

C 常用算法程序集

徐士良 编

清华大学出版社

(京)新登字 158 号

内 容 简 介

本书是针对工程上常用的行之有效的算法而编写的 C 语言函数程序集。书中包括了近年来出现的许多新算法。全书分为数值计算与非数值计算两部分。其中数值计算部分的内容包括: 插值、数值积分、线性代数方程组的求解、矩阵运算、矩阵特征值及特征向量的计算、非线性方程(组)的求解、常微分方程(组)的求解、极值问题、拟合与逼近、数据处理与回归分析、数学变换与滤波、特殊函数; 非数值计算部分的内容包括: 排序、查找、屏幕操作、图形操作等。

书中所有算法函数程序均在 IBM-PC 系列及其兼容机上调试通过, 并有配套软盘。

本书可供广大科研人员、工程技术人员及管理工作者阅读使用, 也可作为高等院校的《数值分析》或《程序设计》等类似课程的参考书。

版权所有, 翻印必究。

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

C 常用算法程序集

徐士良 编

责任编辑 范素珍

清华大学出版社出版

北京 清华园

国防工业出版社印刷厂印刷

新华书店总店科技发行所发行

开本: 787× 1092 1/16 印张: 43.75 字数: 1038 千字

1994 年 1 月第 1 版 1995 年 12 月第 4 次印刷

印数: 20001—23000

ISBN 7-302-01343-8/TP · 519

定价: 33.60 元

前 言

目前, C 语言已不仅被计算机软件工作者用来设计或开发系统软件, 而且也为广大计算机应用工作者(计算工作者、 科学研究人员及工程技术人员) 所喜欢使用。用 C 语言提供工程中实用的算法函数程序还很少, 特别是系统地提供数值计算中的 C 语言常用算法函数程序几乎还没有。与 FORTRAN 语言相比, 这是一个很大的差距, 对于广大计算机应用工作者来说是很不方便的。故编写了此书。

由于 C 语言所涉及到的概念很多, 使用又很灵活, 而本书的重点是提供实用的 C 算法函数程序。因此, 在本书的算法函数程序中, 尽量采用最基本的 C 语句与规则, 并且特别注意到与其它常用计算机语言(如 BASIC、FORTRAN) 所相通的编程习惯和风格。这样做的理由有两个方面: 一是考虑到本书的读者主要是计算机应用人员, 而不是软件设计与开发人员, 读者只要对 C 语言稍有入门即可使用本书解决实际问题; 二是考虑到程序的效率主要靠算法来保证, 而作为程序集主要应强调易读、通用, 尽量避免由于程序中语句的细节而影响本书应起的作用或造成读者对程序的困惑不解。

本书的宗旨是实用。书中的所有函数程序都经过实际计算的检验。对于 IBM-PC 系列及其兼容机, 书中的所有算法函数程序均有配套的软盘。各程序在软盘中的文件名格式为 \CHX\ 函数名. C, 其中 X 为该函数程序在本书中的章号。

本书所涉及的数值计算问题的算法可以算得上“全”。因为在数值计算这部分中, 包括了工程上常用的行之有效的算法, 有些是近年来出现的新算法, 它们基本可以满足各学科领域中经常遇到的实际计算问题的需要。而对于非数值计算部分的算法, 本书并不求“全”, 只是提供了一些广大读者共同关心的最基本的算法。其中第十五、十六章的函数程序与硬件有关, 只适用于 IBM-PC 系列及其兼容机。

