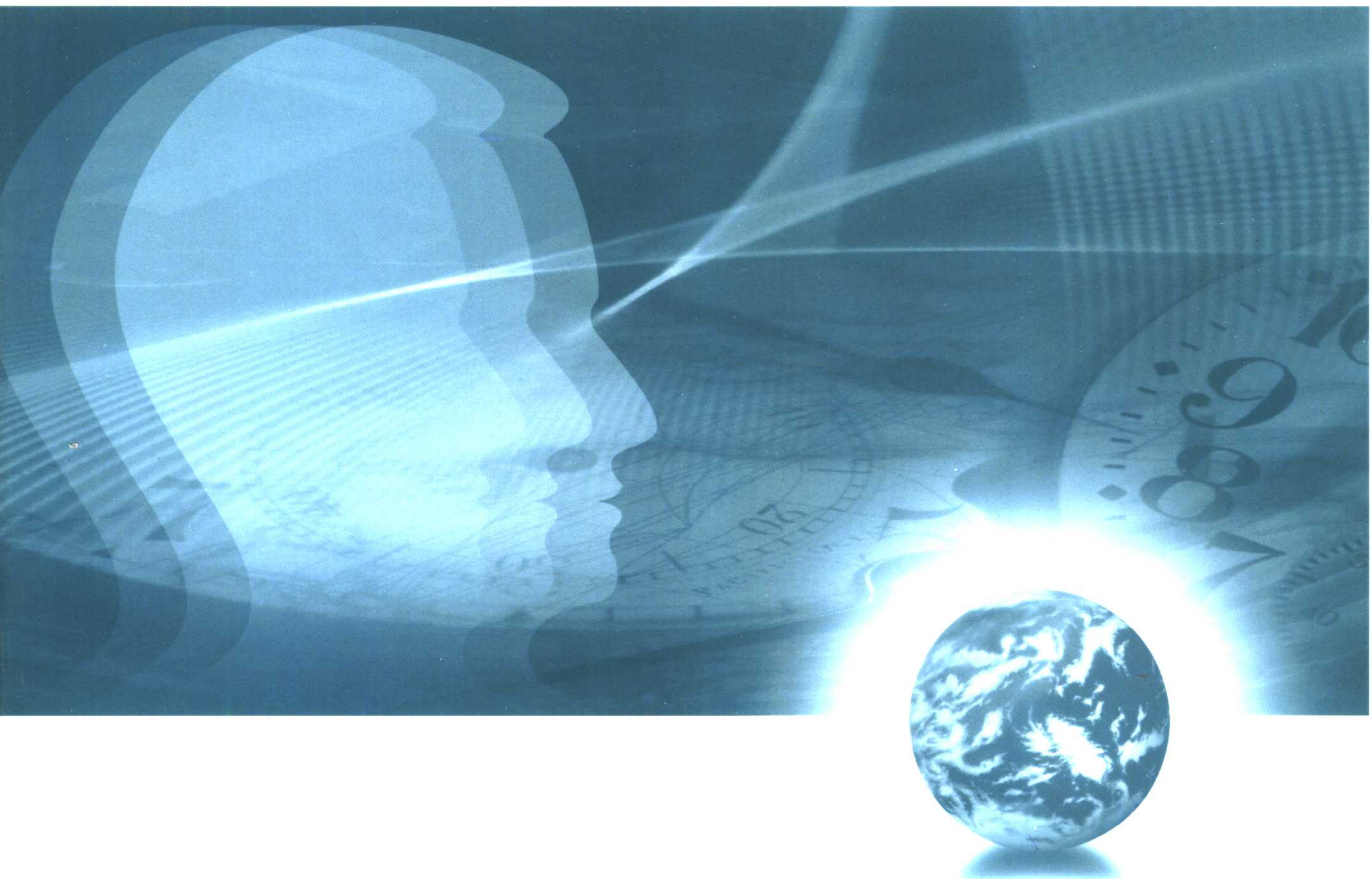
高等院校计算机科学与技术“十五”规划教材

数值分析算法描述与习题解答



徐士良 编著

机械工业出版社
CHINA MACHINE PRESS

高等院校计算机科学与技术“十五”规划教材

数值分析

算法描述与习题解答

徐士良　编著

机械工业出版社

本书是《数值分析与算法》教材的配套用书，书中给出了教材每一章主要算法的C语言描述，以及教材中习题的详细解答。

本书也可作为其他有关数值分析课程的参考书。

图书在版编目（CIP）数据

数值分析算法描述与习题解答/徐士良编著. —北京：机械工业出版社，2003.3

高等院校计算机科学与技术“十五”规划教材

ISBN 7-111-11781-6

Ⅰ. 数… Ⅱ. 徐… Ⅲ. 电子计算机－计算方法－高等学校－自学参考资料 Ⅳ. TP301.6

中国版本图书馆CIP数据核字（2003）第017983号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）

策　　划：胡毓坚

责任编辑：时　静

责任印制：付方敏

三河市宏达印刷有限公司印刷·新华书店北京发行所发行

2003年4月第1版·第1次印刷

787mm×1092mm 1/16 · 11.5印张 · 278千字

0001－5000册

定价：17.00元

出版说明

信息技术高度普及的今天，具备一定层次的信息技术素养成为社会素质教育的一个重要目标，由此对高等院校的计算机专业教育提出了更高更新的要求。教育水平提高的关键是教学质量，那么对教学质量有直接影响的教材建设就成为了计算机专业教育的根本，为重中之重。

