快速 Fourier 变换算法及 Matlab 程序实现①

王海鹃
(通化师范学院　数学系,吉林 通化 134002)

摘　要:介绍了快速 Fourier 变换算法(FFT)的核心思想及其算法描述,并用 Matlab 程序设计语言实现了 FFT 算法. 最后,举例说明用 FFT 算法计算复函数 f(x) 的插值函数.

关键词:FFT;算法描述;Matlab 程序
中图分类号:O241　**文献标识码**:A　**文章编号**:1008 - 7974(2006)04 - 0013 - 03

1　引言

在 1965 年,卜力和塔基在 *Mathematica Of Computation* 上发表了一篇论文,在此论文中,他们采用一种方法来求取内插三角多项式的常数项,如果我们选取适当的 N,则只须作 $O(N\log_2^N)$ 次的乘法运算及 $O(N\log_2^N)$ 的加法运算. 若一个问题有几千个数据,则只须作数千次的运算,而非上百万次. 通常把这类工作量为 $N\log_2^N$ 的算法统称为快速 Fourier 变换.

快速 Fourier 变换算法(Fast Fourier Transform Algorithm, FFT)使许多科学领域在内插三角多项式的使用上有了革命性的进展. 此方法须将数据点的数目能够加以分解成为某一乘幂的形式,通常是以 2 的乘幂为数目,即逐次分半.

2　快速 Fourier 变换算法的核心思想

FFT 算法的核心思想就是尽量减少乘法次数. 在逐次分半法中恒取 $N = 2^p$,并把一次 N 点变换的过程分解为 $p = N\log_2^N$ 步,每步计算一个简化的傅氏变换,工作量为 N 个复运算,因而总工作量为 $N\log_2^N$. 由于分点数 N 的选取在多数场合下人们有主动权,因此取 $N = 2^p$ 这一特殊形式并不是严重的限制. 现用公式 $G_j = \sum_{k=0}^{N-1} g_k W^{jk}\ (j = 0,1,\cdots,N-1)$ 计算全部 G_j,其中 $W = e^{-i\frac{2\pi}{N}}$ (正变换)或 $W = e^{i\frac{2\pi}{N}}$ (逆变换),$\{g_k\}\ (k = 0,1,\cdots,N-1)$ 是已知复数数列. 表面上看有 N^2 个 W^{jk} 的项,而这 N^2 个项中实际上只有 N 个不同的值 $W^0, W^1, \cdots, W^{N-1}$. 把 g_k 中各项先按 $W^r\ (0 \le r \le N-1)$ 归类,然后把同类项中 g_k 先加起来再和 W^r 相乘,这样就能大量地减少乘法运算的次数.

3　快速 Fourier 变换的算法描述

下面是执行 $m = 2^p$ 时的快速 Fourier 变换算法,式中 p 为一个正整数.

就数据点 $\{(x_j, y_j)\}\mid_{j=0}^{2m-1}$,式中 $m = 2^p$, $x_j = -\pi + j\pi/m\ (j = 0,1,\cdots,2m-1)$,计算离散近似式 $F(x) = \frac{1}{m}\sum_{k=0}^{2m-1} c_k e^{ikx} = \frac{1}{m}\sum_{k=0}^{2m-1} c_k(\cos(kx) + i\sin(kx))$,式中 $i = \sqrt{-1}$.

Input: $m, p; y_0, y_1, \cdots, y_{2m-1}$.
Output: 复数 $c_0, \cdots, c_{2m-1}$;实数 $a_0, \cdots, a_m; b_1, \cdots, b_{m-1}$.
Step1　初值 $M = m$; $q = p$; $\zeta = e^{\pi i/m}$.
Step2　循环 $j = 0,1,\cdots,2m-1$ 执行 $c_j = y_j$.

① **收稿日期**:2006 - 03 - 15
作者简介:王海鹃(1979 -),女,通化师范学院数学系教师、硕士.

Step3 **循环** $j = 1,2,\cdots,M$ **执行** $\varepsilon_j = \zeta, \varepsilon_{j+M} = -\varepsilon_j$.

Step4 **令** $K = 0; \varepsilon_0 = 1$.

Step5 **循环** $L = 1,2,\cdots p+1$ **执行** Step6～Step12

Step6 While($K < 2m-1$) **执行** Step7～Step11

Step7 **循环** $j = 1,2,\cdots,M$ **执行** Step8～Step10

Step8 $K = k_p \cdot 2^p + k_{p-1} \cdot 2^{p-1} + \cdots + k_1 \cdot 2^1 + k_0$;(**分解** k)

$K_1 = K/2^q$;

$K_2 = k_q \cdot 2^p + k_{q+1} \cdot 2^{p-1} + \cdots + k_p \cdot 2^q$;

Step9 $\eta = c_{K+M} \cdot \varepsilon_{K_1}$; $C_{K+M} = C_K - \eta$; $c_k = c_K + \eta$

Step10 $K = K+1$

Step11 $K = K+M$

Step12 $K = 0; M = M/2$; $q = q-1$

Step13 While($K < 2m-1$) **执行** Step14～Step16

Step14 $K = k_p \cdot 2^p + k_{p-1} \cdot 2^{p-1} + \cdots + k_1 \cdot 2 + k_0$;(**分解** k)

