合肥工業大學

《计算方法》

实 验 指 导 书

计算机学院

前　言

《计算方法》所涉及的都是可以直接在计算机上求解的数值方法。这些方法既有数学的抽象性与严密的科学性，又有应用的广泛性与实验的高度技术性。故结合课堂教学，本课程配备适当的上机实验(8 个学时)，通过实验可以加强学生对数学模型的总体分析，算法选取，上机调试和结果分析等环节的训练。

本实验指导书共包含 5 个实验，要求学生在 8 个实验课时内完成，并提交实验报告,以此可作为《计算方法》课程成绩评定的一部分。

目　录

前　言.....2
实验一　　插值与拟合.....4
1　实验目的.....4
2　实验内容.....4
3　算法基本原理.....4
4　算法描述.....5
5　计算用例的参考输出.....5
6 思考题.....5
实验二　　数值积分.....6
1 实验目的.....6
2　实验内容.....6
3　算法基本原理.....7
4　算法描述.....8
5 计算用例的参考输出.....9
6 思考题.....9
实验三　　数值微分.....9
1　实验目的.....9
2　实验内容.....9
3　算法基本原理.....10
4　算法描述.....10
5 计算用例的参考输出.....11
6 思考题.....12
实验四　　非线性方程求根迭代法.....13
1　实验目的.....13
2　实验内容.....13
3　算法基本原理.....13
4　算法描述.....14
5 计算用例的参考输出.....14
6　思考题.....14
实验五　　求解线性方程组.....15
1 实验目的.....15
2　实验内容.....15
3　算法基本原理.....15
4　算法描述.....16
5 计算用例的参考输出.....17
6 思考题.....17
实验报告内容要求.....18

实验一 插值与拟合

1 实验目的

(1) 明确插值多项式和分段插值多项式各自的优缺点；
(2) 编程实现拉格朗日插值算法，分析实验结果体会高次插值产生的龙格现象；
(3) 理解最小二乘拟合，并编程实现线性拟合，掌握非线性拟合转化为线性拟合的方法
(4) 运用常用的插值和拟合方法解决实际问题。

2 实验内容

(1) 对于 $\boldsymbol{f(x)=\frac{1}{1+x^2},\ -5\le x\le 5}$

要求选取 11 个等距插值节点，分别采用拉格朗日插值和分段线性插值，计算 x 为 0.5, 4.5 处的函数值并将结果与精确值进行比较。

输入：区间长度，n(即 n+1 个节点)，预测点

输出：预测点的近似函数值，精确值，及误差

（2）已知 $\sqrt{1}=1,\sqrt{4}=2,\sqrt{9}=3,$ 用牛顿插值公式求 $\sqrt{5}$ 的近似值。

输入：数据点集，预测点。

输出：预测点的近似函数值

3 算法基本原理

当精确函数 y = f(x) 非常复杂或未知，在一系列节点 x_0 … x_n 处测得函数值 $y_0 = f(x_0)$, … $y_n = f(x_n)$，希望由此构造一个简单易算的近似函数 $g(x) \approx f(x)$，满足条件 $g(x_i) = f(x_i)$ (i = 0, … n)。这里的 g(x) 称为 f(x) 的插值函数，由插值函数可以去近似估计 f(x)在一些未知点处的函数值。最常用的插值函数是多项式插值。

所谓多项式插值即求 n 次多项式 $P_n(x)=a_0+a_1x+\cdots+a_nx^n$ 使得 $\boldsymbol{P_n(x_i)=f(x_i)}$

基于基函数的拉格朗日插值是构造多项式插值最基本方法。也是推导数值微积分和微分方程数值解的公式的理论基础。

拉格朗日插值公式为： $\boldsymbol{p_n(x)=\sum_{k=0}^{n}y_kl_k(x)}$ 其中 $\boldsymbol{l_k(x)=\prod_{\substack{j=0\\ j\neq k}}^{n}\frac{x-x_j}{x_k-x_j}}$

当结点比较多，次数较高的插值多项式往往发生 Runge 现象，分段低次插值是避免 Runge 现象的重要手段。分段一次插值将整个区间分段，在每个小区间上，用一次多项式逼近 f (x)，直观上即用折线代替曲线，只要区间足够小就可以保证很好的逼近效果，但曲线缺乏光滑性。