限于水平, 书中难免有错误之处, 恳请读者批评指正。

作 者

1993 年 1 月

目 录

第一篇 数值计算

第一章 插值	1
1.1 一元全区间不等距插值	1
1.2 一元全区间等距插值	3
1.3 一元三点不等距插值	5
1.4 一元三点等距插值	7
1.5 有理不等距插值	9
1.6 有理等距插值	12
1.7 埃尔米特不等距插值	14
1.8 埃尔米特等距插值	16
1.9 埃特金不等距逐步插值	17
1.10 埃特金等距逐步插值	20
1.11 光滑不等距插值	22
1.12 光滑等距插值	27
1.13 第一种边界条件的三次样条函数插值、微商与积分	31
1.14 第二种边界条件的三次样条函数插值、微商与积分	36
1.15 第三种边界条件的三次样条函数插值、微商与积分	41
1.16 二元三点插值	46
1.17 二元全区间插值	50
第二章 数值积分	53
2.1 变步长梯形求积法	53
2.2 自适应梯形求积法	55
2.3 变步长辛卜生求积法	57
2.4 龙贝格求积法	59
2.5 高斯求积法	62
2.6 切比雪夫求积法	65
2.7 高振荡函数求积法	67
2.8 计算一维积分的有理分式法	70
2.9 变步长辛卜生二重积分法	73
2.10 计算二重积分的有理分式法	77
2.11 计算多重积分的高斯方法	81
第三章 线性代数方程组的求解	86
3.1 全选主元高斯消去法	86
3.2 复系数方程组的全选主元高斯消去法	89
3.3 全选主元高斯-约当消去法	93

3.4	复系数方程组的全选主元高斯-约当消去法	97
3.5	求解三对角线方程组的追赶法	100
3.6	一般带型方程组的求解	104
3.7	求解对称方程组的分解法	109
3.8	求解对称正定方程组的平方根法	113
3.9	求解大型稀疏方程组的全选主元高斯-约当消去法	116
3.10	求解托伯利兹方程组的列文逊方法	119
3.11	高斯-赛德尔迭代法	124
3.12	求解线性最小二乘问题的豪斯荷尔德变换法	126
3.13	求解线性最小二乘问题的广义逆法	130
第四章	矩阵运算	133
4.1	实矩阵相乘	133
4.2	复矩阵相乘	135
4.3	一般实矩阵求逆的全选主元高斯-约当法	137
4.4	一般复矩阵求逆的全选主元高斯-约当法	141
4.5	对称正定矩阵的求逆	147
4.6	托伯利兹矩阵求逆的特兰特方法	150
4.7	求广义逆的奇异值分解法	154
4.8	求行列式值的全选主元高斯消去法	157
4.9	对称正定矩阵的乔里斯基分解及行列式的求值	160
4.10	矩阵的三角分解	162
4.11	一般实矩阵的 QR 分解	165
4.12	一般实矩阵的奇异值分解	169
第五章	矩阵特征值及特征向量的计算	186
5.1	约化对称矩阵为对称三对角阵的豪斯荷尔德变换法	186
5.2	实对称三对角阵的全部特征值及相应特征向量的计算	191
5.3	约化一般实矩阵为赫申伯格矩阵的初等相似变换法	195
5.4	求赫申伯格矩阵全部特征值的 QR 方法	197
5.5	求实对称矩阵特征值及特征向量的雅可比法	204
第六章	非线性方程(组)的求解	209
6.1	对分法	209
6.2	牛顿法	212
6.3	非线性方程的有理分式解法	214
6.4	求实系数代数方程全部根的 QR 方法	217
6.5	求实系数代数方程全部根的牛顿-下山法	219
6.6	求复系数代数方程全部根的牛顿-下山法	225
6.7	求非线性方程组一组实根的梯度法	231
6.8	求非线性方程组一组实根的拟牛顿法	235
6.9	求解无约束条件下优化问题的广义逆法	239
第七章	常微分方程(组)的求解	247
7.1	全区间积分的定步长欧拉方法	247
7.2	积分一步的变步长欧拉方法	250

