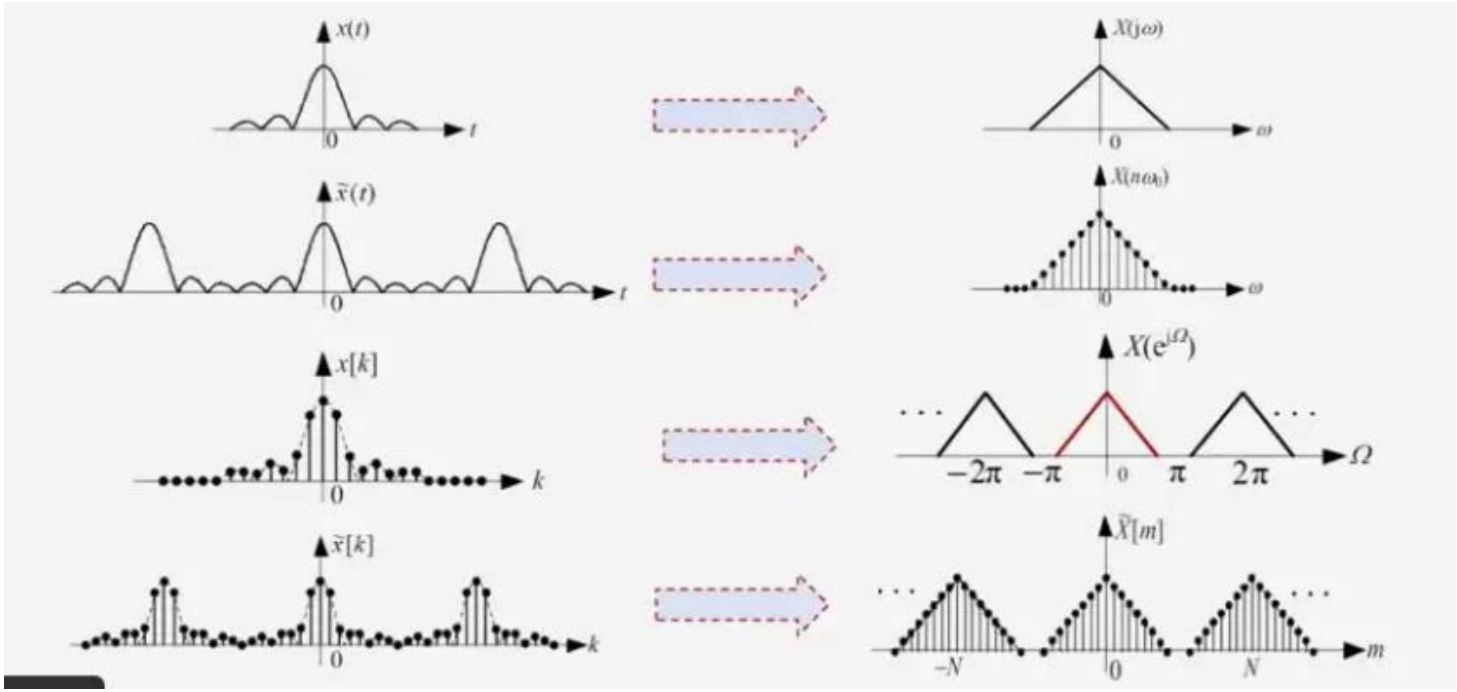
第十章 快速傅里叶变换

$$f(t)=\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}F(\omega)e^{i\omega t}d\omega \qquad F(\omega)=\frac{1}{\sqrt{2\pi}}\int_{0}^{T}f(t)e^{-i\omega t}dt$$



应用：

$$a(t)f''(t)+b(t)f'(t)+c(t)=g(t)$$

$$A(\omega)(i\omega)^2F(\omega)+B(\omega)(i\omega)F(\omega)+C(\omega)=G(\omega)$$

离散傅里叶变换（DFT）

$$X(k)=\sum_{n=0}^{N-1}x(n)W_N^{nk},\qquad k=0,1,\cdots,N-1$$

$$x(n)=\frac{1}{N}\sum_{k=0}^{N-1}X(k)W_N^{-nk},\qquad n=0,1,\cdots,N-1$$

$$X(1)=x(0)W_N^0+x(1)W_N^1+x(2)W_N^2+\cdots+x(N-1)W_N^{N-1}$$

N次复数乘法，N－1次复数加法 ✗ N

快速傅里叶变换（FFT）

$$W_N^{nk} = e^{-j\frac{2\pi}{N}kn}$$

周期性：$W_N^{m+lN} = W_N^m$

对称性：$W_N^{m+\frac{N}{2}} = -W_N^m$

1965年，图基和库利提出了**快速傅里叶变换**（FFT），
不断把长序列分解成短序列，再进行DFT，
并利用周期性和对称性来减少DFT的运算次数。

对于一个长度为 N 的离散信号来讲，我们对其取离散傅里叶变换有：

$$X(k) := \mathbf{DFT}\left\{x(n)\right\} = \sum_{n=0}^{N-1} x(n) \cdot e^{-j\frac{2\pi}{N}nk}, \quad k = 0, 1, \ldots, n-1 \quad (1)$$

其中 $\mathbf{DFT}\left\{\bullet\right\}$ 是 $\bullet$ 的**离散傅里叶变换**，其逆变换为：

$$x(n) = \mathbf{IDFT}\left\{X(k)\right\} = \frac{1}{N}\sum_{k=0}^{N-1} X(k) \cdot e^{j\frac{2\pi}{N}nk}, \quad n = 0, 1, \cdot, N-1 \quad (2)$$