4 算法描述

(1) 拉格朗日插值算法见流程图

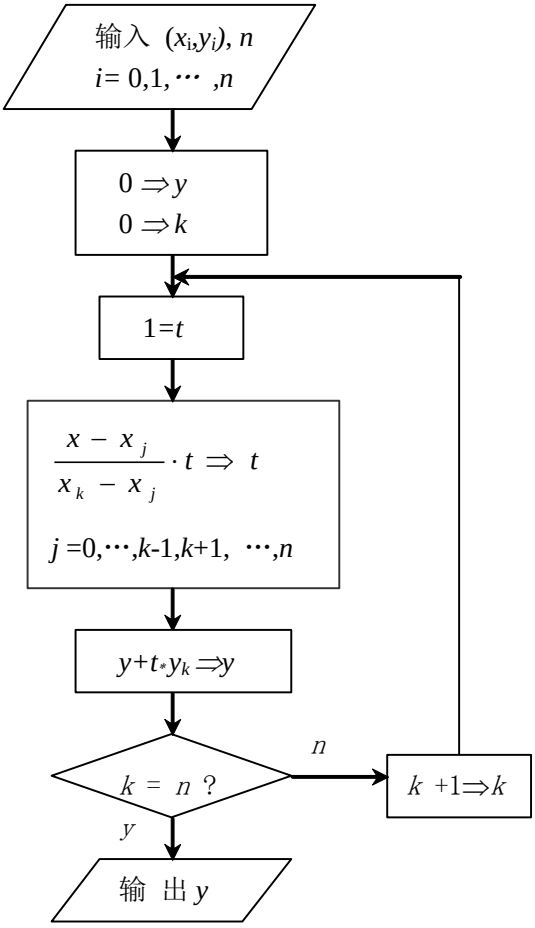
(2) 分段线性插值算法要注意以下关键点：

A. 定位：预测点x在第几个区间？

B. 假定x在第k个区间，

即 $x \in [x_k, x_{k+1}]$

一次插值公式为 $p_1(x) = y_k \frac{x - x_{k+1}}{x_k - x_{k+1}} + y_{k+1} \frac{x - x_k}{x_{k+1} - x_k}$



5 计算用例的参考输出

（1）实验1参考输出如下

X	y(精确)	y(拉格朗日)	y(分段线性)	误差(拉)	误差(分)
0.500000	0.800000	0.843408	0.750000	-0.043408	0.050000
4.500000	0.047059	1.578721	0.048643	-1.537662	-0.001584

6 思考题

(1) 由实验 1 体会插值中龙格现象产生的原因

(2) 牛顿插值和拉格朗日插值有什么区别和联系？

实验二 数值积分

1 实验目的

(1) 熟悉复化梯形方法、复化 Simpson 方法、梯形递推算法、龙贝格算法；
(2) 能编程实现复化梯形方法、复化 Simpson 方法、梯形递推算法、龙贝格算法；
(3) 理解并掌握自适应算法和收敛加速算法的基本思想；
(4) 分析实验结果体会各种方法的精确度，建立计算机求解定积分问题的感性认识

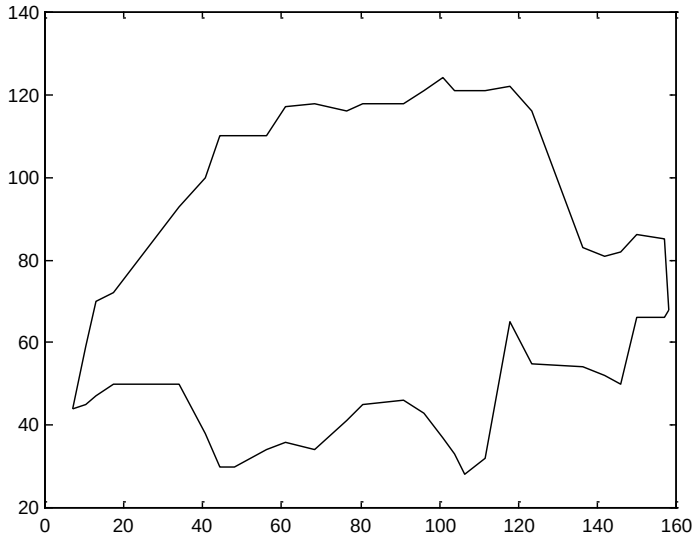
2 实验内容

(1) 用龙贝格算法计算 $\int_0^1 \frac{\sin x}{x} dx$

输入：积分区间，误差限

输出：序列 Tn，Sn,Cn,Rn 及积分结果（参考书本 P81 的表 2-4）

（2）设计方案计算国土面积



为了计算瑞士国土的面积,首先对地图作了如下测量:以西向东方向为 x 轴,由南向北方向为 y 轴,选择方便的原点,并将从最西边界到最东边界在 x 轴上的区间适当地划分为若干段,在每个分点的 y 方向测出南边界点和北边界点的 y 坐标，数据如表(单位 mm):