7.3	定步长维梯方法	254
7.4	全区间积分的定步长龙格-库塔法	257
7.5	积分一步的变步长龙格-库塔法	261
7.6	积分一步的变步长基尔方法	264
7.7	全区间积分的变步长基尔方法	269
7.8	全区间积分的变步长默森方法	273
7.9	积分一步的有理分式法	277
7.10	全区间积分的有理分式法	283
7.11	全区间积分的双边法	287
7.12	全区间积分的阿当姆斯预报-校正法	291
7.13	全区间积分的哈明方法	295
7.14	积分一步的特雷纳方法	300
7.15	全区间积分的特雷纳方法	305
7.16	吉尔方法	310
7.17	二阶微分方程边值问题的数值解法	327
第八章	极值问题	332
8.1	一维极值有理法	332
8.2	n 维极值有理法	336
8.3	不等式约束线性规划问题	340
8.4	求 n 维极值的单形调优法	345
8.5	求约束条件下 n 维极值的复形调优法	351
第九章	拟合与逼近	360
9.1	最小二乘曲线拟合	360
9.2	切比雪夫曲线拟合	365
9.3	矩形域的最小二乘曲面拟合	369
9.4	最佳一致逼近的里米兹方法	377
9.5	一般平面域的最小二乘曲面拟合	382
第十章	数据处理与回归分析	391
10.1	随机样本分析	391
10.2	一元线性回归分析	395
10.3	多元线性回归分析	397
10.4	逐步回归分析	402
第十一章	数学变换与滤波	415
11.1	傅里叶级数逼近	415
11.2	快速傅里叶变换	418
11.3	快速沃什变换	426
11.4	五点三次平滑	428
11.5	离散随机线性系统的卡尔曼滤波	430
11.6	$\alpha\beta\gamma$ 滤波	437
第十二章	特殊函数	442
12.1	伽马函数	442
12.2	不完全伽马函数	444

12.3	误差函数	447
12.4	第一类整数阶贝塞耳函数	449
12.5	第二类整数阶贝塞耳函数	454
12.6	变型第一类整数阶贝塞耳函数	459
12.7	变型第二类整数阶贝塞耳函数	464
12.8	不完全贝塔函数	468
12.9	正态分布函数	471
12.10	t-分布函数	473
12.11	χ^2 -分布函数	475
12.12	F-分布函数	476
12.13	正弦积分	478
12.14	余弦积分	480
12.15	指数积分	481
12.16	第一类椭圆积分	483
12.17	第二类椭圆积分	486

第二篇 非数值计算

第十三章	排序	490
13.1	整数冒泡排序	490
13.2	实数冒泡排序	492
13.3	字符冒泡排序	494
13.4	字符串冒泡排序	495
13.5	整数快速排序	497
13.6	实数快速排序	500
13.7	字符快速排序	503
13.8	字符串快速排序	505
13.9	整数希尔排序	507
13.10	实数希尔排序	509
13.11	字符希尔排序	511
13.12	字符串希尔排序	513
13.13	整数堆排序	515
13.14	实数堆排序	517
13.15	字符堆排序	520
13.16	字符串堆排序	521
13.17	关键字成员为整数的结构排序	523
13.18	关键字成员为实数的结构排序	527
13.19	关键字成员为字符的结构排序	531
13.20	关键字成员为字符串的结构排序	534
13.21	磁盘文件排序	538
13.22	拓扑分类	543
第十四章	查找	546