$j = k_0 \cdot 2^p + k_1 \cdot 2^{p-1} + \cdots + k_{p-1} + k_p$;

Step15 If($j > k$) Then swap(c_j, c_k)

Step16 $K = K+1$

Step17 $a_0 = c_0/m; a_m = \mathrm{Re}al(e^{-m\pi} c_m/m)$

Step18 **循环** $j = 1,\cdots,m-1$ **执行** $a_j = \mathrm{Re}al(e^{-j\pi} c_j/m); b_j = \mathrm{Im}ag(e^{-j\pi} c_j/m)$

4 FFT 算法的 Matlab 程序实现

4.1 快速 Fourier 变换(FFT)算法 Matlab 程序如下:($N = 2^p$)

```
function z = FFTmatlab(a)
  N = length(a) ;N1 = N ;p = 0 ;
  while p < = N
    p = p + 1 ; r = N1/2 ;
    if r = = 1
      break
    end
    N1 = r ;
  end
  w = exp( - i *2 *pi/N) ;
  for q = 1 :p
    n = 2^q ;
for k = 0 :N/n - 1
  for j = 0 :n/2 - 1
    m = k *n ;
    b(m + j + 1) = a(m/2 + j + 1) + a(m/2 + j + 1 + N/2) ;
    b(m + j + 1 + n/2) = (a(m/2 + j + 1) - a(m/2 + j + 1 + N/2))
     *w^(m/2) ;
  end
end
a = b ;
end
z = a ;
```

4.2 用 FFT 算法计算复函数 f(x) 的插值函数程序如下:

```
function z = insert(f ,t)
  N = length(f) ;
  c = FFTmatlab(f) / N ;
  s = 0 ;
for i = 1 :N
  s = s + c(i) *exp(j *(i - 1) *t) ;
end
z = s ;
```

4.3 用 FFT 算法计算复函数 f(x) 的插值函数绘图程序如下:

```
function whj(f)
  tic ;
  N = length(f) ;
  for k = 1 :N
    x(k) = real(f(k)) ;
    y(k) = imag(f(k)) ;
    z(k) = abs (f(k)) ;
  end
  for k = 1 :1001
    s = insert(f ,2 *pi *(k - 1)/1000) ;
    xx(k) = real(s) ;
    yy(k) = imag(s) ;
    zz(k) = abs (s) ;
  end
  figure(N)
  plot (x, y, 'b * ',xx ,yy, 'r - ')
  title( '插值复函数的复平面图形')
  x1 = max(xx) ;y1 = max(yy) ;x2 = min(xx) ;
  y2 = min(yy) ;z1 = max (zz) ;
  text(x1 ,0 , '实轴')
```

```
text(0,y1,'虚轴')
text(0,0,'O-原点')
hold on
xxx=[x2 x1];yyy=[y2 y1];
zzz=[0 z1];o=[0 0];
plot(xxx,o,'k-',o,yyy,'k-')
figure(N+50)
plot3(x,y,z,'b*',xx,yy,zz,'r-')
title('插值复函数的模函数图形')
text(x1,0,0,'实轴')
text(0,y1,0,'虚轴')
text(0,0,z1,'模轴')
text(0,0,0,'O-原点')
hold on
plot3(xxx,o,o,'k-',o,yyy,o,'k-',o,o,zzz,'k-')
toc
```

5 结束语

MATLAB 软件具有强大的数值计算功能,可以处理诸如矩阵计算,插值和拟合计算,完成各种方程求解和优化问题等;具有方便的绘图功能和完善的图形可视化功能,已成为强有力的计算工具. 快速 Fourier 变换算法,它在数据处理问题中能成万倍地节省计算时间,使得应用 FFT 解决许多实际问题成为可能.

参考文献:

[1]莱特希尔. 傅里叶变换与广义函数[M]. 北京:科学出版社,1985.
[2]黄友谦、李岳生. 数值逼近[M]. 北京:人民教育出版社,1987.
[3]徐利治、王仁宏、周蕴时. 函数逼近的理论与方法[M]. 上海:上海科技出版社,1983.
[4]李庆扬、王能超、易大义. 数值分析(第三版)[M]. 武汉:华中工学院出版社,1987.
[5]张彦仲、沈乃汉. 快速傅里叶变换及沃尔什变换[M]. 北京:航空工业出版社,1989.
[6]楼顺天、陈生谭、雷虎民. MATLAB 程序设计语言[M]. 西安:西安电子科技大学出版社,2000.

Fast Fourier Transform Algorithm and Matlab Procedure's Realization

WANG Hai-juan

(Department of Mathematics, Tonghua Teachers College, Tonghua, Jilin, 134002)

Abstract: The paper introduces the core thought and algorithm description of fast fourier transform (FFT) algorithm, and carried out FFT algorithm with the Matlab program design language. Finally, explains with example how to calculate duplicate function f(x) interpolating function with FFT algorithm.

Key words: FFT; algorithm described; matlab procedure.