x	7.0	10.5	13.0	17.5	34.0	40.5	44.5	48.0	56.0
y1	44	45	47	50	50	38	30	30	34
y2	44	59	70	72	93	100	110	110	110
x	61.0	68.5	76.5	80.5	91.0	96.0	101.0	104.0	106.5
y1	36	34	41	45	46	43	37	33	28
y2	117	118	116	118	118	121	124	121	121
x	111.5	118.0	123.5	136.5	142.0	146.0	150.0	157.0	158.0
y1	32	65	55	54	52	50	66	66	68
y2	121	122	116	83	81	82	86	85	68

瑞士地图的外形如图(比例尺 18mm:40km)

问题：
试由测量数据计算瑞士国土的近似面积，并与其精确值 41288 平方公里比较
提示：
A. 将计算国土面积问题转化为求积分问题
当被积函数为表格形式（仅提供一组数据点，函数形式未知），使用梯形公式最为简单

3 算法基本原理

在许多实际问题中，常常需要计算定积分$\int_a^b f(x)dx$的值。根据微积分学基本定理，若被积函数 f(x)在区间[a,b]上连续，只要能找到 f(x)的一个原函数 F(x)，便可利用牛顿-莱布尼兹公式$\int_a^b f(x)=F(b)-F(a)$求得积分值。

但是在实际使用中，往往遇到如下困难，而不能使用牛顿-莱布尼兹公式。

（1） 找不到用初等函数表示的原函数

（2） 虽然找到了原函数，但因表达式过于复杂而不便计算

（3） f(x)是由测量或计算得到的表格函数

由于以上种种困难，有必要研究积分的数值计算问题。

利用插值多项式 $\boldsymbol{P_n(x)\approx f(x)}$ 则积分$\int_a^b f(x)dx$转化为$\int_a^b \boldsymbol{P_n(x)dx}$，显然易算。

$\int_a^b \boldsymbol{P_n(x)dx}$称为插值型求积公式。最简单的插值型求积公式是梯形公式和 Simpson 公式,。当求积结点提供较多，可以分段使用少结点的梯形公式和 Simpson 公式，并称为复化梯形公式、复化 Simpson 公式。如步长未知，可以通过误差限的控制用区间逐次分半的策略自动选取步长的方法称自适应算法。梯形递推公式给出了区间分半前后的递推关系。由梯形递推公式求得梯形序列，相邻序列值作线性组合得 Simpson 序列, Simpson 序列作线性组合得柯特斯序列, 柯特斯序列作线性组合的龙贝格序列。若$|R_2-R_1|<\varepsilon$,则输出 R2;否则…依此类推。如此加工数据的过程叫龙贝格算法，如下图所示：

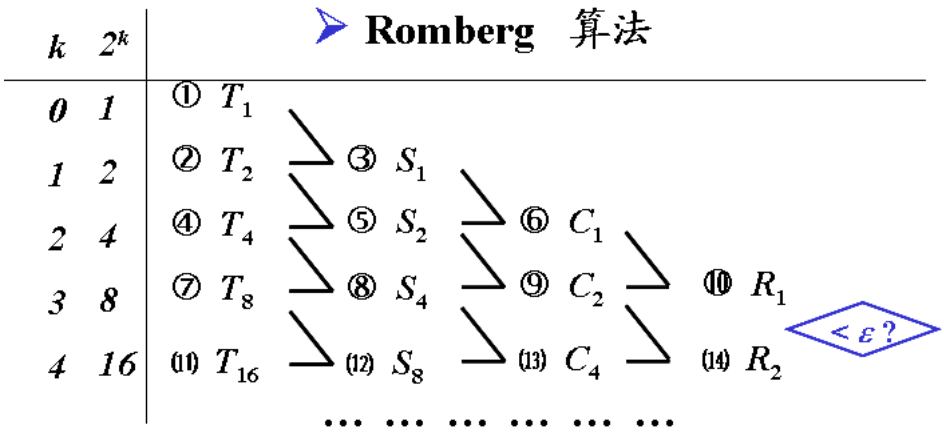


图 2.1 龙贝格算法数据加工流程

复化梯形公式 $T_n = \frac{1}{2}h[f(x_0)+2\sum_{k=1}^{n-1}f(x_k)+f(x_n)]$

复化 Simpson 公式 $S_n = \frac{h}{6}[f(x_0)+2\sum_{k=1}^{n-1}f(x_k)+f(x_n)+4\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})]$