其中 $\mathbf{IDFT}\left\{\bullet\right\}$ 是 $\bullet$ 的**离散逆傅里叶变换**。

从式 (1) 中我们可以发现，如果要求第 k 点的 $\mathbf{DFT}$ 值，我们需要做：

(i)：

$$X(k)=\sum_{n=0}^{N-1}\bullet \tag{3}$$

$N-1$ 次加法；和：

(ii)：

$$x(0)\cdot e^{-j\frac{2\pi}{N}0k}+x(1)\cdot e^{-j\frac{2\pi}{N}1k}+\ldots+x(N-1)\cdot e^{-j\frac{2\pi}{N}(N-1)k} \tag{4}$$

N 次乘法运算。这就意味着如果要计算 N 个 $X(k),\ (k=0,1,\ldots,N-1)$ 的值那么总共需要计算 N^2 次的乘法和 $N\cdot(N-1)$ 次的加法。这样的计算量大小在 N 比较小的时候是体现不出来的，但是当 N 很大时这样的计算量是十分庞大的。所以，我们要设法改进算法，来减少运算量。

那我们应该从何处入手来解决优化这个算法呢？答案就是从表达式本身入手来解决。因为在式 (1) 中我们发现，求和项中含有 $e^{-j\frac{2\pi}{N}nk}$ 这一项，这一项的性质可以说是相当不错的。下面我们就先来研究一下这一项的性质。

我们定义：

$$W_N := e^{-j\frac{2\pi}{N}}, \quad W_N^{nk} := e^{-j\frac{2\pi}{N}nk} \tag{5}$$

将式 (5) 代入到式 (1) 中我们可以得到：

$$X(k) := \mathbf{DFT}\left\{x(n)\right\} = \sum_{n=0}^{N-1} x(n) \cdot W_N^{nk}, \quad k = 0, 1, \ldots, n-1 \tag{6}$$

比如，我们现在有一个长度为 $N=4$ 的信号 $x(n)$ ，则通过式 (6) 可以得到：

$$X(k) := \mathbf{DFT}\left\{x(n)\right\} = \sum_{n=0}^{3} x(n) \cdot W_4^{nk}, \quad k = 0, 1, 2, 3 \tag{7}$$

当 $k = 0, 1, 2, 3$ 时，我们可以得到下面的关系式：

$$\begin{aligned}
k = 0 &: X(0) = \textstyle\sum_{n=0}^{3} x(n) \cdot W_4^{nk} = x(0) \cdot W_4^0 + x(1) \cdot W_4^0 + x(2) \cdot W_4^0 + x(3) \cdot W_4^0 \\
k = 1 &: X(1) = \textstyle\sum_{n=0}^{3} x(n) \cdot W_4^{nk} = x(0) \cdot W_4^0 + x(1) \cdot W_4^1 + x(2) \cdot W_4^2 + x(3) \cdot W_4^3 \\
k = 2 &: X(2) = \textstyle\sum_{n=0}^{3} x(n) \cdot W_4^{nk} = x(0) \cdot W_4^0 + x(1) \cdot W_4^2 + x(2) \cdot W_4^4 + x(3) \cdot W_4^6 \\
k = 3 &: X(3) = \textstyle\sum_{n=0}^{3} x(n) \cdot W_4^{nk} = x(0) \cdot W_4^0 + x(1) \cdot W_4^3 + x(2) \cdot W_4^6 + x(3) \cdot W_4^9
\end{aligned} \tag{8}$$

将式 (8) 写为矩阵形式为：

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} W_4^0 & W_4^0 & W_4^0 & W_4^0 \\ W_4^0 & W_4^1 & W_4^2 & W_4^3 \\ W_4^0 & W_4^2 & W_4^4 & W_4^6 \\ W_4^0 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix} \cdot \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ x(3) \end{bmatrix} \tag{9}$$

利用性质

$$W_N^{mN} = W_N^0 = 1$$

$$W_4^{nk} \overset{(19)}{=} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & W_4^2 & W_4^3 \\ 1 & W_4^2 & 1 & W_4^6 \\ 1 & W_4^3 & W_4^6 & W_4^9 \end{bmatrix}$$

利用

$$W_4^2 = W_4^{0+\frac{4}{2}} = -1 = W_4^{1\cdot 4+\frac{4}{2}} = W_4^6$$

进一步化简为

$$W_4^{nk} \overset{(19),(25)}{=} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & W_4^1 & -1 & W_4^3 \\ 1 & -1 & 1 & -1 \\ 1 & W_4^3 & -1 & W_4^9 \end{bmatrix}$$

最终得到

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & W_4^1 & -W_4^1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -W_4^1 & W_4^1 \end{bmatrix} \cdot \begin{bmatrix} x(0) \\ x(2) \\ x(1) \\ x(3) \end{bmatrix}$$

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ X(3) \end{bmatrix} \overset{(33),(34)}{=} \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \\ \hline \boldsymbol{C} & \boldsymbol{D} \end{array}\right] \cdot \begin{bmatrix} \vec{x}_{\text{even}} \\ \vec{x}_{\text{odd}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A} \cdot \vec{x}_{\text{even}} + \boldsymbol{B} \cdot \vec{x}_{\text{odd}} \\ \boldsymbol{C} \cdot \vec{x}_{\text{even}} + \boldsymbol{D} \cdot \vec{x}_{\text{odd}} \end{bmatrix}$$