14.1	关键字成员为整型、实型及字符的结构体数组的顺序查找	546
14.2	关键字成员为字符串的结构体数组的顺序查找	548
14.3	随机磁盘文本文件的顺序查找	550
14.4	整型有序表的对分查找	553
14.5	实型有序表的对分查找	556
14.6	字符串有序表的对分查找	558
14.7	整型、实型及字符型关键字成员有序的结构体数组的对分查找	561
14.8	字符串关键字成员有序的结构体数组的对分查找	564
14.9	按关键字有序的随机磁盘文本文件的对分查找	568
14.10	顺序磁盘文本文件的字符串匹配	572
第十五章	屏幕操作	576
15.1	设置光标	576
15.2	确定光标位置	578
15.3	清除屏幕	578
15.4	保存屏幕	579
15.5	恢复屏幕	580
15.6	复制屏幕	582
15.7	滚动屏幕	584
15.8	按属性显示字符	586
15.9	分屏显示文件	588
15.10	翻页分屏显示文件	591
15.11	读当前显示状态	592
15.12	菜单显示与选择	596
15.13	设置显示方式	597
第十六章	图形操作	597
16.1	读写象点	597
16.2	直线	600
16.3	矩形及其填充	602
16.4	由圆心、起点及圆心角画圆弧	603
16.5	由圆心、起点及圆心角画扇形及填充	607
16.6	由圆心及半径画圆及填充	612
16.7	由圆上三点画圆及填充	615
16.8	图形的复制与平移	618
16.9	图形的保存与恢复	620
16.10	椭圆及填充	623
16.11	抛物线	626
16.12	双曲线	629
16.13	三次多项式曲线	633
16.14	矩形区域三维图形的透视图	636
16.15	矩形区域三维图形的平行投影	639
16.16	圆形区域三维图形的平行投影	643
第十七章	其它	646

17.1	0 到 1 之间均匀分布的随机数	646
17.2	任意区间均匀分布的随机整数	648
17.3	皇后问题	650
17.4	约瑟夫问题	654
17.5	迷宫问题	655
17.6	最短距离问题	664
附录一	C 常用算法函数表	667
附录二	ROMBIOS 显示器 I/O 及键盘 I/O 中断功能表	683
附录三	屏幕显示方式	686
附录四	字符显示属性	687
附录五	关于《C 常用算法程序集》配套软件的说明	689
参考文献		690

第一篇 数值计算

第一章 插 值

1.1 一元全区间不等距插值

一、功能

给定 n 个不等距结点 x_i ($i = 0, 1, \dots, n-1$) 上的函数值 $y_i = f(x_i)$, 用拉格朗日 (Lagrange) 插值公式, 计算指定插值点 t 处的函数近似值 $z = f(t)$ 。

二、方法说明

为了避免龙格(Runge)现象对计算结果的影响, 在 n 个结点中自动选择八个结点进行插值, 且使指定插值点 t 位于它们的中间。即选取满足 $x_k < x_{k+1} < x_{k+2} < x_{k+3} < t < x_{k+4} < x_{k+5} < x_{k+6} < x_{k+7}$ 的八个结点, 利用七次拉格朗日插值多项式计算插值点 t 处的函数近似值 $z = f(t)$, 即

$$z = \sum_{i=k}^{k+7} y_i \prod_{\substack{j=k \\ j \neq i}}^{k+7} [(t - x_j)/(x_i - x_j)]$$

当插值点 t 位于包含 n 个结点的区间外时, 将仅取区间某端的四个结点进行插值; 而当插值点 t 靠近 n 个结点的两端时, 选取的结点数将少于八个。

三、函数语句

double aalgrg(x, y, n, t)

四、形参说明

x ——双精度实型一维数组, 长度为 n 。存放给定 n 个结点的值 x_i , 要求 $x_0 < x_1 < \dots < x_{n-1}$ 。

y ——双精度实型一维数组, 长度为 n 。存放 n 个给定结点上的函数值 $y_i = f(x_i)$ ($i = 0, 1, \dots, n-1$)。

n ——整型变量。给定结点的个数。

t ——双精度实型变量。存放指定插值点的值。

五、函数程序(文件名: aalgrg.c)

double aalgrg(x, y, n, t)

int n;

```
double t, x[ ], y[ ];
{
    int i, j, k, m;
    double z, s;
    z= 0.0;
    if ( n< 1) return(z);
    if ( n= = 1) { z= y[ 0];return( z); }
    if ( n= = 2)
        { z= (y[0]* (t- x[ 1]) - y[ 1]* (t- x[0]))/(x[ 0]- x[ 1]);
          return(z);
        }
    i= 0;
    while ((x[i]< t)&&(i< n)) i= i+ 1;
    k= i- 4;
    if ( k< 0) k= 0;
    m= i+ 3;
    if ( m> n- 1) m= n- 1;
    for (i= k; i< = m; i+ + )
        { s= 1.0;
          for (j= k; j< = m; j+ + )
              if (j!= i) s= s* (t- x[j])/(x[i]- x[j]);
          z= z+ s* y[i];
        }
    return( z);
}
```

