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 $(i=0,1,\cdots,n-1)$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$x_0<x_1<\cdots<x_{n-1}$。

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 $y_i=f(x_i)$ $(i=0,1,\cdots,n-1)$。

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 $x_i=x_0+ih$ $(i=0,1,\cdots,n-1)$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

x0——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 x0=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 $(i=0,1,2,\cdots,n-1)$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 $x_0<x_1<\cdots<x_{n-1}$，其相应的函数值为 y_i $(i=0,1,\cdots,n-1)$。

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 $|x_k-t|>|t-x_{k+1}|$时，取三个结点为 x_k,x_{k+1},x_{k+2}，而当 $|x_k-t|<|t-x_{k+1}|$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 $|x_k-t|>|t-x_{k+1}|$时，$m=k$；

当 $|x_k-t|<|t-x_{k+1}|$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 $x_i=x_0+ih\ \ (i=0,1,\cdots,n-1)$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

计算在插值点 $t=0.23, 0.63, 0.95$处的函数近似值。

在利用本函数时，x0=0.1，h=0.1，n=10。

主函数程序(文件名：adlgrg0.c)为：

```
#include "stdio.h"
#include "adlgrg.c"
main( )
{ double t,z,x0,h;
  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=0.23; z=adlgrg(x0,h,10,y,t);
  printf("t=%6.3f, z=%e\n",t,z);
  t=0.63; z=adlgrg(x0,h,10,y,t);
  printf("t=%6.3f, z=%e\n",t,z);
  t=0.95; z=adlgrg(x0,h,10,y,t);
  printf("t=%6.3f, z=%e\n",t,z);
}
```

运行结果为：

```
t=0.230,    z=7.94497e-01
t=0.630,    z=5.32567e-01
t=0.950,    z=3.86716e-01
```

1.5 有理不等距插值

一、功能

给定 n 个不等距结点 x_i $(i=0,1,\cdots,n-1)$上的函数值 $y_i=f(x_i)$，利用有理插值法计算指定插值点 t 处的函数近似值。

二、方法说明

设给定的 n 个不等距结点为 $x_0<x_1<\cdots<x_{n-1}$，相应的函数值为 y_i $(i=0,1,\cdots,n-1)$，则可以构造一个 n 节连分式

$$\varphi(x)=b_0+\cfrac{x-x_0}{b_1+\cfrac{x-x_1}{b_2+\cdots+\cfrac{x-x_{n-2}}{b_{n-1}}}}$$

其中常数项 b_0及部分分母 $b_1,b_2,\cdots,b_{n-1}$由下列递推公式计算：

$$\varphi_0(x_j)=y_j,\quad j=0,1,\cdots,n-1$$

$$\varphi_{i+1}(x_j)=\frac{x_j-x_i}{\varphi_i(x_j)-\varphi_i(x_i)},\quad \begin{array}{l} i=0,1,\cdots,n-2 \\ j=i+1,\cdots,n-1 \end{array}$$

$$b_i = \varphi_i(x_i),\ i = 0,1,\cdots,n-1$$

在实际进行插值计算时，一般在指定插值点 t 的前后各取四个结点就够了。此时，计算八节连分式的值 $\varphi(t)$ 作为插值点 t 处的函数近似值。

三、函数语句

double aepq(x,y,n,t)

四、形参说明

x——双精度实型一维数组，长度为 n。存放给定的 n 个不等距结点值。

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：aepq.c)

```
#include "math.h"
double aepq(x,y,n,t)
int n;
double t,x[ ],y[ ];
{ int i,j,k,m,l;
  double z,h,b[8];
  z=0.0;
  if (n<1) return(z);
  if (n==1) { z=y[0]; return(z);}
  if (n<=8) { k=0; m=n;}
  else if (t<x[4]) { k=0; m=8;}
  else if (t>x[n-5]) { k=n-8; m=8;}
  else
    { k=1; j=n;
      while (j-k!=1)
        { i=(k+j)/2;
          if (t<x[i-1]) j=i;
          else k=i;
        }
      k=k-4; m=8;
    }
  b[0]=y[k];
  for (i=2;i<=m;i++)
    { h=y[i+k-1]; l=0;
      for (j=1;j<=i-1;j++)
        if (l==0)
          { if (fabs(h-b[j-1])+1.0==1.0) l=l+1;
```

```
            else h=(x[i+k-1]-x[j+k-1])/(h-b[j-1]);
          }
      b[i-1]=h;
      if (l!=0) b[i-1]=1.0e+35;
    }
  z=b[m-1];
  for (i=m-1;i>=1;i--) z=b[i-1]+(t-x[i+k-1])/z;
  return(z);
}
```

