1）在[0，1]上用二分法；**

**程序代码：**



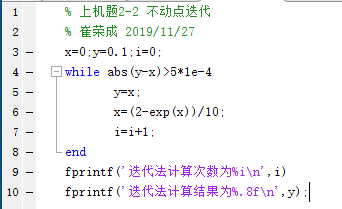
**运行结果：**



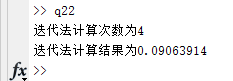
方程的近似根为：x\*=（a+b）/2 = 0.09033203 。步长为i=11。

**（2）取初值，并用迭代；**

**程序代码：**



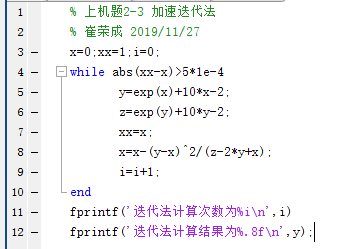
**运行结果：**



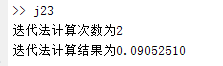
方程的近似根为：x=0.09063914 。步长为i=4 。

**（3）加速迭代的结果；**

**程序代码：**



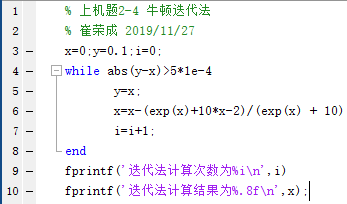
**运行结果：**



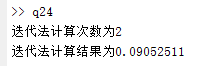
方程的近似根为：x= 0.09052510。步长为i=2 。

**（4）取初值，并用牛顿迭代法；**

**程序代码：**



**运行结果：**



方程的近似根为：x= 0.09052511。步长为i=2 。

**（5）分析绝对误差。**

**计算精确解：**

通过MATLAB指令求精确解：

solve('exp(x)+10\*x-2=0')

方程的精确解为x= 0.09052510130725。

**计算各种计算方法的绝对误差：**

二分法：

不动点迭代：

加速迭代：

牛顿法：

**误差分析：**

在统一标准精度的下，选取不同的迭代方法，迭代速度有很大的不同。究其原因主要是收敛阶在起作用。二分法与不动点法的收敛阶是线性收敛阶，加速迭代和牛顿迭代是二阶收敛阶。所以，在上面的实验中，我们可以看到同一标准精度的要求下，二分法迭代了11次，不动点迭代了4次，加速迭代和牛顿迭代都是2次，二阶迭代速度明显大于一阶。绝对误差来看，二分法的绝对误差是最大的。而加速迭代的绝对误差最小。

3．钢水包使用次数多以后，钢包的容积增大，数据如下：

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| y | 6.42 | 8.2 | 9.58 | 9.5 | 9.7 | 10 | 9.93 | 9.99 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 10.49 | 10.59 | 10.60 | 10.8 | 10.6 | 10.9 | 10.76 |

试从中找出使用次数和容积之间的关系，计算均方差。（注：用拟合）

**解：**

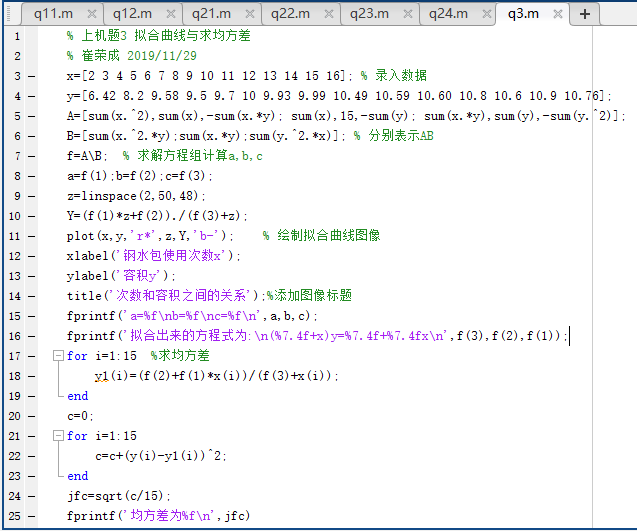
**解题思路：**由题意可知待拟合的曲线模型是，将模型变为，分析可以采用非线性最小二程法。按照最小二乘原理，应选取参数a，b，c使得表达式达到极小值。计算Z关于a、b、c的偏导数，并令这些偏导数等于0，相对应的方程组如下所示：