梯形递推公式 $T_{2n} = \frac{T_n}{2}+\frac{h}{2}\sum_{k=0}^{n-1}f(x_{k+\frac{1}{2}})$

加权平均公式：$\frac{4T_{2n}-T_n}{4-1}=S_n$　$\frac{4^2S_{2n}-S_n}{4^2-1}=C_n$　$\frac{4^3C_{2n}-C_n}{4^3-1}=R_n$

龙贝格算法大大加快了误差收敛的速度，由梯形序列 O(h2) 提高到龙贝格序列的 O(h8)

4 算法描述

(1) 梯形递推算法见流程图

(2) 龙贝格算法见流程图

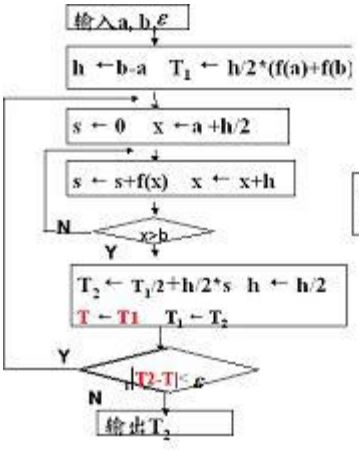


图 2.2 梯形递推算法流程图

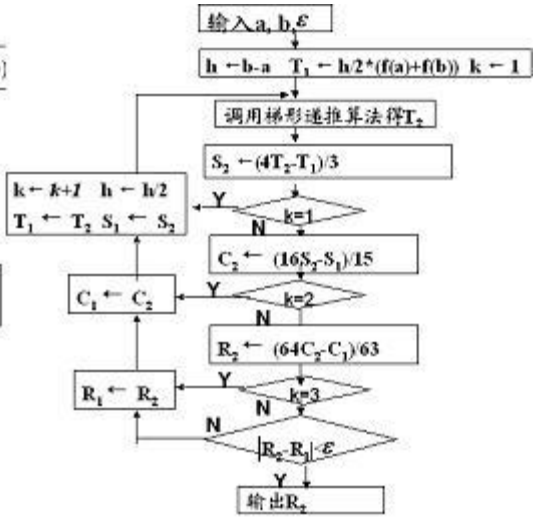


图 2.3 龙贝格算法流程图

5 计算用例的参考输出

实验参考输出如下

```
k       T        S        C        R
0 0.920735
1 0.939793 0.946146
2 0.944514 0.946087 0.946083
3 0.945691 0.946083 0.946083 0.946083
4 0.945985 0.946083 0.946083 0.946083
the 4 result is:0.946083
```

图 2.4 龙贝格参考输出

6 思考题

(1) 龙贝格算法除了本流程图的实现方法，用二维数组的存储方式如何实现？
(2) 根据定积分的计算方法，考虑二重积分的计算问题。

实验三 数值微分

1 实验目的

(1) 熟悉数值微分中 Euler 法，改进 Euler 法，Rung-Kutta 方法；
(2) 能编程实现 Euler 法，改进 Euler 法，Rung-Kutta 方法；
(3) 通过实验结果分析各个算法的优缺点;
(4) 明确步长对算法的影响并理解变步长的 Rung-Kutta 方法

2 实验内容

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

取 h=0.1 时用 Euler 法，改进 Euler 法，Rung-Kutta 方法求其数值解并与精确解进行比较。
输入：求解区间，初值，数值解个数
输出：数值解