适逢高等院校计算机专业教育改革的关键时期，为配合相关的教材建设，机械工业出版社会同全国在该领域内享誉盛名、具备雄厚师资和技术力量的高等院校，包括清华大学、上海交通大学、南京大学、成都电子科技大学、东南大学、西安电子科技大学、解放军理工大学、北京科技大学等重点名校，组织了多位长期从事教学工作的骨干教师，集思广益，对当前高等院校的教学现状开展了广泛而深入的研讨，继而紧密结合当前技术发展需要并针对教学改革所提出的问题，精心编写了这套面向普通高等院校计算机专业的系列教材，并陆续出版。

本套教材内容覆盖了普通高等院校计算机专业学生的必修课程，另外还恰如其分地添加了一些选修课程，总体上分为基础、软件、硬件、网络和多媒体五大类。在编写过程中，对教学改革力度比较大、内容新颖以及各院校急需的并且适应社会经济发展的新教材，优先选择出版。

本套教材注重系统性、普及性和实用性，力求达到专业基础课教材概念清晰、深度合理的标准，并且注意与专业课教学的衔接；专业课教材覆盖面广、深浅适中，在体现相关领域最新发展的同时注重理论联系实际。全套教材体现了教育改革的最新思想，可作为高等院校计算机科学与技术专业的教学用书，同时也是培训班和自学使用的最佳教材。

机械工业出版社

前　言

为便于读者对《数值分析与算法》这本教材中各章主要算法的理解与应用，特编写了本书。

本书的每一章与教材《数值分析与算法》的各章对应。且每章均分为两部分：第一部分给出了教材中所介绍的主要算法的 C 语言描述，同时给出一个例题程序，便于读者理解和掌握如何调用该函数程序；第二部分给出了教材中习题的详细解答。

本书中所有的算法函数程序以及相应的例题程序都经过实际调试运行，给出的运行结果基本上都是计算机实际输出的结果。

本书所有 C 语言程序均可在机械工业出版社网站（www.cmpbook.com）上下载。

本书不仅可以作为《数值分析与算法》一书的配套教材，也可以作为高等理工科院校非数学专业的《数值分析》或《计算方法》等课程的参考教材。在本书编写过程中，白小玲、葛兵、徐娟、徐艳、马尔妮等同志做了大量的工作，在此表示感谢。