六、例：已知一元不等距列表函数如下：

i	1	2	3	4	5
x _i	0.10	0.15	0.25	0.40	0.50
y _i	0.904837	0.860708	0.778801	0.670320	0.606531
i	6	7	8	9	10
x _i	0.57	0.70	0.85	0.93	1.00
y _i	0.565525	0.496585	0.427415	0.394554	0.367879

表中各函数值 y_i 具有六位有效数字, 利用拉格朗日插值公式计算插值点 $t= 0.63$ 处的函数近似值。

主函数程序(文件名: aalgrg0. c) 为:

```
# include "aalgrg. c "
```

```
# include "stdio.h "
main( )
{ double t, z;
  static double x[ 10]= {0. 10, 0. 15, 0. 25, 0. 40, 0. 50, 0. 57, 0. 70, 0. 85,
                        0. 93, 1. 00};
  static double y[ 10]= {0. 904837, 0. 860708, 0. 778801, 0. 670320, 0. 606531,
                        0. 565525, 0. 496585, 0. 427415, 0. 394554, 0. 367879};
  t= 0. 63; z= aalgrg(x, y, 10, t);
  printf( "t= %6. 3f,      z= %e\n ", t, z);
}
```

运行结果为: t= 0. 630, z= 5. 32591e- 01

1. 2 一元全区间等距插值

一、功能

给定 n 个等距结点 $x_i = x_0 + ih$ ($i = 0, 1, \dots, n-1$) 上的函数值 $y_i = f(x_i)$, 用拉格朗日 (Lagrange) 插值公式, 计算指定插值点 t 处的函数近似值 $z = f(t)$ 。

二、方法说明

同 1.1 节

三、函数语句

```
double  ablgrg(x0, h, n, y, t)
```

四、形参说明

x_0 ——双精度实型变量。等距结点中的第一个结点值。

h ——双精度实型变量。等距结点的步长。

n ——整型变量。等距结点的个数。

y ——双精度实型一维数组, 长度为 n 。存放 n 个等距结点上的函数值。

t ——双精度实型变量。指定插值点的值。

五、函数程序(文件名: ablgrg. c)

```
double ablgrg(x0, h, n, y, t)
int n;
double x0, h, t, y[ ];
{ int i, j, k, m;
  double z, s, xi, xj;
  float p, q;
  z= 0. 0;
  if ( n< 1) return(z);
  if ( n= = 1) { z= y[ 0]; return( z); }
  if ( n= = 2)
```

```
        { z= (y[1]* (t- x0)- y[0]* (t- x0- h))/h;  return(z);
        }
if (t> x0)
    { p= (t- x0)/h; i= (int)p; q= (float)i;
      if (p> q) i= i+ 1;
    }
else i= 0;
k= i- 4;
if (k< 0) k= 0;
m= i+ 3;
if (m> n- 1) m= n- 1;
for (i= k;i< = m;i+ + )
    { s= 1.0; xi= x0+ i* h;
      for (j= k; j< = m; j+ + )
          if (j!= i)
              { xj= x0+ j* h;
                s= s* (t- xj)/(xi- xj);
              }
      z= z+ s* y[i];
    }
return(z);
}
```

六、例：设函数 $y= e^{-x}$ 的函数列表如下(函数值 y_i 具有六位有效数字)：

i	1	2	3	4	5
x_i	0.1	0.2	0.3	0.4	0.5
y_i	0.904837	0.818731	0.740818	0.670320	0.606531
i	6	7	8	9	10
x_i	0.6	0.7	0.8	0.9	1.0
y_i	0.548812	0.496585	0.449329	0.406570	0.367879