这样就简化了计算

六、例

设函数

$$P(t)=e^{-t},t\geqslant 0$$

取 N＝64，K＝6，周期 T＝6.4，则步长 $h=\frac{T}{N}=0.1$，且取等分区间的中点，即

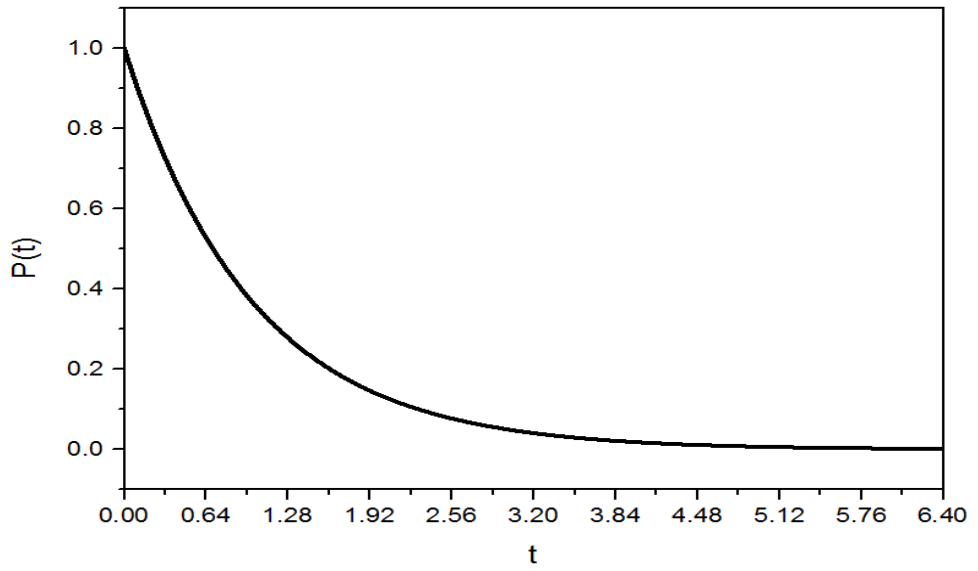
$$P_i=e^{-(i-0.5)h},\ i=1,2,\cdots,N$$

计算：

(1) P_i 的离散傅里叶变换 F_i，且计算 F_i 的模与幅角。即 L＝0，IL＝1。

(2) F_i 的逆傅里叶变换 p_i 的模。即 L＝1，IL＝1。

主程序(文件名：KKFFT0.FOR)为



快速傅里叶变换

一、功能

用 FFT 计算离散傅里叶(Fourier)变换。

二、方法说明

计算 n 个采样点

$$P=\{p_0,p_1,\cdots,p_{n-1}\}$$

的离散傅里叶变换，可以归结为计算多项式

$$F(x)=p_0+p_1x+p_2x^2+\cdots+p_{n-1}x^{n-1}$$

在各 n 次单位根 $1,\omega,\omega^2,\cdots,\omega^{n-1}$ 上的值，即

$$F_0=p_0+p_1+p_2+\cdots+p_{n-1}$$

$$F_1=p_0+p_1\omega+p_2\omega^2+\cdots+p_{n-1}\omega^{n-1}$$

$$F_2=p_0+p_1\omega^2+p_2(\omega^2)^2+\cdots+p_{n-1}(\omega^2)^{n-1}$$

$$\cdots\cdots$$

$$F_{n-1}=p_0+p_1\omega^{n-1}+p_2(\omega^{n-1})^2+\cdots+p_{n-1}(\omega^{n-1})^{n-1}$$

其中 $\omega=e^{-j2\pi/n}$($j=\sqrt{-1}$)为 n 次单位元根。

若 n 是 2 的 k 次幂，即 $n=2^k(k>0)$。则 $F(x)$ 可以分解为关于 x 的偶次幂与奇次幂两部分，即

$$F(x)=p_0+p_2x^2+\cdots+p_{n-2}x^{n-2}+x(p_1+p_3x^2+\cdots+p_{n-1}x^{n-2})$$

若令

$$P_{even}(x^2)=p_0+p_2x^2+\cdots+p_{n-2}x^{n-2}$$
$$P_{odd}(x^2)=p_1+p_3x^2+\cdots+p_{n-1}x^{n-2}$$

则有

$$F(x)=P_{even}(x^2)+xP_{odd}(x^2)$$

并且有

$$F(-x)=P_{even}(x^2)-xP_{odd}(x^2)$$

由此可以看出，为求 F 在各 n 次单位根上的值，只要求 P_{even} 和 P_{odd} 在 $1,\omega^2,\cdots,(\omega^{(n/2)-1})^2$ 上的值就行了。

而 P_{even} 和 P_{odd} 同样可以分解成关于 x 的偶次幂和奇次幂两部分。以此类推，一直分解

下去，最后只要求二次单位根 1 与－1 上的值。

在实际计算时，可以将上述过程倒过来进行，这就是 FFT 算法。

三、子程序语句

SUBROUTINE KKFFT(PR,PI,N,K,FR,FI,L,IL)

四、形参说明

PR,PI——均为双精度实型一维数组，长度为 N，输入兼输出参数。调用时分别存放 N 个采样输入的实部与虚部(当 L＝0 时，即计算傅里叶变换)或傅里叶变换的 N 个实部与虚部(当 L＝1 时，即计算逆傅里叶变换)；返回时分别存放傅里叶变换的模与幅角(当 L＝0 时)或逆傅里叶变换的模与幅角(当 L＝1 时)，其中幅角的单位为度。