限于水平，书中难免会有错误和不当之处，恳请读者批评指正。

徐士良

目　录

第1章 绪　　论

1.1 主要算法描述

算法1-1 简单二分法求方程实根

参数说明：

a 双精度实型变量。求根区间的左端点。

b 双精度实型变量。求根区间的右端点。

eps 双精度实型变量。求根的精度要求。

f 双精度实型指针变量。指向计算方程左端函数 $f(x)$ 值的函数。

本函数返回一个双精度实型值，即方程根的近似值。

```
#include "stdio.h"
#include "math.h"
double  root(a, b, eps, f)
double a, b, eps, (*f)();
{ double f0, f1, c;
  f0=(*f)(a);
  while (fabs(a-b)>=eps)      /*不满足精度要求继续迭代*/
    { c=(a+b)/2;  f1=(*f)(c);       /*取区间中点*/
      if (f1==0) return(c);         /*区间中点已满足方程，返回*/
      if (f0*f1>0)  a=c;      /*用区间中点代替左端点，即取区间的后半部分*/
      else b=c;               /*用区间中点代替右端点，即取区间的前半部分*/
    }
  c=(a+b)/2;      /*取区间中点作为方程实根的近似值*/
  return(c);
}
```

【例 1-1】 用简单二分法求方程

$$f(x)=x+\ln x-2.2=0$$

在区间[1，2]上的一个实根。精度要求为 $\varepsilon=10^{-6}$ 。

主函数与计算 f(x)值的函数如下：

```
/*root0.c*/
#include "root.c"
main()
{ double a, b, eps, f();
  a=1;  b=2;  eps=0.000001;
  printf("x=%7.3f\n", root(a, b, eps, f));
```

```
}
 double  f(x)
 double x;
 { double y;
  y=x+log(x)-2.2;
  return(y);
}
```

运行结果为

x= 1.681

算法 1-2　回溯法求解皇后问题

参数说明：

n 整型变量。皇后问题的阶数。

```
#include "stdio.h"
#include "math.h"
#include "stdlib.h"
void queen(int n)
{ int  i, j, k, jt, *q;
 q=malloc(n*sizeof(int));          /*申请长度为 n 的一维数组 q*/
 for (i=0;  i<n;  i++)  q[i]=0;    /*开始时，所有行的皇后均在第 1 列*/
 i=0;      /*从第 1 个皇后开始安排*/
 jt=1;
 printf("\n");
 printf("%d queen problem\n", n);
 while(jt==1)
   { if (q[i]<n)         /*当前皇后在棋盘内*/
      { k=0;
        while ((k<i)&&((q[k]-q[i])*
              (fabs(q[k]-q[i])-fabs(k-i)))!=0)  k=k+1;
            /*判断当前皇后与前面的皇后是否有攻击*/
       if (k<i)  /*当前皇后与前面的皇后有攻击*/
            q[i]=q[i]+1;    /*当前皇后右移一个位置*/
       else        /*当前皇后与前面的皇后没有攻击*/
        { if (i==n-1)    /*当前安排好的是最后一个皇后*/
            { for (j=0;  j<n;  j++)        /*输出互不攻击的布局*/
                  printf("%5d", q[j]+1);
              printf("\n");
              q[n-1]=q[n-1]+1;
              /*最后一个皇后右移一个位置，以便寻找下一个互不攻击的布局*/
              }
          else  i=i+1;          /*考虑安排下一个皇后*/
          }
      }
   else        /*当前皇后无法安排*/
     { q[i]=0;      /*将当前皇后重新安置在第 1 列*/
```

```
        i=i-1;     /*考虑重新安排上一个皇后*/
        if (i<0)    /*无路可退，释放内存空间，结束*/
          { printf("\n");   free(q);   return;   }
        q[i]=q[i]+1;    /*当前皇后右移一个位置*/
      }
    }
  }
```

【例 1-2】 用回溯法求解 n 阶皇后问题。其中 n 由键盘输入。

主函数如下：

```
/*queen0.c*/
#include "queen.c"
main()
{ int n;
  printf("input n=");
  scanf("%d", &n);
  queen(n);
}
```

当 $n=4$ 时，运行结果为

```
input n=4<回车>(由键盘输入)
4 queen problem
    2     4     1     3
    3     1     4     2
```

1.2 习题 1 解答

1．设 $\sqrt{20}=4.472136$ 具有 7 位有效数字，试确定下列各近似数的有效数字位数：

(1) $\sqrt{20}\approx4.426$　　(2) $\sqrt{20}\approx4.477$　　(3) $\sqrt{20}\approx4.47$