通过对上述的偏导数方程组进行整理，可以将之写成的的形式，

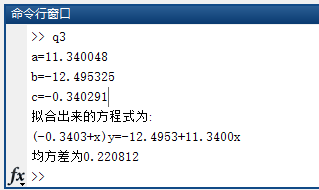
即

利用 MATLA进行的求解，对应的就是a、b、c三个参数的值。

**程序代码：**



**运行结果：**



****

**结论分析：**因为误差题中的指标为，其驻点方程组为非线性方程组，求解比较困难，通过改进指标后：，得到驻点方程组为线性方程组，更便于计算。

4．设，，

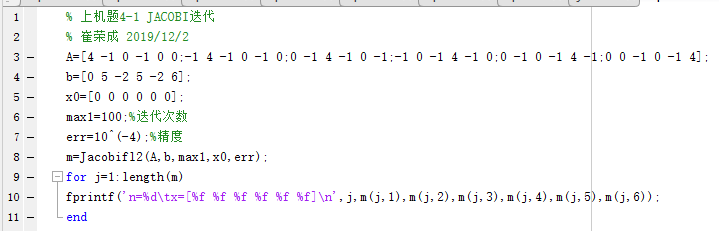
分析下列迭代法的收敛性，并求的近似解及相应的迭代次数。

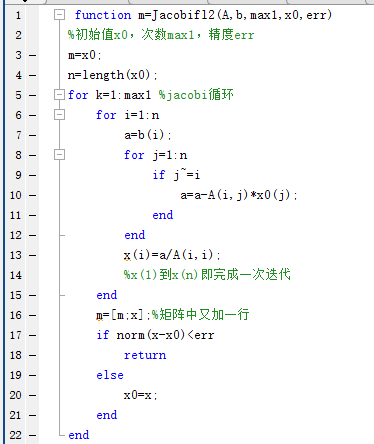
* 1. JACOBI迭代；
  2. GAUSS-SEIDEL迭代；
  3. SOR迭代（取，找到迭代步数最少的）。

**解：**

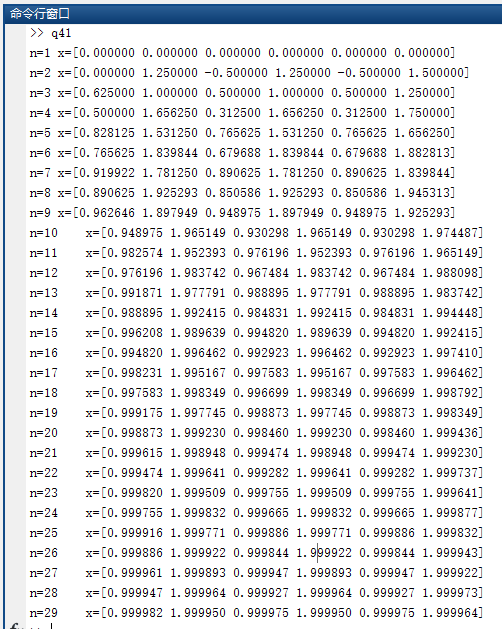
* 1. **JACOBI迭代；**

**程序代码：**





**运行结果：**

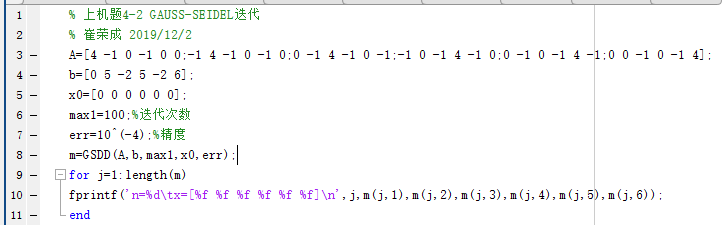


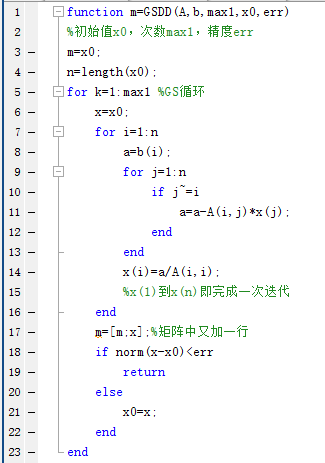
**迭代分析：**

从运行结果可以看出共迭代29次，每次迭代是将A按照LDU分解为下三角阵+对角阵+上三角阵的形式，然后在逐次迭代。计算公式比较简单，每次迭代只计算一次矩阵和向量的相乘，且计算过程中矩阵A不变。但是收敛速度较慢，且占用的空间内存大。