N——整型变量，输入参数。输入的点数，要求 $N=2^K(K>0)$。

K——整型变量，输入参数。要求满足 $N=2^K(K>0)$。

FR,FI——均为双精度实型一维数组，长度为 N，输出参数。分别存放傅里叶变换的实部与虚部(当 L＝0 时)或逆傅里叶变换的实部与虚部(当 L＝1 时)。

L——整型变量，输入参数。若 L＝0，表示计算傅里叶变换；若 L＝1，表示计算逆傅里叶变换。

IL——整型变量，输入参数。若 IL＝0，表示不计算傅里叶变换或逆变换的模与幅角；若 IL＝1，表示要计算傅里叶变换或逆变换的模与幅角。

五、子程序(文件名：KKFFT.FOR)

```
PROGRAM KKFFT0
DIMENSION PR(64),PI(64),FR(64),FI(64)
      DOUBLE PRECISION PR,PI,FR,FI
      N=64
      K=6
      DO 10 I=1,N
         PR(I)=EXP(-0.1*(I-0.5))
         PI(I)=0.0
10    CONTINUE
      WRITE(*,*)
      WRITE(*,20) (PR(I),I=1,N)
20    FORMAT(1X,4D15.6)
      WRITE(*,30)
30    FORMAT(2X,'    ')
      CALL KKFFT(PR,PI,N,K,FR,FI,0,1)
      WRITE(*,20) (FR(I),I=1,N)
      WRITE(*,20) (FI(I),I=1,N)
      WRITE(*,20) (PR(I),I=1,N)
      WRITE(*,20) (PI(I),I=1,N)
      CALL KKFFT(FR,FI,N,K,PR,PI,1,1)
      WRITE(*,20) (FR(I),I=1,N)
      WRITE(*,*)
      END
```

序列 $P_i(i=1,2,\cdots,64)$：

.951229D+00	.860708D+00	.778801D+00	.704688D+00
.637628D+00	.576950D+00	.522046D+00	.472367D+00
.427415D+00	.386741D+00	.349938D+00	.316637D+00
.286505D+00	.259240D+00	.234570D+00	.212248D+00
.192050D+00	.173774D+00	.157237D+00	.142274D+00
.128735D+00	.116484D+00	.105399D+00	.953691D−01
.862936D−01	.780817D−01	.706512D−01	.639279D−01
.578443D−01	.523397D−01	.473589D−01	.428521D−01
.387742D−01	.350843D−01	.317456D−01	.287246D−01
.259911D−01	.235177D−01	.212797D−01	.192547D−01
.174224D−01	.157644D−01	.142642D−01	.129088D−01
.116786D−01	.105672D−01	.956160D−02	.865170D−02
.782838D−02	.708341D−02	.640933D−02	.579940D−02
.524752D−02	.474815D−02	.429630D−02	.388746D−02
.351752D−02	.318278D−02	.287990D−02	.260584D−02
.235786D−02	.213348D−02	.193045D−02	.174675D−02