(4) $\sqrt{20}\approx4.47464$　　(5) $\sqrt{20}\approx4.467576$

解：设 $x=\sqrt{20}=4.472136$。

(1) $x^*=4.426$。$|x-x^*|=|4.472136-4.426|=0.046136<0.05$。因此，$x^*=4.426$ 准确到小数点后第 1 位，即 $x^*=4.426$ 具有 2 位有效数字。

(2) $x^*=4.477$。$|x-x^*|=|4.472136-4.477|=0.004864<0.005$。因此，$x^*=4.477$ 准确到小数点后第 2 位，即 $x^*=4.477$ 具有 3 位有效数字。

(3) $x^*=4.47$。$|x-x^*|=|4.472136-4.47|=0.002136<0.005$。因此，$x^*=4.47$ 准确到小数点后第 2 位，即 $x^*=4.47$ 具有 3 位有效数字。

(4) $x^*=4.47464$。$|x-x^*|=|4.472136-4.47464|=0.002504<0.005$。因此，$x^*=4.47464$ 准确到小数点后第 2 位，即 $x^*=4.47464$ 具有 3 位有效数字。

(5) $x^*=4.467576$。$|x-x^*|=|4.472136-4.467576|=0.00456<0.005$。因此，$x^*=4.467576$ 准确到小数点后第 2 位，即 $x^*=4.467576$ 具有 3 位有效数字。

2．在下列各对数中，设 x_T 为准确值，确定 x_A 的有效数字位数：

(1) $x_T=451.01$，$x_A=451.023$

(2) $x_T=-0.04518$，$x_A=-0.045113$

(3) $x_T=23.4604$，$x_A=23.4213$

解：(1) $|x_T-x_A|=|451.01-451.023|=0.013<0.05$。因此，$x_A=451.023$ 关于 $x_T=451.01$ 准确到小数点后第 1 位，即 $x_A=451.023$ 关于 $x_T=451.01$ 具有 4 位有效数字。

(2) $|x_T-x_A|=|-0.04518+0.045113|=0.000067<0.0005$。因此，$x_A=-0.045113$ 关于 $x_T=-0.04518$ 准确到小数点后第 3 位，即 $x_A=-0.045113$ 关于 $x_T=-0.04518$ 具有 2 位有效数字。

(3) $|x_T-x_A|=|23.4604-23.4213|=0.0391<0.05$。因此，$x_A=23.4213$ 关于 $x_T=23.4604$ 准确到小数点后第 1 位，即 $x_A=23.4213$ 关于 $x_T=23.4604$ 具有 3 位有效数字。