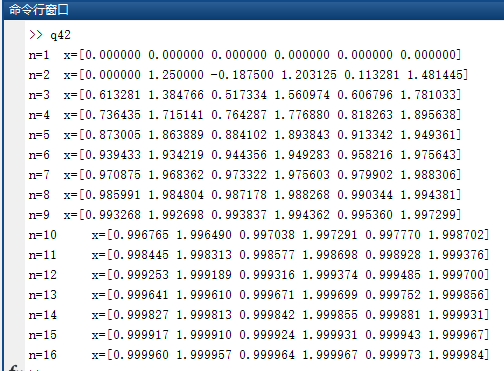
* 1. **GAUSS-SEIDEL迭代；**

**程序代码：**





**运行结果：**

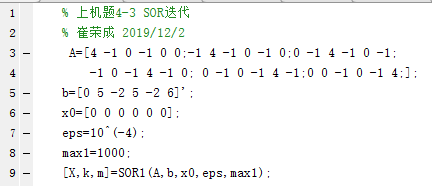


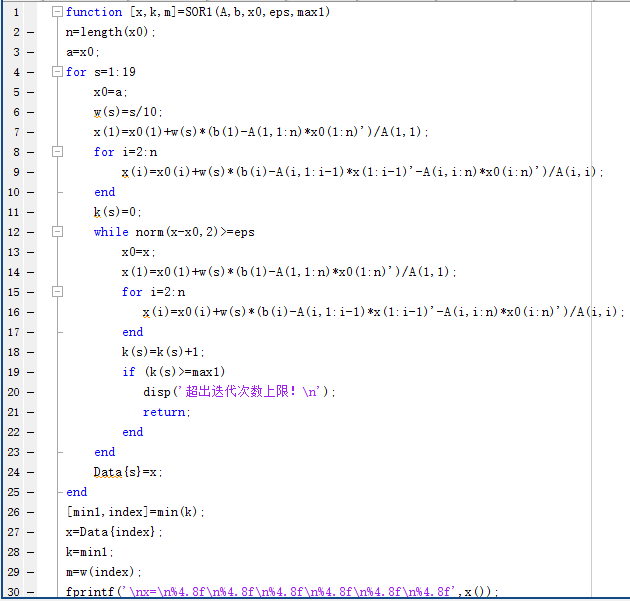
**迭代分析：**

从运行结果可以看出共迭代16次，每次迭代也是进行矩阵分解，是Jacobi迭代的一种修正方法，收敛速度要快，但是只能在一定条件下收敛，容易发散。

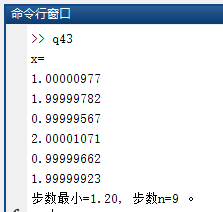
* 1. **SOR迭代（取，找到迭代步数最少的）。**

**程序代码：**





**运行结果：**



**迭代比较：**

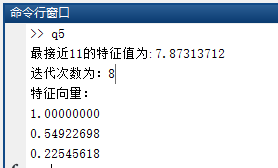
比较三种迭代方法，Jacobi迭代法迭代次数是29次，Gauss-Seidel迭代法迭代次数是16，速度要比Jacobi迭代法快。而在SOR迭代中选取不同的松弛因子迭代的次数各不相同且可能会相差很大。所以改变松弛因子的值，不仅会影响迭代过程的收敛速度，也会影响其收敛性。