六、例：设函数 $f(x)=\dfrac{1}{1+25x^2}$ 的10个不等距结点处的函数值如下表(具有六位有效数字)：

x	-1.00	-0.80	-0.65	-0.40	-0.30
$f(x)$	0.0384615	0.0588236	0.0864865	0.200000	0.307692
x	0.00	0.20	0.45	0.80	1.00
$f(x)$	1.00000	0.500000	0.164948	0.0588236	0.0384615

利用有理插值计算 $t=-0.85$ 及 $t=0.25$ 处的函数近似值。

主函数程序(文件名：aepq0.c)为：

```
#include "stdio.h"
#include "aepq.c"
main( )
{ double t,z;
  static double x[10]={-1.0,-0.8,-0.65,-0.4,-0.3,
                       0.0,0.2,0.45,0.8,1.0};
  static double y[10]={0.0384615,0.0588236,0.0864865,0.2,0.307692,1.0,
                       0.5,0.164948,0.0588236,0.0384615};
  t=-0.85; z=aepq(x,y,10,t);
  printf("t=%6.3f, z=%e\n",t,z);
  t=0.25; z=aepq(x,y,10,t);
  printf("t=%6.3f, z=%e\n",t,z);
}
```

运行结果为：

```
t=-0.850,    z=5.24591e-02
t=0.250,     z=3.90244e-01
```

1.6 有理等距插值

一、功能

给定 n 个等距结点 $x_i=x_0+ih$ $(i=0,1,\cdots,n-1)$上的函数值 $y_i=f(x_i)$，利用有理插值法计算指定插值点 t 处的函数近似值。

二、方法说明

同1.5节。

三、函数语句

double afpq(x0,h,n,y,t)

四、形参说明

x0——双精度实型变量。n 个等距结点中的第一个结点值。

h——双精度实型变量。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 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：afpq.c)

```
#include "math.h"
double afpq(x0,h,n,y,t)
int n;
double x0,h,t,y[ ];
{ int i,j,k,m,l;
  double z,hh,xi,xj,b[8];
  z=0.0;
  if (n<1) return(z);
  if (n==1) { z=y[0]; return(z);}
  if (n<=8) { k=0; m=n;}
  else if (t<(x0+4.0*h)) { k=0; m=8;}
  else if (t>(x0+(n-5)*h)) { k=n-8; m=8;}
  else { k=(int)((t-x0)/h)-3; m=8;}
  b[0]=y[k];
  for (i=2;i<=m;i++)
    { hh=y[i+k-1]; l=0;
      for (j=1;j<=i-1;j++)
        if (l==0)
          { if (fabs(hh-b[j-1])+1.0==1.0) l=l+1;
            else
              { xi=x0+(i+k-1)*h;
```

```
              xj=x0+(j+k-1)*h;
              hh=(xi-xj)/(hh-b[j-1]);
            }
        }
        b[i-1]=hh;
        if (l!=0) b[i-1]=1.0e+35;
      }
    z=b[m-1];
    for (i=m-1;i>=1;i--)
      z=b[i-1]+(t-(x0+(i+k-1)*h))/z;
    return(z);
  }
```

六、例：设函数 $f(x)=\dfrac{1}{1+25x^2}$ 的11个等距结点上的函数值如下表(具有六位有效数字)：

x	-1.00	-0.80	-0.60	-0.40	-0.20	0.00
$f(x)$	0.0384615	0.0588236	0.100000	0.200000	0.500000	1.00000
x	0.20	0.40	0.60	0.80	1.00	
$f(x)$	0.500000	0.200000	0.100000	0.0588236	0.0384615	

利用有理插值法计算在插值点 $t=-0.75$ 及 $t=-0.05$ 处的函数近似值。

在利用本函数时，x0=-1.00，h=0.20，n=11。

主函数程序(文件名：afpq0.c)为

```
#include "stdio.h"
#include "afpq.c"
main( )
{ double x0,h,t,z;
  static double y[11]={0.0384615,0.0588236,0.1,0.2,0.5,1.0,0.5,0.2,0.1,
                       0.0588236,0.0384615};
  x0=-1.0; h=0.2;
  t=-0.75; z=afpq(x0,h,11,y,t);
  printf("t=%6.3f, z=%e\n",t,z);
  t=-0.05; z=afpq(x0,h,11,y,t);
  printf("t=%6.3f, z=%e\n",t,z);
}
```

运行结果为：

```
t=-0.750,    z=6.63901e-02
```

t=−0.050,　　z=9.41176e−01

1.7　埃尔米特不等距插值

一、功能

给定 n 个不等距结点 x_i $(i=0,1,\cdots,n-1)$ 上的函数值 $f(x_i)$ 及一阶导数值 $f'(x_i)$，用埃尔米特(Hermite)插值公式计算指定插值点 t 处的函数近似值 $f(t)$。