3．下列各数均准确到末位数字，请指出它们的有效数字位数：

(1) 0.2300×10^2　(2) 45.678　(3) 0.00465×10^{-3}　(4) 0.012375

解：(1) 4 位　(2) 5 位　(3) 3 位　(4) 5 位

4．下列各算式中的数均准确到末位数字，请指出其运算结果的准确值所在的范围：

(1) $1.23+4.6$　(2) $4.6-1.23$　(3) 3×4

解：(1) $[1.225+4.55,\ 1.235+4.65]=[5.775,\ 5.885]$

(2) $[4.55-1.235,\ 4.65-1.225]=[3.315,\ 3.425]$

(3) $[2.5\times3.5,\ 3.5\times4.5]=[8.76,\ 15.75]$

5．设$\sqrt{399}=19.975$ 具有五位有效数字，求方程 $x^2-40x+1=0$ 的两个实根。

解：根据一元二次方程求根公式有

$$x_{1,2}=\frac{40\pm\sqrt{40^2-4}}{2}=20\pm\sqrt{399}$$

因此

$$x_1=20+\sqrt{399}=39.975$$

$$x_2=\frac{1}{x_1}=\frac{1}{39.975}=0.025015$$

6．求方程 $x^2-74x+2=0$ 的两个根，使它们至少具有四位有效数字。其中 $\sqrt{1367}=36.974$。

解：根据一元二次方程求根公式有

$$x_{1,2}=\frac{74\pm\sqrt{74^2-4\times2}}{2}=37\pm\sqrt{1367}$$

因此

$$x_1=37+\sqrt{1367}=73.974$$

$$x_2=\frac{2}{x_1}=\frac{2}{73.974}=0.027037$$

7．设 $x>0$，x 的相对误差为 δ，估计 $f(x)=\ln x$ 的绝对误差。

解：因为 $|E_r(x)|=|\dfrac{E(x)}{x}|=\delta$，所以 $f(x)$的绝对误差为

$$|E[\ln x]|=|(\ln x)'_x E(x)|=|\frac{1}{x}E(x)|=|E_r(x)|=\delta$$

8．设 x 的绝对误差为 η，估计 $f(x)=\sin x$ 在 $x=0$ 处的绝对误差。

解：因为 $|E(x)|=\eta$，所以 $f(x)=\sin x$ 的绝对误差为

$$|E[\sin x]|=|(\sin x)'_x E(x)|=|\cos x E(x)|=\eta|\cos x|$$

因此，$f(x)=\sin x$ 在 $x=0$ 处的绝对误差为

$$|E[\sin x]|_{x=0}=\eta$$

9．设 x 的绝对误差为 η，求 $y=e^{0.2x}$ 在 $x=1$ 处的相对误差。

解：因为 $|E(x)|=\eta$，所以 $f(x)=e^{0.2x}$ 的相对误差为

$$|E_r[e^{0.2x}]|=|\frac{E[e^{0.2x}]}{e^{0.2x}}|=0.2|E(x)|=0.2\eta$$

因此，$f(x)=e^{0.2x}$ 在 $x=1$ 处的相对误差为 0.2η。

10．计算圆面积要使相对误差限为1%，问度量半径 R 时允许的相对误差限是多少？

解：圆面积为 $S=\pi R^2$，因此，圆面积的相对误差为

$$|E_r(S)|=|\frac{E(S)}{S}|=\frac{2\pi R|E(R)|}{\pi R^2}=2|\frac{E(R)}{R}|=2|E_r(R)|$$

根据已知条件 $|E_r(S)|=1\%$，因此有 $2|E_r(R)|=1\%$，解出 $|E_r(R)|=0.5\%$。即度量半径 R 时允许的相对误差限为0.5%。