1. 用逆幂迭代法求最接近于11的特征值和特征向量，准确到。

**程序代码：**



**运行结果：**



6．用经典R-K方法求解初值问题

（1），， ；

（2），， 。

和精确解比较，分析结论。

**解：**

**（1）**

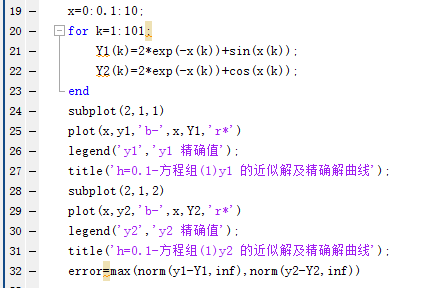
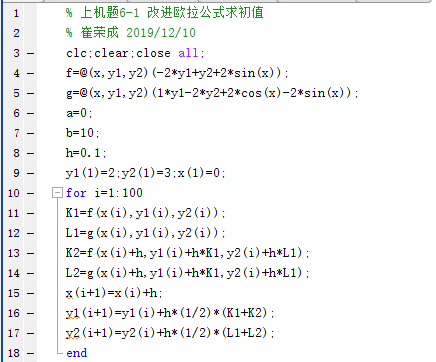
**解题思路：**

由式，， 可得 初值条件为

则由经典R-K方法可得算法为

1. **取步长h=0.1，用改进欧拉计算**

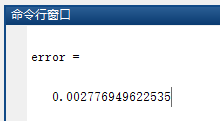
**程序代码**：



**运行结果：**



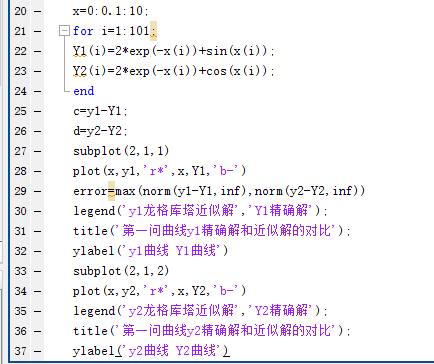
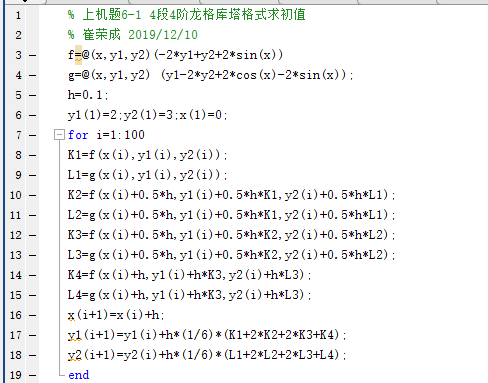
**误差结果：**



通过图像看不太出误差，但是从放大图像，或者从输出结果可以看出还是有误差存在，改用四段四阶龙格库塔公式求解。

1. **取步长h=0.1，用四段四阶龙格库塔计算**

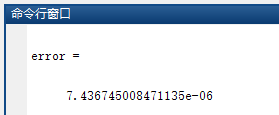
**程序代码：**



**运行结果：**



**误差结果：**



**（2）**

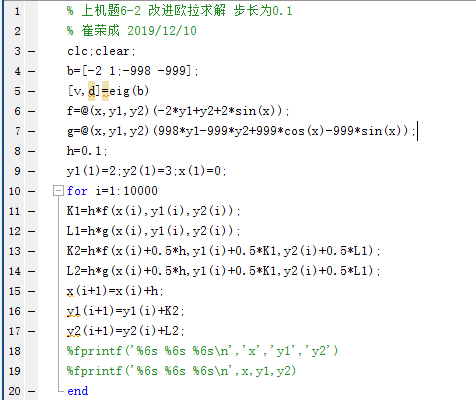
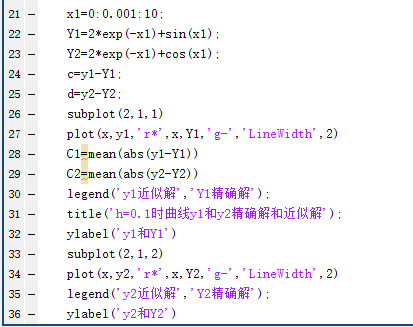
**解题思路：**