二、方法说明

设函数 $f(x)$ 在 n 个不等距结点 $x_0<x_1<\cdots<x_{n-1}$ 上的函数值及一阶导数值分别为：

$$y_i = f(x_i),\ y'_i = f'(x_i),\ i = 0,1,\cdots,n-1$$

则 $f(x)$ 可用埃尔米特插值多项式

$$P_{2n-1}(x) = \sum_{i=0}^{n-1}[y_i + (x - x_i)(y'_i - 2y_i l'_i(x_i))]l_i^2(x)$$

近似代替。其中：

$$l_i(x) = \prod_{\substack{j=0\\j\neq i}}^{n-1}[(x - x_j)/(x_i - x_j)]$$

$$l'_i(x_i) = \sum_{\substack{j=0\\j\neq i}}^{n-1}\frac{1}{x_i - x_j}$$

在实际进行插值计算时，为了减少计算工作量，用户可以适当地将远离插值点的结点抛弃。通常，只需取插值点 t 的前后各四个结点就够了。

三、函数语句

```
double  aghmt(x,y,dy,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。

dy——双精度实型一维数组，长度为 n。存放给定 n 个不等距结点上的一阶导数值。

n——整型变量。给定的不等距结点个数。

t——双精度实型变量。指定插值点的值。

五、函数程序(文件名：aghmt.c)

```
double aghmt(x,y,dy,n,t)
int n;
double t,x[ ],y[ ],dy[ ];
{ int i,j;
  double z,p,q,s;
  z=0.0;
  for (i=1;i<=n;i++)
    { s=1.0;
      for (j=1;j<=n;j++)
```

```
      if (j!=i) s=s*(t-x[j-1])/(x[i-1]-x[j-1]);
    s=s*s;
    p=0.0;
    for (j=1;j<=n;j++)
      if (j!=i) p=p+1.0/(x[i-1]-x[j-1]);
    q=y[i-1]+(t-x[i-1])*(dy[i-1]-2.0*y[i-1]*p);
    z=z+q*s;
  }
  return(z);
}
```

六、例：设函数 $f(x)=e^{-x}$在10个不等距结点上的函数值及一阶导数值如下表(具有六位有效数字)：

x	0.10	0.15	0.30	0.45	0.55
$f(x)$	0.904837	0.860708	0.740818	0.637628	0.576950
$f'(x)$	−0.904837	−0.860708	−0.740818	−0.637628	−0.576950
x	0.60	0.70	0.85	0.90	1.00
$f(x)$	0.548812	0.496585	0.427415	0.406570	0.367879
$f'(x)$	−0.548812	−0.496585	−0.427415	−0.406570	−0.367879

利用埃尔米特插值公式计算在插值点 $t=0.356$处的函数近似值 $f(t)$。

主函数程序(文件名：aghmt0.c)为：

```
#include "stdio.h"
#include "aghmt.c"
main( )
{ int i;
  double t,z;
  static double x[10]={0.1,0.15,0.3,0.45,0.55,0.6,0.7,0.85,0.9,1.0};
  static double y[10]={0.904837,0.860708,0.740818,0.637628,0.576950,
                       0.548812,0.496585,0.427415,0.406570,0.367879};
  double dy[10];
  for (i=0;i<=9;i++) dy[i]=-y[i];
  t=0.356; z=aghmt(x,y,dy,10,t);
  printf("t=%6.3f, z=%e\n",t,z);
}
```

运行结果为： t=0.356，   z=7.00480e-01

1.8 埃尔米特等距插值

一、功能

给定 n 个等距结点 x_i $(i=0,1,\cdots,n-1)$上的函数值 $f(x_i)$及一阶导数值 $f'(x_i)$，利用埃尔米特(Hermite)插值公式计算指定插值点 t 处的函数近似值 $f(t)$。

二、方法说明

同1.7节。

三、函数语句

```
double  ahhmt(x0,h,n,y,dy,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。

dy——双精度实型一维数组，长度为 n。存放给定 n 个等距结点上的一阶导数值。

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：ahhmt.c)

```
double ahhmt(x0,h,n,y,dy,t)
int n;
double x0,h,t,y[ ],dy[ ];
{ int i,j;
  double z,s,p,q;
  z=0.0;
  for (i=1;i<=n;i++)
    { s=1.0; q=x0+(i-1)*h;
      for (j=1;j<=n;j++)
        { p=x0+(j-1)*h;
          if (j!=i) s=s*(t-p)/(q-p);
        }
      s=s*s;
      p=0.0;
      for (j=1;j<=n;j++)
        if (j!=i) p=p+1.0/(q-(x0+(j-1)*h));
      q=y[i-1]+(t-q)*(dy[i-1]-2.0*y[i-1]*p);
      z=z+q*s;
    }
  return(z);
```