11．正方形的边长约为 100cm，为使其面积的误差不超过 1cm^2，则在测量边长时允许的最大误差为多少？

解：正方形面积为 $S=a^2$，$a=100\text{cm}$，$E(S)=1\text{cm}^2$。因为

$$|E(S)|=2a|E(a)|$$

所以有

$$|E(a)|=\frac{1}{2a}|E(S)|=\frac{1}{200}\text{cm}=0.005\text{cm}$$

即在测量边长时允许的最大误差为0.005cm。

12．设 $\sqrt{20}$ 的一个近似数的相对误差为 0.1%，则该近似数具有多少位有效数字？

解：设 $x=\sqrt{20}$，根据定理1-2有

$$\frac{1}{2(x_1+1)}10^{-(n-1)}=1\%$$

其中 $x_1=4$。由此解出 $n=3$。因此，该近似数至少具有3位有效数字。

13．设 $\sqrt{34}$ 的一个近似数具有四位有效数字，问此近似数的相对误差为多少？

解：设 $x=\sqrt{34}$。根据定理1-1，该近似数的相对误差为

$$|E_r(x)|\leqslant\frac{1}{2x_1}10^{-(n-1)}$$

其中 $x_1=5$，$n=4$。代入上式有

$$|E_r(x)|\leqslant\frac{1}{10}10^{-(4-1)}=0.01\%$$

14．设$\sqrt{105}$的一个近似数具有四位有效数字，试估计该近似数的相对误差。

解：设 $x=\sqrt{105}$。根据定理 1-1，该近似数的相对误差为

$$|E_r(x)|\leqslant\frac{1}{2x_1}10^{-(n-1)}$$

其中 $x_1=1$，$n=4$。代入上式有

$$|E_r(x)|\leqslant\frac{1}{2}10^{-(4-1)}=0.005\%$$

15．下列五个算式互相等价，如果取$\sqrt{2}=1.41$。分别计算按各算式计算时的相对误差。

(1) $(\sqrt{2}-1)^6$　　(2) $(3-2\sqrt{2})^3$　　(3) $99-70\sqrt{2}$

(4) $\dfrac{1}{(\sqrt{2}+1)^6}$　　(5) $\dfrac{1}{(3+2\sqrt{2})^3}$

解：设 $x=\sqrt{2}$，$x^*=1.41$，则$|E(x)|=0.005$。

(1) 设 $f_1(x)=(x-1)^6$，则有

$$|E_r[f_1(x)]|=\left|\frac{E[f_1(x)]}{f_1(x)}\right|=\left|\frac{6E(x)}{x-1}\right|=\frac{0.03}{0.41}=7.3\%$$

(2) 设 $f_2(x)=(3-2x)^3$，则有

$$|E_r[f_2(x)]|=\left|\frac{\mathrm{E}[f_2(x)]}{f_2(x)}\right|=\left|\frac{-6\mathrm{E}(x)}{3-2x}\right|=\frac{0.03}{0.18}=16.7\%$$

(3) 设 $f_3(x)=99-70x$，则有

$$|E_r[f_3(x)]|=\left|\frac{E[f_3(x)]}{f_3(x)}\right|=\left|\frac{-70E(x)}{99-70x}\right|=\frac{0.35}{0.3}=116.7\%$$

(4) 设 $f_4(x)=\dfrac{1}{(x+1)^6}$，则有

$$|E_r[f_4(x)]|=\left|\frac{E[f_4(x)]}{f_4(x)}\right|=\left|\frac{-6E(x)}{x+1}\right|=\frac{0.03}{2.41}=1.245\%$$

(5) 设 $f_5(x)=\dfrac{1}{(3+2x)^3}$，则有

$$|E_r[f_5(x)]|=\left|\frac{E[f_5(x)]}{f_5(x)}\right|=\left|\frac{-6E(x)}{3+2x}\right|=\frac{0.03}{5.82}=0.515\%$$

16．设有函数 $f(x)=\ln(x+\sqrt{x^2+1})$。当$|x|$的值较大时，怎样计算 $f(x)$的值才能避免有效数字的丢失？

解：当 $x<0$ 时，$-x\approx\sqrt{x^2+1}$，因此，运算 $x+\sqrt{x^2+1}$ 会严重丢失有效数字。在这种情况下，应作如下变换：

$$f(x)=\ln(x+\sqrt{x^2+1})=-\ln(\sqrt{x^2+1}-x)$$

即当 $x>0$ 时，按

$$f(x)\mp\ln(x+\sqrt{x^2+1})$$

计算；当 x<0 时，按

$$f(x)=-\ln(\sqrt{x^2+1}-x)$$

计算。

17．设 $y_0=28$ 为准确值，一个递推公式为

$$y_n=y_{n-1}-\frac{1}{100}\sqrt{783}\text{，}n=1\text{，}2\text{，}\cdots\text{，}100$$

若取 $\sqrt{783}=27.982$(具有五位有效数字)，问递推计算到 y_{100} 时的误差为多少？

解：$\sqrt{783}$ 的误差为 0.0005。$\frac{1}{100}\sqrt{783}$ 的误差为 0.000005。

需递推 100 次，因此，$E[y_{100}]=100\times0.000005=0.0005$。

18．设有如下三项递推关系：

$$\begin{cases}P_0=1,\ P_1=\dfrac{1}{3}\\ P_{n+1}=\dfrac{7}{3}P_n-\dfrac{2}{3}P_{n-1},\ n=1,2\cdots\end{cases}$$

若取 $P_1=0.333333$，试估计用该递推公式计算得到的 P_{10} 的绝对误差。

解：由三项递推关系式解出：$t_1=\frac{1}{3}$，$t_2=2$。得到误差序列为

$$\Delta P_n=\varepsilon_A t_1^n+\varepsilon_B t_2^n\approx\varepsilon_B t_2^n=2^n\varepsilon_B$$