3 算法基本原理

在许多科学技术问题中，建立的模型常常以常微分方程的形式表示。然而，除了少数特殊类型的常微分方程能用解析方法求其精确解外，要给出一般方程解析解的表达式是困难的。所以只能用近似方法求其数值解，在实际工作中常用计算机求常微分方程的数值解。所谓常微分方程的数值解即对于常微分方程初值问题 $\begin{cases} y' = f(x,y) \\ y(x_0) = y_0 \end{cases}$ 计算出在一系列节点 a = $x_0 < x_1 < \cdots < x_n = b$ 处的未知函数 y(x)近似值 $y_0, y_1, \cdots y_n$，即找到一系列离散点（x_0, y_0）（x_1, y_1）（x_2, y_2）…（x_n, y_n）近似满足常微分方程。数值解法的基本思想用差商代替导数，实现连续问题离散化，选取不同的差商代替导数可以得到不同公式。

改进欧拉公式是常用方法之一，包括预测和校正两步。先用欧拉公式进行预报，再将预报值代入梯形公式进行校正，从而达到二阶精度。通过龙格-库塔法我们可以获得更高精度。经典龙格-库塔法即在区间$[x_n, x_{n+1}]$取四点，并对这四点的斜率进行加权平均作为平均斜率，通过泰勒公式寻找使局部截断误差为 $O(h^5)$（即 4 阶精度）的参数满足条件。

改进的欧拉公式：　　预测 $y_{n+1} = y_n + \frac{h}{2}(f(x_n, y_n) + f(x_{n+1}, \bar{y}_{n+1}))$

校正 $\bar{y}_{n+1} = y_n + hf(x_n, y_n)$

四阶（经典）龙格-库塔公式

$$
\begin{aligned}
y_{n+1} &= y_n + \tfrac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\
K_1 &= f(x_n,\ y_n) \\
K_2 &= f(x_n + \tfrac{h}{2},\ y_n + \tfrac{h}{2}K_1) \\
K_3 &= f(x_n + \tfrac{h}{2},\ y_n + \tfrac{h}{2}K_2) \\
K_4 &= f(x_n + h,\ y_n + hK_3)
\end{aligned}
$$

4 算法描述

(1) 改进的欧拉算法见流程图
(2) 经典龙格-库塔算法见流程图

5 计算用例的参考输出

（1）实验1参考输出如下

x	y(4阶龙格)	y(改进)	y(欧拉)	y(精确)
0.100000	1.095446	1.095909	1.100000	1.095445
0.200000	1.183217	1.184097	1.191818	1.183216
0.300000	1.264912	1.266201	1.277438	1.264911
0.400000	1.341642	1.343360	1.358213	1.341641
0.500000	1.414215	1.416402	1.435133	1.414214
0.600000	1.483242	1.485955	1.508966	1.483240
0.700000	1.549196	1.552514	1.580338	1.549193
0.800000	1.612455	1.616474	1.649784	1.612452
0.900000	1.673324	1.678166	1.717780	1.673320
1.000000	1.732056	1.737867	1.784771	1.732051