由式，， 可得 初值条件为

则由经典R-K方法可得算法为

**①长h=0.1**

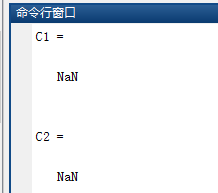
**程序代码：**

**运行结果：**

****

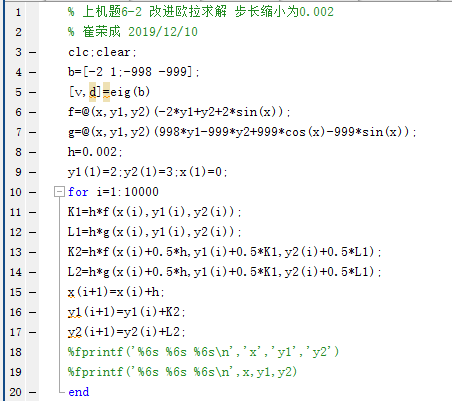
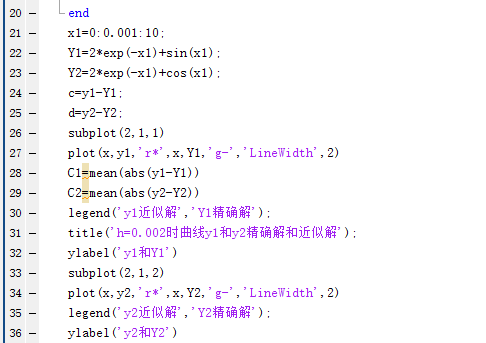
**误差结果：**



从图像和输出结果可以看出，此时误差值无法求解，出现刚性问题，选择缩小步长进行实验。

**②缩小步长为0.002**

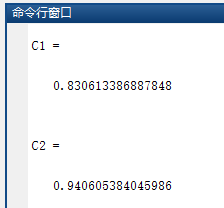
**程序代码：**

**运行结果：**

****

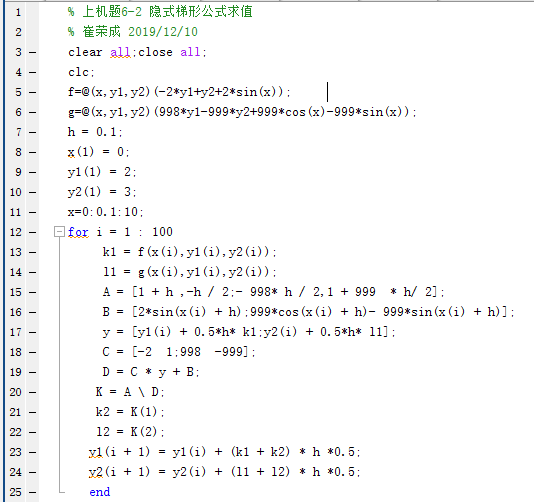
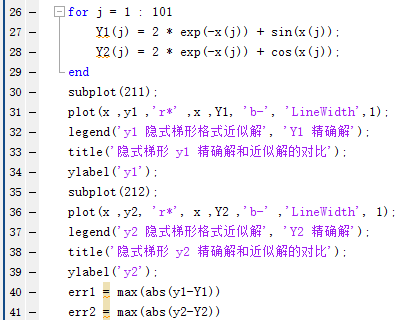
**误差结果：**



从图像中可以看出，虽然能够进行求解，但是误差还很大，缩小步长导致计算量增大，而且误差不可控。此时选用隐式公式来进行求解。

**③换成隐式梯形公式**

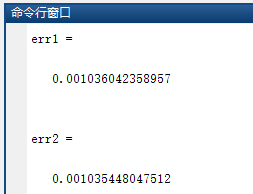
**程序代码：**

**运行结果：**

****

**误差结果：**



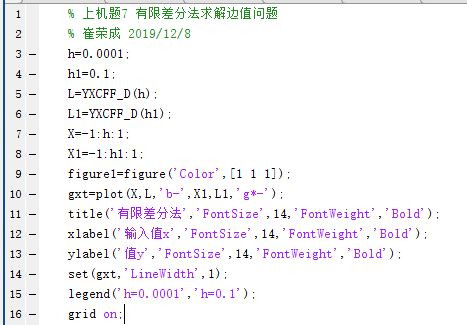
7．用有限差分法求解边值问题（h=0.1）：

.

把上式改写成标准形式：

其中f(x)=0，q(x)=1+x^2。有限元差分法的MATLAB实现如下：

**程序代码：**



**运行结果：**

****