计算在插值点 $t= 0.25, 0.63, 0.95$ 处的函数近似值。
在利用本函数时, 其中 $x_0= 0.1, h= 0.1, n= 10$ 。

主函数程序(文件名: ablgrg0.c) 为:

```
# include "stdio.h "
# include "ablgrg.c "
main( )
```

```

{ double x0, h, t, z;
  static double y[10] = {0.904837, 0.818731, 0.740818, 0.670320, 0.606531,
                          0.548812, 0.496585, 0.449329, 0.406570, 0.367879};

  x0= 0.1; h= 0.1;
  t= 0.25; z= ablgrg(x0, h, 10, y, t);
  printf( "t= %6.3f, z= %e\n", t, z);
  t= 0.63; z= ablgrg(x0, h, 10, y, t);
  printf( "t= %6.3f, z= %e\n", t, z);
  t= 0.95; z= ablgrg(x0, h, 10, y, t);
  printf( "t= %6.3f, z= %e\n", t, z);
}

```

运行结果为:

```

t= 0.250,      z= 7.78801e- 01
t= 0.630,      z= 5.32592e- 01
t= 0.950,      z= 3.86741e- 01

```

1.3 一元三点不等距插值

一、功能

给定 n 个不等距结点 x_i ($i = 0, 1, 2, \dots, n-1$) 上的函数值 $y_i = f(x_i)$, 用抛物插值公式计算指定插值点 t 处的函数近似值 $z = f(t)$ 。

二、方法说明

设 n 个不等距结点为 $x_0 < x_1 < \dots < x_{n-1}$, 其相应的函数值为 y_i ($i = 0, 1, \dots, n-1$)。

为了计算指定插值点 t 处的函数近似值, 选取最靠近插值点 t 的三个结点, 即如果 $x_k < t < x_{k+1}$, 则当 $x_{k-1} < t < x_{k+2}$ 时, 取三个结点为 x_k, x_{k+1}, x_{k+2} , 而当 $x_{k-1} < t < x_{k+1}$ 时, 取三个结点为 x_{k-1}, x_k, x_{k+1} , 然后用抛物插值公式计算 t 点的函数近似值, 即

$$z = \sum_{i=m}^{m+2} y_i \prod_{\substack{j=m \\ j \neq i}}^{m+2} [(t - x_j) / (x_i - x_j)]$$

其中:

当 $x_{k-1} < t < x_{k+2}$ 时, $m = k$;

当 $x_{k-1} < t < x_{k+1}$ 时, $m = k-1$ 。

如果插值点 t 位于包含 n 个结点的区间外, 则取区间某端的两个结点作线性插值。

三、函数语句

```
double  aclgrg(x, y, n, t)
```

四、形参说明

x ——双精度实型一维数组, 长度为 n 。存放给定 n 个不等距结点的值。

y ——双精度实型一维数组, 长度为 n 。存放给定 n 个不等距结点上的函数值。

n ——整型变量。给定不等距结点的个数。

t——双精度实型变量。指定插值点的值。

五、函数程序(文件名: aclgrg.c)

```
# include "math.h "
double aclgrg(x,y,n,t)
int n;
double t,x[ ],y[ ];
{ int i,j,k,m;
  double z,s;
  z= 0.0;
  if (n< 1) return(z);
  if (n== 1) { z= y[ 0]; return(z); }
  if (n== 2)
    { z= (y[0]* (t- x[ 1]))- y[ 1]* (t- x[0]))/(x[ 0]- x[ 1]);
      return(z);
    }
  if (t<= x[ 1]) { k= 0; m= 2; }
  else if (t>= x[n- 2]) { k= n- 3; m= n- 1; }
  else
    { k= 1; m= n;
      while (m- k!= 1)
        { i= (k+ m)/ 2;
          if (t< x[ i- 1]) m= i;
          else k= i;
        }
      k= k- 1; m= m- 1;
      if (fabs(t- x[k])< fabs(t- x[m])) k= k- 1;
      else m= m+ 1;
    }
  z= 0.0;
  for (i= k; i<= m; i++ )
    { s= 1.0;
      for (j= k; j<= m; j++ )
        if (j!= i) s= s* (t- x[j])/(x[i]- x[j]);
      z= z+ s* y[i];
    }
  return(z);
}
```