图 3.1Euler 法，改进 Euler 法，R-K 方法参考输出

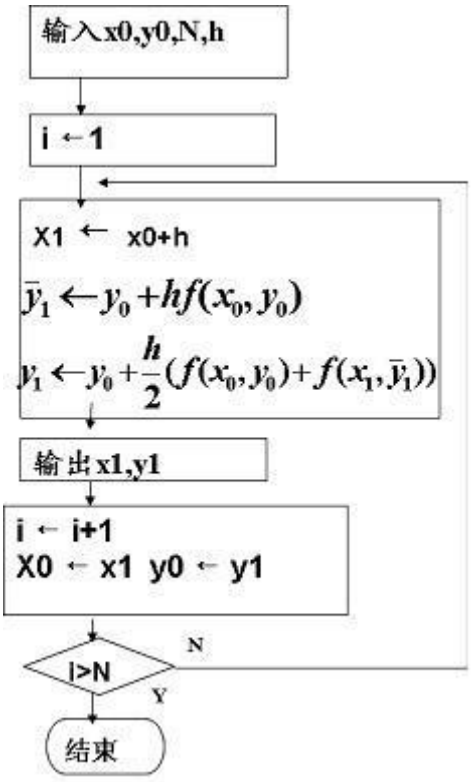


图 3.2 改进欧拉算法

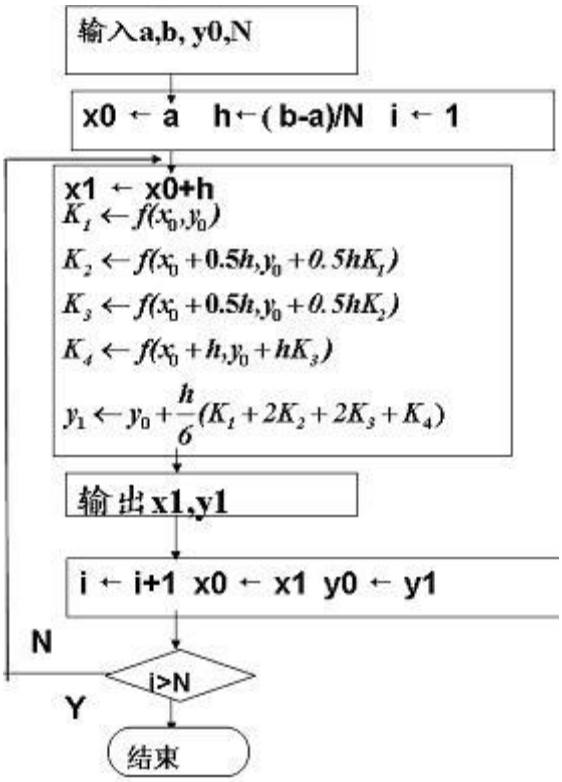


图 3.3 经典龙格库塔算法

6 思考题

(1) 步长 h 对上述算法有何影响？
(2) 变步长的经典龙格–库塔算法如何实现？

实验四 非线性方程求根迭代法

1 实验目的

(1) 熟悉非线性方程求根简单迭代法，牛顿迭代及牛顿下山法
(2) 能编程实现简单迭代法，牛顿迭代及牛顿下山法
(3) 认识选择迭代格式的重要性
(4) 对迭代速度建立感性的认识；分析实验结果体会初值对迭代的影响

2 实验内容

（1）用牛顿下山法解方程 $x^3-x-1=0$（初值为 0.6）

输入：初值，误差限，迭代最大次数，下山最大次数
输出：近似根各步下山因子

3 算法基本原理

求非线性方程组的解是科学计算常遇到的问题，有很多实际背景．各种算法层出不穷，其中迭代是主流算法。只有建立有效的迭代格式，迭代数列才可以收敛于所求的根。因此设计算法之前，对于一般迭代进行收敛性的判断是至关重要的。牛顿法也叫切线法，是迭代算法中典型方法，只要初值选取适当，在单根附近，牛顿法收敛速度很快，初值对于牛顿迭代至关重要。当初值选取不当可以采用牛顿下山算法进行纠正。