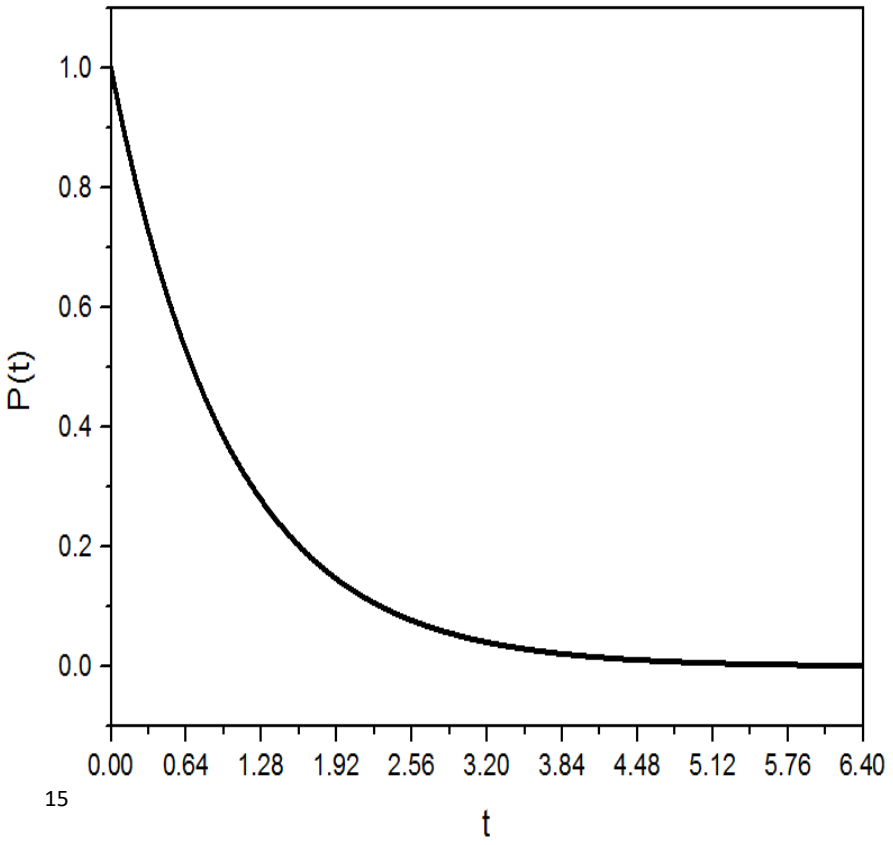
P_i 的傅里叶变换 $F_i(i=1,2,\cdots,64)$的实部：

.997923D+01	.531844D+01	.243865D+01	.146438D+01
.106110D+01	.861244D+00	.748901D+00	.679837D+00
.634482D+00	.603158D+00	.580650D+00	.563957D+00
.551251D+00	.541370D+00	.533549D+00	.527264D+00
.522149D+00	.517944D+00	.514456D+00	.511543D+00
.509098D+00	.507039D+00	.505301D+00	.503836D+00
.502604D+00	.501574D+00	.500723D+00	.500030D+00
.499482D+00	.499067D+00	.498775D+00	.498603D+00
.498546D+00	.498603D+00	.498775D+00	.499067D+00
.499482D+00	.500030D+00	.500723D+00	.501574D+00
.502604D+00	.503836D+00	.505301D+00	.507039D+00
.509098D+00	.511543D+00	.514456D+00	.517944D+00
.522149D+00	.527264D+00	.533549D+00	.541370D+00
.551251D+00	.563957D+00	.580650D+00	.603157D+00
.634482D+00	.679837D+00	.748900D+00	.861244D+00
.106110D+01	.146438D+01	.243865D+01	.531844D+01

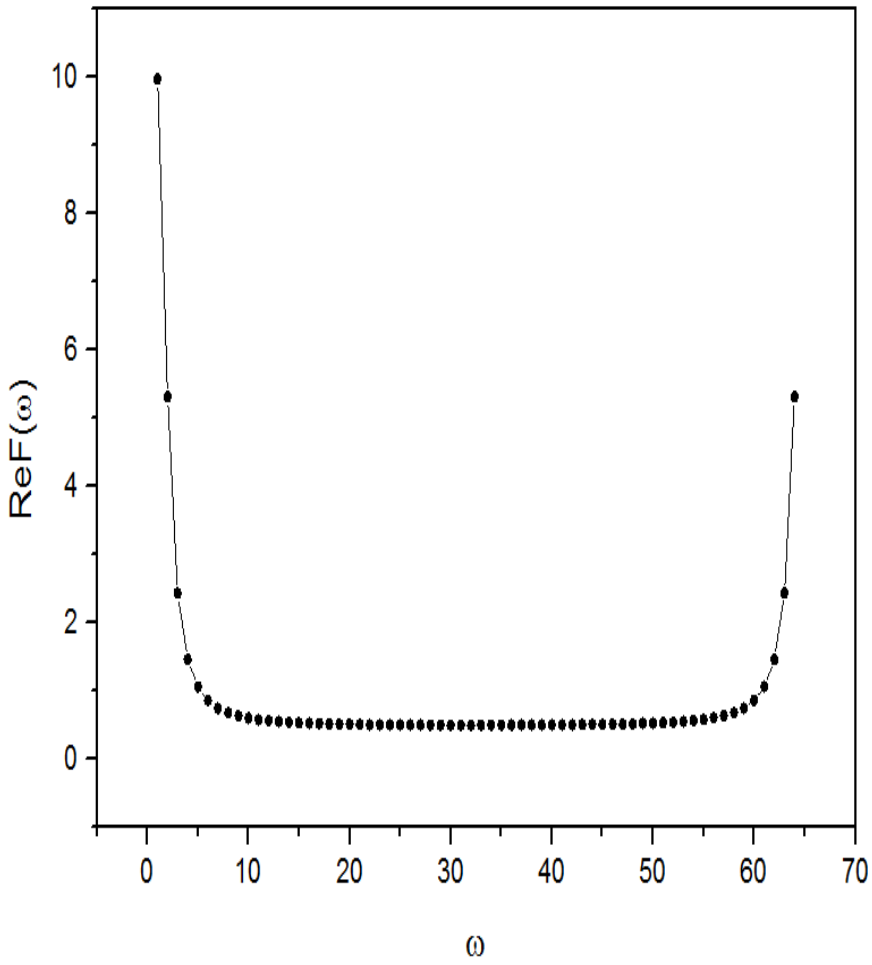
P_i 的傅里叶变换 $F_i(i=1,2,\cdots,64)$的虚部：

.000000D+00	−.473967D+01	−.382485D+01	−.286773D+01
−.223986D+01	−.181854D+01	−.152015D+01	−.129842D+01
−.112707D+01	−.990376D+00	−.878443D+00	−.784767D+00
−.704911D+00	−.635744D+00	−.575000D+00	−.520997D+00
−.472460D+00	−.428403D+00	−.388053D+00	−.350793D+00
−.316123D+00	−.283634D+00	−.252985D+00	−.223890D+00
−.196104D+00	−.169417D+00	−.143644D+00	−.118622D+00
−.942032D−01	−.702537D−01	−.466479D−01	−.232677D−01