六、例：设给定列表函数如下(其函数值精确到末位数字)：

x	1. 615	1. 634	1. 702	1. 828	1. 921
y= f (x)	2. 41450	2. 46459	2. 65271	3. 03035	3. 34066

用抛物插值法求 f (x)在 x= 1. 682, 1. 813 处的函数近似值。

主函数程序(文件名: aclgrg0. c) 为

```
# include "stdio.h "
# include "aclgrg. c "
main( )
{ double t, z;
  static double x[ 5]= {1. 615, 1. 634, 1. 702, 1. 828, 1. 921};
  static double y[ 5]= {2. 41450, 2. 46459, 2. 65271, 3. 03035, 3. 34066};
  t= 1. 682; z= aclgrg(x, y, 5, t);
  printf( "x= % 6. 3f, f(x)= %e\n ",t, z);
  t= 1. 813; z= aclgrg(x, y, 5, t);
  printf( "x= % 6. 3f, f(x)= %e\n ",t, z);
}
```

运行结果为:

```
x= 1. 682,      f(x)= 2. 59624e+ 00
x= 1. 813,      f(x)= 2. 98281e+ 00
```

1. 4 一元三点等距插值

一、功能

给定 n 个等距结点 $x_i= x_0+ ih$ ($i= 0, 1, \dots, n- 1$) 上的函数值 $y_i= f (x_i)$, 用抛物插值公式计算指定插值点 t 处的函数近似值。

二、方法说明

同 1. 3 节。

三、函数语句

```
double adlgrg(x0, h, n, y, t)
```

四、形参说明

- x0——双精度实型变量。给定 n 个等距结点中的第一个结点值。
- h——双精度实型变量。给定 n 个等距结点的步长。
- n——整型变量。给定等距结点的个数。
- y——双精度实型一维数组, 长度为 n。存放给定 n 个等距结点上的函数值。
- t——双精度实型变量。存放指定插值点的值。

五、函数程序(文件名: adlgrg. c)

```
# include "math.h "
```

```
double adlgrg(x0, h, n, y, t)
int n;
double x0, h, t, y[ ];
{ int i, j, k, m;
  double z, s, xi, xj;
  z= 0.0;
  if ( n< 1) return(z);
  if ( n= = 1) { z= y[ 0]; return(z); }
  if ( n= = 2)
    { z= (y[1]* (t- x0)- y[ 0]* (t- x0- h))/h;
      return(z);
    }
  if ( t< = x0+ h) { k= 0; m= 2; }
  else if ( t> = x0+ (n- 3)* h) { k= n- 3; m= n- 1; }
  else
    { i= (int)((t- x0)/h)+ 1;
      if ( fabs(t- x0- i* h)> = fabs(t- x0- (i- 1)* h))
        { k= i- 2; m= i; }
      else {k= i- 1; m= i+ 1; }
    }
  z= 0.0;
  for (i= k; i< = m; i+ + )
    { s= 1.0; xi= x0+ i* h;
      for (j= k; j< = m; j+ + )
        if (j!= i)
          { xj= x0+ j* h; s= s* (t- xj)/(xi- xj); }
      z= z+ s* y[i];
    }
  return(z);
}
```

六、例：设函数 $f(x) = e^{-x}$ 在 10 个等距结点上的函数值如下表(函数值具有六位有效数字)：

x	0.1	0.2	0.3	0.4	0.5
e^{-x}	0.904837	0.818731	0.740818	0.670320	0.606531
x	0.6	0.7	0.8	0.9	1.0
e^{-x}	0.548812	0.496585	0.449329	0.406570	0.367879