一般迭代：$x_{k+1}=\phi(x_k)$

$x=\phi(x)\Leftrightarrow f(x)=0$

牛顿公式：$x_{k+1}=x_k-\dfrac{f(x_k)}{f'(x_k)}$

牛顿下山公式：$x_{k+1}=x_k-\lambda\dfrac{f(x_k)}{f'(x_k)}$

下山因子 $\lambda=1,\dfrac{1}{2},\dfrac{1}{2^2},\dfrac{1}{2^3},\cdots$

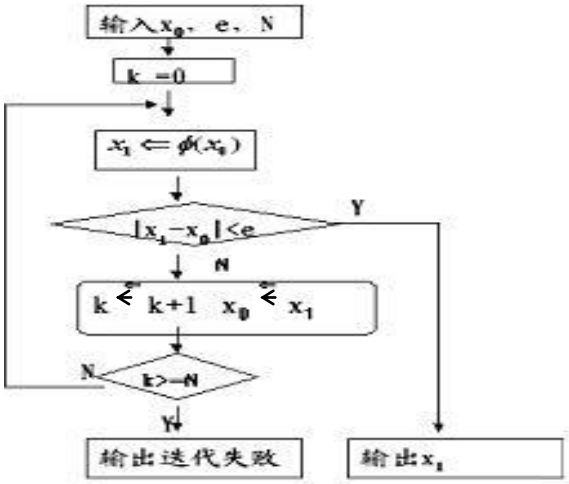


图 4.1 一般迭代算法流程图

下山条件 $|f(x_{k+1})|<|f(x_k)|$

4 算法描述

一般迭代算法见流程图
牛顿下山算法见流程图：

5 计算用例的参考输出

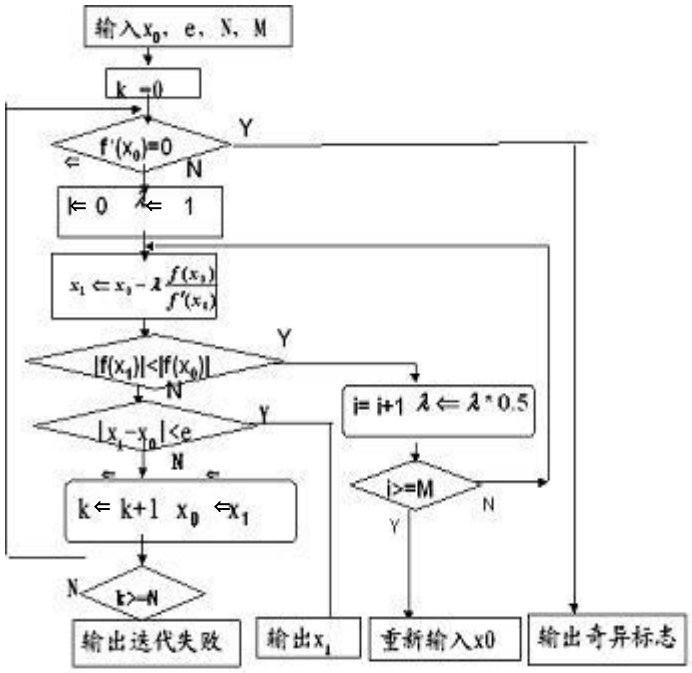


图 4.2 牛顿下山算法流程图

实验1输出参考书本p130页结果

6 思考题

（1)在实验 1 当最大下山比较大(如超过 20)在 x0=0.1 及其附近将会产生什么结果？原因？
（2）如何将牛顿公式中的导数近似替换成差商并编程实现？
（3）分析迭代收敛和发散的原因

实验五　求解线性方程组

1 实验目的

(1) 熟悉求解线性方程组的有关理论和方法；
(2) 能编程实现雅可比及高斯-塞德尔迭代法、列主元高斯消去法、约当消去，追赶法
(3) 通过测试，进一步了解各种方法的优缺点
(4) 根据不同类型的方程组，选择合适的数值方法

2　实验内容

（1）用 Gauss - Seidel 迭代法求解方程组 $\begin{cases} 10x_1 - x_2 - 2x_3 = 7.2 \\ -x_1 + 10x_2 - 2x_3 = 8.3 \\ -x_1 - x_2 + 5x_3 = 4.2 \end{cases}$

输入：系数矩阵 A，最大迭代次数 N，初始向量，误差限 e
输出：解向量

（2）用选主元高斯消去求行列式值 $\begin{vmatrix} 3 & -2 & 1 & 4 \\ -7 & 5 & -3 & -6 \\ 2 & 1 & -1 & 3 \\ 4 & -3 & 2 & 8 \end{vmatrix}$

提示：

A. $|A| = \begin{vmatrix} a_{11} & \cdots & a_{1n} \\ \cdots & \cdots & \cdots \\ a_{n1} & \cdots & a_{nn} \end{vmatrix} = \cdots = (-1)^m \begin{vmatrix} b_{11} & \cdots & b_{1n} \\ & \cdots & \cdots \\ & & b_{nn} \end{vmatrix} = (-1)^m b_{11} \cdots b_{nn}$

B. 消元结果直接存储在系数矩阵中

C. 当消元过程发生两行对调的情况为偶数次时，行列式值为对角线乘积，否则为对角线乘积的相反数。

3　算法基本原理

无论是三次样条还是拟合问题最终都归结为线性方程组，求解线性方程组在数值分析中非常重要，在工程计算中也不容忽视。

线性方程组大致分迭代法和直接法。只有收敛条件满足时，才可以进行迭代。雅可比及

高斯-塞德尔是最基本的两类迭代方法，最大区别是迭代过程中是否引用新值进行剩下的计算。消元是最简单的直接法，并且也十分有效的，列主元高斯消去法对求解一般的线性方程组都适用，同时可以用来求矩阵对应的行列式。约当消去实质是经过初等行变换将系数矩阵化为单位阵，主要用来求矩阵的逆。在使用直接法，要注意从空间、时间两方面对算法进行优化。

高斯-塞德尔迭代：$x_i^{(k+1)}=\frac{1}{a_{ii}}\left(b_i-\sum_{j=1}^{i-1}a_{ij}x_j^{(k+1)}-\sum_{j=i+1}^{n}a_{ij}x_j^{(k)}\right)\quad i=1,2,...,n$