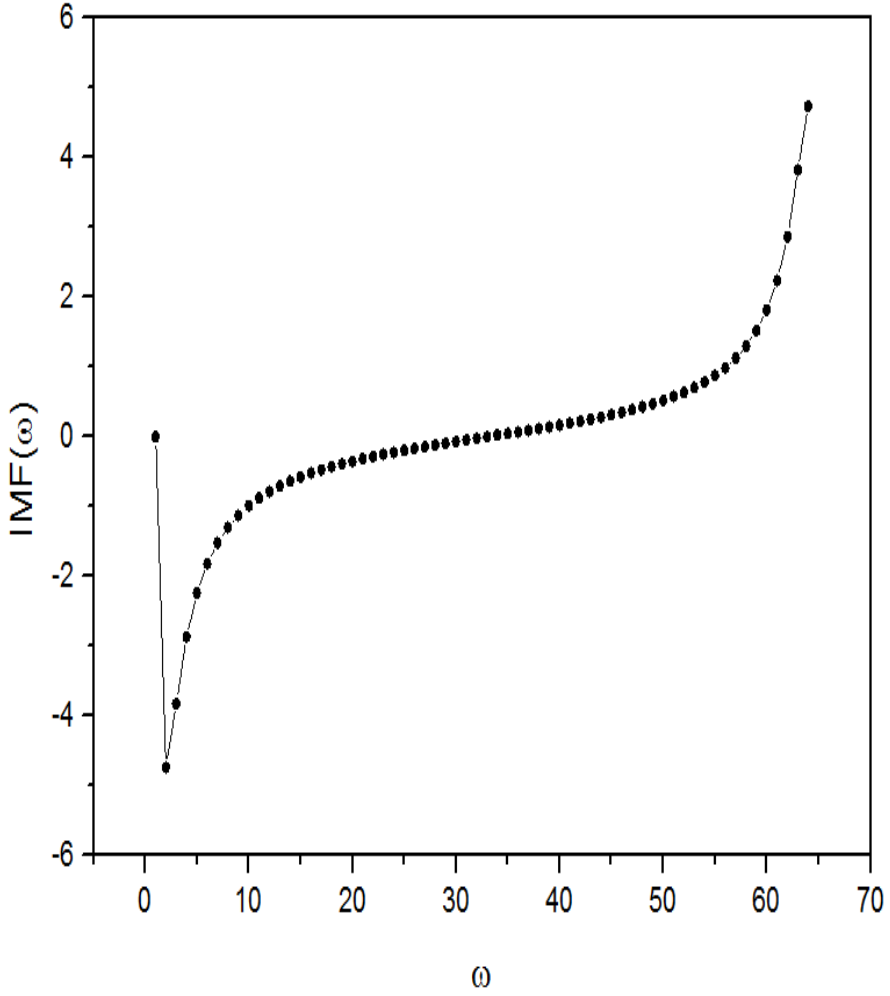
原函数如下



像函数的实部



像函数的虚部

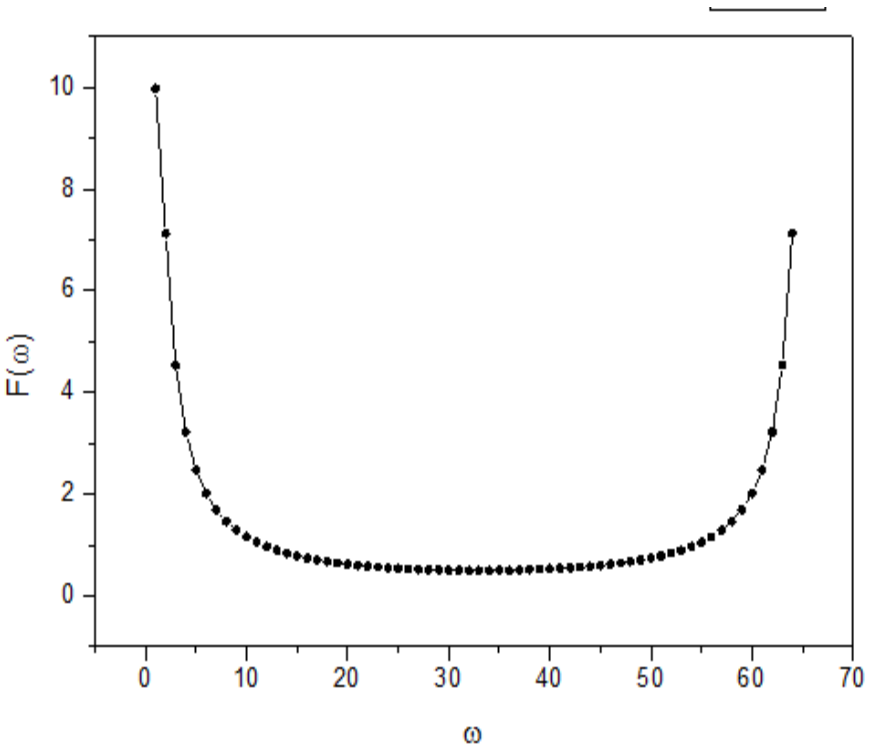


傅里叶变换的像函数的模

.000000D+00	.232682D-01		
.942034D-01	.118622D+00	.466483D-01	
.196104D+00	.223890D+00	.143644D+00	.702540D-01
.316123D+00	.350793D+00	.252985D+00	.169417D+00
.472460D+00	.520997D+00	.388054D+00	.283634D+00
.704911D+00	.784767D+00	.575000D+00	.428404D+00
.112707D+01	.129842D+01	.878443D+00	.635744D+00
.223985D+01	.286773D+01	.152015D+01	.990377D+00
		.382485D+01	.181854D+01
			.473967D+01