即

$$|\Delta P_{10}|\approx2^{10}\varepsilon_B$$

其中 $B=\frac{P_1-P_0t_1}{t_2-t_1}$，而 $|\Delta P_1|=0.5\times10^{-6}$。因此，

$$\varepsilon_B\approx\frac{|\Delta P_1|}{t_2-t_1}=0.3\times10^{-6}$$

第 2 章　矩阵与线性代数方程组

2.1　主要算法描述

算法 2-1　全选主元高斯消去法求解实系数线性代数方程组

参数说明：

n 整型变量。方程组的阶数。

a 双精度实型二维数组，体积为 $n\times n$。存放方程组的系数矩阵，返回时将被破坏。

b 双精度实型一维数组，长度为 n。存放方程组右端的常数向量；返回方程组的解向量。

本函数返回整型标志值。若返回的标志值为 0，则表示原方程组的系数矩阵奇异，输出信息“fail”；若返回的标志值不为 0，则表示正常返回。

```
#include "stdlib.h"
#include "math.h"
#include "stdio.h"
int rgauss(n, a, b)
int n;
double a[], b[];
{ int *js, l, k, i, j, is, p, q;
  double d, t;
  js=malloc(n*sizeof(int));          /*开辟用于记忆列交换信息的动态空间*/
  l=1;                               /*置非奇异标志*/
  for (k=0; k<=n-2; k++)
    { d=0.0;
     for (i=k; i<=n-1; i++)        /*全选主元*/
       for (j=k; j<=n-1; j++)
         { t=fabs(a[i*n+j]);
          if (t>d) { d=t;  js[k]=j;  is=i; }  /*记忆行、列交换信息*/
         }
     if (d+1.0==1.0) l=0;    /*置奇异标志*/
     else
      { if (js[k]!=k)
          for (i=0; i<=n-1; i++)      /*列交换*/
           { p=i*n+k;  q=i*n+js[k];
             t=a[p];  a[p]=a[q];  a[q]=t;
           }
        if (is!=k)
          { for (j=k; j<=n-1; j++)     /*行交换*/
              { p=k*n+j;  q=is*n+j;
```

```
                t=a[p]; a[p]=a[q]; a[q]=t;
              }
            t=b[k]; b[k]=b[is]; b[is]=t;
          }
      }
      if (l==0)      /*奇异返回*/
        { free(js); printf("fail\n");
          return(0);
        }
      d=a[k*n+k];
      for (j=k+1; j<=n-1; j++)  /*归一化*/
        { p=k*n+j; a[p]=a[p]/d; }
      b[k]=b[k]/d;
      for (i=k+1; i<=n-1; i++)     /*消元*/
        { for (j=k+1; j<=n-1; j++)
            { p=i*n+j;
              a[p]=a[p]-a[i*n+k]*a[k*n+j];
            }
          b[i]=b[i]-a[i*n+k]*b[k];
        }
    }
  d=a[(n-1)*n+n-1];
  if (fabs(d)+1.0==1.0)    /*奇异返回*/
    { free(js); printf("fail\n");
      return(0);
    }
  b[n-1]=b[n-1]/d;         /*计算解向量的最后一个分量*/
  for (i=n-2; i>=0; i--)  /*回代*/
    { t=0.0;
      for (j=i+1; j<=n-1; j++)
        t=t+a[i*n+j]*b[j];
      b[i]=b[i]-t;
    }
  js[n-1]=n-1;
  for (k=n-1; k>=0; k--)      /*恢复解向量*/
    if (js[k]!=k)
      { t=b[k]; b[k]=b[js[k]]; b[js[k]]=t; }  /*解向量行交换*/
  free(js);      /*释放动态空间*/
  return(1);     /*返回正常标志*/
}
```

【例 2-1】 用高斯消去法求解下列方程组：

$$\begin{cases} 0.2388x_0 + 0.2471x_1 + 0.2568x_2 + 1.2671x_3 = 1.8471 \\ 0.1968x_0 + 0.2071x_1 + 1.2168x_2 + 0.2271x_3 = 1.7471 \\ 0.1581x_0 + 1.1675x_1 + 0.1768x_2 + 0.1871x_3 = 1.6471 \\ 1.1161x_0 + 0.1254x_1 + 0.1397x_2 + 0.1490x_3 = 1.5471 \end{cases}$$