列主元高斯消去法：列主元$|a_{lk}|=\max_{k\le i\le n}|a_{ik}|\neq 0;$

消元 $\begin{cases}a_{ij}^{(k+1)}=a_{ij}^{(k)}-m_{ik}a_{kj}^{(k)}\\ b_i^{(k+1)}=b_i^{(k)}-m_{ik}b_k^{(k)}\end{cases}$ $(i,j=k+1,...,n)$

回代 $x_i=\dfrac{b_i^{(i)}-\sum_{j=i+1}^{n}a_{ij}^{(i)}x_j}{a_{ii}^{(i)}}\qquad (i=n,...,1)$

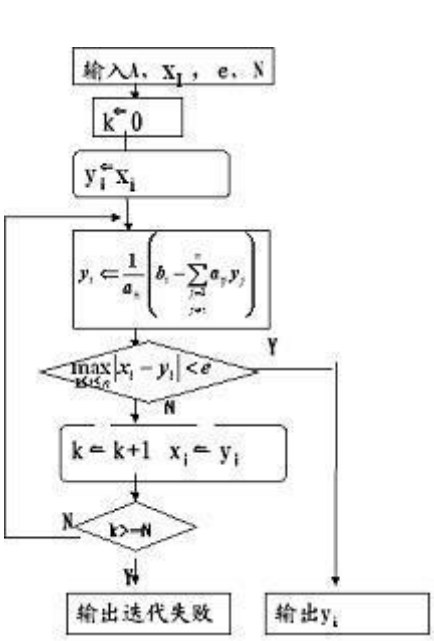


图 5.1G-S 迭代算法流程图

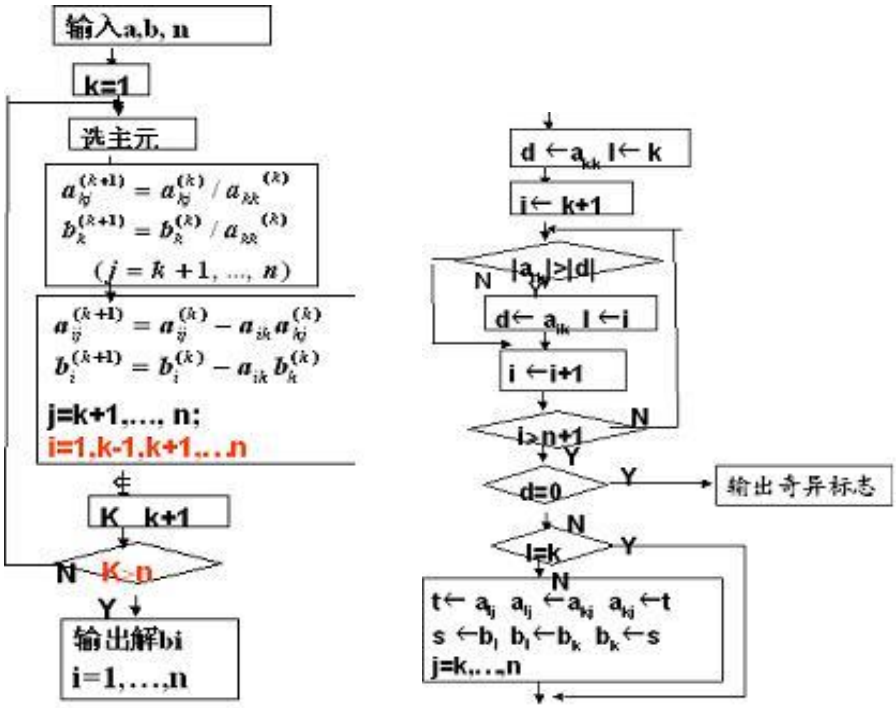


图 5.2 列主元的高斯消去流程

4 算法描述

Gauss - Seidel 算法见流程图
列主元的高斯消去算法见流程图

5 计算用例的参考输出

实验 1 输出参考书本 p146 页 表 5-2
实验 2 参考输出：18

6 思考题

(1) 对于具体的线性方程组该选用何种方法达到节省时间和空间的目的？

实验报告内容要求

一、实验名称

二、实验目的

三、实验内容

四、基本原理（计算公式）

五、算法设计与实现（流程图，关键点）

六、输入与输出

七、结果讨论和分析