P_i的傅里叶变换 $F_i(i=1,2,\cdots,64)$的模：

.997923D+01	.712393D+01	.453613D+01	.321998D+01
.247849D+01	.201217D+01	.169461D+01	.146563D+01
.129339D+01	.115959D+01	.105300D+01	.966389D+00
.894861D+00	.835016D+00	.784410D+00	.741246D+00
.704172D+00	.672157D+00	.644399D+00	.620268D+00
.599262D+00	.580979D+00	.565094D+00	.551341D+00
.539507D+00	.529413D+00	.520919D+00	.513908D+00
.508288D+00	.503987D+00	.500952D+00	.499146D+00
.498546D+00	.499146D+00	.500952D+00	.503987D+00
.508288D+00	.513908D+00	.520919D+00	.529414D+00
.539507D+00	.551341D+00	.565094D+00	.580979D+00
.599262D+00	.620268D+00	.644399D+00	.672157D+00
.704172D+00	.741246D+00	.784410D+00	.835016D+00
.894861D+00	.966389D+00	.105300D+01	.115959D+01
.129339D+01	.146563D+01	.169461D+01	.201217D+01
.247849D+01	.321998D+01	.453613D+01	.712392D+01

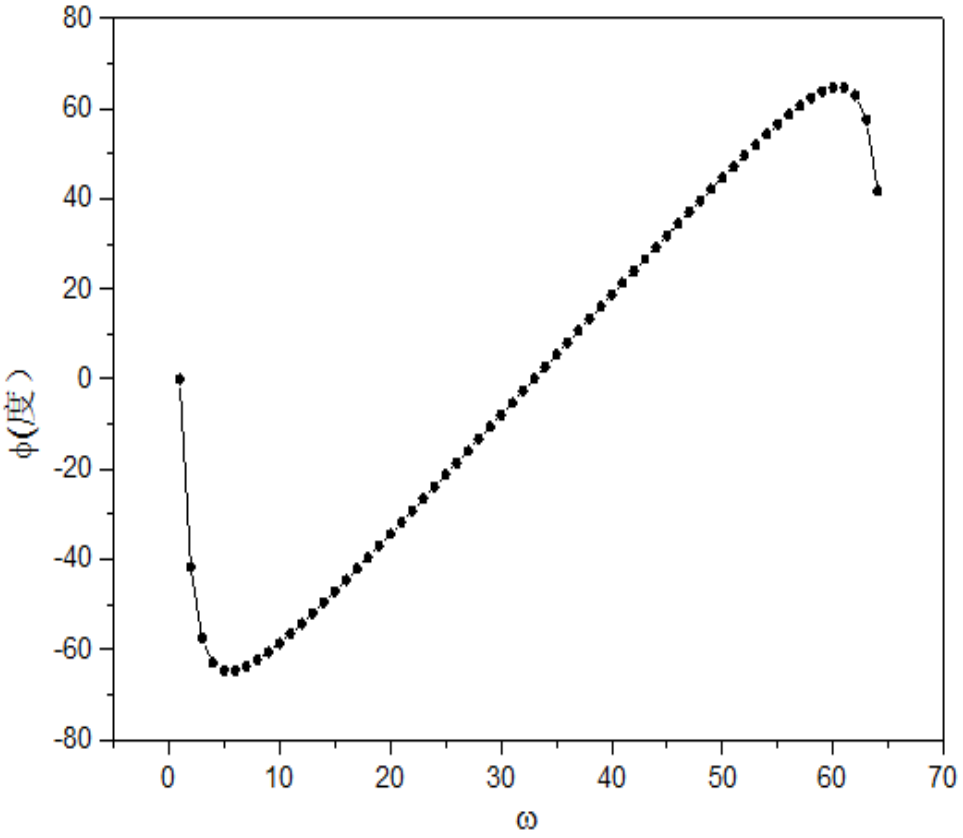


P_i的傅里叶变换 $F_i(i=1,2,\cdots,64)$的幅角(单位为度)：

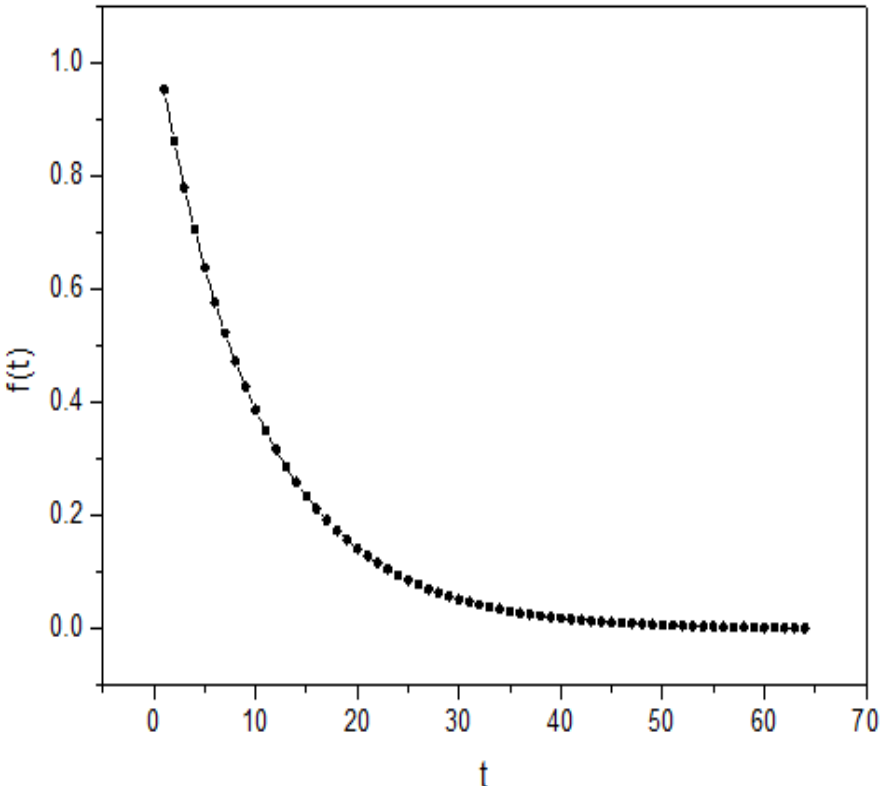
.000000D+00	−.417067D+02	−.574792D+02	−.629494D+02
−.646513D+02	−.646581D+02	−.637729D+02	−.623640D+02
−.606228D+02	−.586578D+02	−.565353D+02	−.542979D+02
−.519741D+02	−.495838D+02	−.471414D+02	−.446575D+02
−.421400D+02	−.395949D+02	−.370272D+02	−.344406D+02
−.318381D+02	−.292223D+02	−.265954D+02	−.239589D+02
−.213145D+02	−.186634D+02	−.160067D+02	−.133455D+02
−.106806D+02	−.801289D+01	−.534304D+01	−.267181D+01
.000000D+00	.267187D+01	.534309D+01	.801292D+01
.106806D+02	.133455D+02	.160068D+02	.186634D+02
.213145D+02	.239589D+02	.265954D+02	.292224D+02
.318381D+02	.344406D+02	.370272D+02	.395950D+02
.421400D+02	.446575D+02	.471414D+02	.495838D+02
.519741D+02	.542979D+02	.565353D+02	.586578D+02
.606228D+02	.623640D+02	.637729D+02	.646581D+02
.646513D+02	.629494D+02	.574792D+02	.417067D+02

的模：

傅里叶变换的像函数的辅角



.951229D+00	.860708D+00	.778801D+00	.704688D+00
.637628D+00	.576950D+00	.522046D+00	.472366D+00
.427415D+00	.386741D+00	.349938D+00	.316637D+00
.286505D+00	.259240D+00	.234570D+00	.212248D+00
.192050D+00	.173774D+00	.157237D+00	.142274D+00
.128735D+00	.116484D+00	.105399D+00	.953692D-01
.862936D-01	.780817D-01	.706512D-01	.639279D-01
.578443D-01	.523397D-01	.473590D-01	.428522D-01
.387742D-01	.350844D-01	.317457D-01	.287247D-01
.259912D-01	.235178D-01	.212798D-01	.192548D-01
.174224D-01	.157645D-01	.142643D-01	.129069D-01
.116786D-01	.105673D-01	.956167D-02	.865179D-02
.782840D-02	.708346D-02	.640938D-02	.579948D-02
.524756D-02	.474822D-02	.429637D-02	.388755D-02
.351755D-02	.318285D-02	.287996D-02	.260593D-02
.235792D-02	.213357D-02	.193053D-02	.174685D-02



逆变换得到的原函数

```
      SUBROUTINE KKFFT(PR,PI,N,K,FR,FI,L,IL)
          DIMENSION PR(N),PI(N),FR(N),FI(N)
          DOUBLE PRECISION PR,PI,FR,FI,P,Q,S,VR,VI,PODDR,PODDI
          DO 20 IT=0,N-1
             M=IT
             IS=0
             DO 10 I=0,K-1
                J=M/2
                IS=2*IS+(M-2*J)
                M=J
   10        CONTINUE
             FR(IT+1)=PR(IS+1)
             FI(IT+1)=PI(IS+1)
   20     CONTINUE
          PR(1)=1.0
```

```
      PI(1)=0.0
      PR(2)=COS(6.283185306/N)
      PI(2)=-SIN(6.283185306/N)
      IF (L.NE.0) PI(2)=-PI(2)
      DO 30 I=3,N
         P=PR(I-1)*PR(2)
         Q=PI(I-1)*PI(2)
         S=(PR(I-1)+PI(I-1))*(PR(2)+PI(2))
         PR(I)=P-Q
         PI(I)=S-P-Q
   30 CONTINUE
      DO 40 IT=0,N-2,2
         VR=FR(IT+1)
         VI=FI(IT+1)
         FR(IT+1)=VR+FR(IT+2)
```

```
      FI(IT+1)=VI+FI(IT+2)
      FR(IT+2)=VR-FR(IT+2)
      FI(IT+2)=VI-FI(IT+2)
40  CONTINUE
    M=N/2
    NV=2
    DO 70 L0=K-2,0,-1
      M=M/2
      NV=2*NV
      DO 60 IT=0,(M-1)*NV,NV
      DO 60 J=0,(NV/2)-1
        P=PR(M*J+1)*FR(IT+J+1+NV/2)
        Q=PI(M*J+1)*FI(IT+J+1+NV/2)
        S=PR(M*J+1)+PI(M*J+1)
        S=S*(FR(IT+J+1+NV/2)+FI(IT+J+1+NV/2))
```

```
            PODDR=P-Q
            PODDI=S-P-Q
            FR(IT+J+1+NV/2)=FR(IT+J+1)-PODDR
            FI(IT+J+1+NV/2)=FI(IT+J+1)-PODDI
            FR(IT+J+1)=FR(IT+J+1)+PODDR
            FI(IT+J+1)=FI(IT+J+1)+PODDI
60      CONTINUE
70  CONTINUE

      IF (L.NE.0) THEN
         DO 80 I=1,N
            FR(I)=FR(I)/N
            FI(I)=FI(I)/N
80       CONTINUE
      END IF
```

```
      IF (IL.NE.0) THEN
         DO 90 I=1,N
            PR(I)=SQRT(FR(I)*FR(I)+FI(I)*FI(I))
            PI(I)=ATAN(FI(I)/FR(I))*360.0/6.283185306
90       CONTINUE
      END IF
      RETURN
      END
```