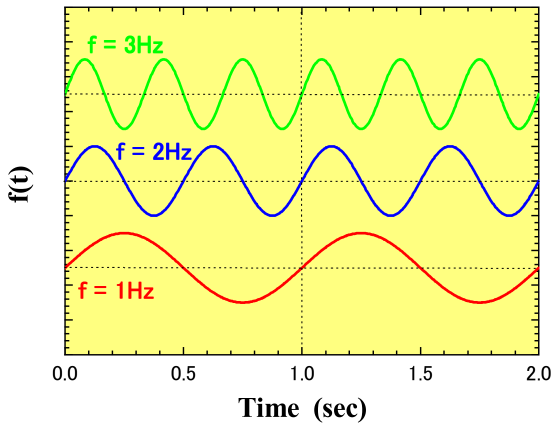
３．フーリエ変換

3.1　周期的な複雑な波形

$f(t) = \sin(\omega t)$, $f(t) = \sin(2\omega t)$, $f(t) = \sin(3\omega t)$ のグラフを図３・１に示す。単純にこれらの波形を重ね合わせると、$f(t) = \sin(\omega t) + \sin(2\omega t) + \sin(3\omega t)$ は右図のように複雑な波形となる。この合成波の時間方向の移動は見られない（時間方向を波の位相と呼ぶ)。しかし、振幅の変調が見られる。



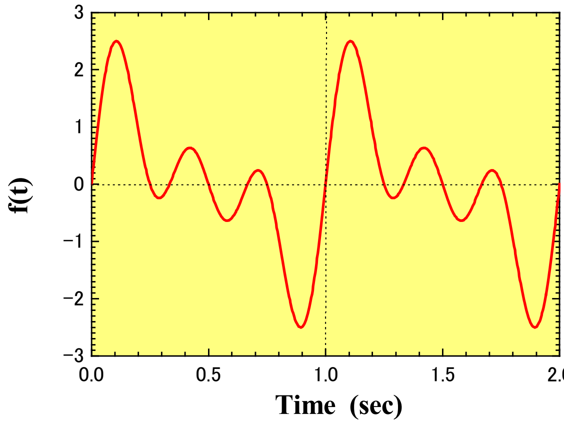


図３・１　周期関数

一方、$f(t) = \cos(\omega t) + \sin(\omega t)$ は、図３・２のようになる。振幅の変調はないが、位相のズレが見られる。例えば、

$$f(t) = A\cdot\cos(\omega t) + B\cdot\sin(\omega t)$$

$$= \sqrt{A^2+B^2}\left\{\frac{A}{\sqrt{A^2+B^2}}\cdot\cos(\omega t) + \frac{B}{\sqrt{A^2+B^2}}\cdot\sin(\omega t)\right\}$$

$$= \sqrt{A^2+B^2}\left\{\sin(\phi)\cdot\cos(\omega t) + \cos(\phi)\cdot\sin(\omega t)\right\}$$

$$= \sqrt{A^2+B^2}\sin(\omega t + \phi) \qquad (3\cdot1)$$

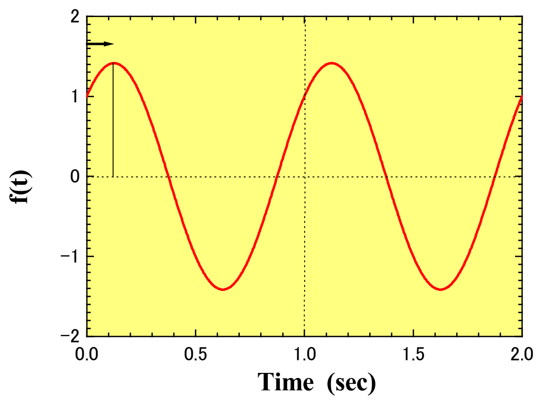


図３・２　$\sin(\omega t) + \cos(\omega t)$

となるので、位相 ϕ だけずれる。

つまり、sin と cos の適当な組み合わせで任意の周期関数を表すことができる。この考えに基づいて式に表すと、

$$f(t) = a_0 + a_1\cos(\omega t) + a_2\cos(2\omega t) + a_3\cos(3\omega t) + \cdots$$
$$+ b_1\sin(\omega t) + b_2\sin(2\omega t) + b_3\sin(3\omega t) + \cdots \qquad (3\cdot2)$$

となる（フーリエ級数)。

3.2　波形合成の意味

周期的な波形は振幅（ a_n, b_n ）と位相（ $n\omega = 2\pi nf$ ）で特徴付けられることがわかったが、実際にはどういう意味なのか？例えば、色は 3 原色（赤(R)・緑(G)・青(B)）の混ぜ方ですべての色を表すことができる。色は周波数（波長）で決まり、色の明るさ（強度）は振幅で決まる。つまり、a_n, b_n, ω_n の組み合わせであらゆる色が作り出される。つまり、料理では砂糖、塩、醤油を何グラムずつ配合するかによっていろいろな味ができるのと同じである。音は、基本振動数、2 倍振動数、3 倍振動数、・・・の割合で波形が決まり、いろいろな音の「音色」が生み出される。

3.3　波の構成要素

（3・2）式のように、簡単な波の重ね合わせで複雑な波形を作れることがわかったが、ここでは逆に複雑な波形から、基本的な波の構成要素（$\omega_0, 2\omega_0$, ・・・ に対応する a_0, a_1, b_1, a_2, b_2, ・・・）を求める。ある音の複雑な繰り返しパターンの基本振動数を f_0（ $\omega_0 = 2\pi f_0$ ）とする。図 3・1 の場合は、周期 $T = 1$ (sec) なので、周波数 $f_0 = 1$ (Hz) となる。

まず、sin, cos の積分を計算してみる。

$$\int_0^T \cos(\omega_0 t)dt = \left|\frac{\sin(\omega_0 t)}{\omega_0}\right|_0^T = 0 \qquad (\ \omega_0 T = 2\pi\ )$$

$$\int_0^T \sin(\omega_0 t)dt = \left|-\frac{\cos(\omega_0 t)}{\omega_0}\right|_0^T = 0$$

となるので、$\int_0^T \cos(2\omega_0 t)dt = 0$, $\int_0^T \cos(3\omega_0 t)dt = 0$, ・・・, $\int_0^T \sin(2\omega_0 t)dt = 0$, $\int_0^T \sin(3\omega_0 t)dt = 0$, ・・・

が求まる。結局、（3・2）式の積分は、$\int_0^T f(t)dt = a_0 T$ になり、

$$a_0 = \frac{1}{T}\int_0^T f(t)dt \qquad (3\cdot 3)$$

ここで、複雑な波形の 1 周期分を積分すれば、a_0 が求まることがわかった。次に、a_1, b_1, a_2, b_2, ・・・を計算する。その前に $\int_0^T \{a_1 \cdot \cos(\omega_0 t) \times a_2 \cdot \cos(2\omega_0 t)\}dt$ の計算をしてみる。

① cos どうしの掛け算の積分（$f_1 \neq f_2$）

$\cos A \cdot \cos B = \dfrac{\cos(A+B) + \cos(A-B)}{2}$ 　の公式を用いて、

$$\int_0^T \{a_1 \cdot \cos(\omega_0 t) \times a_2 \cdot \cos(2\omega_0 t)\}dt = \frac{a_1 a_2}{2}\left[\int_0^T \cos(3\omega_0 t)dt + \int_0^T \cos(-\omega_0 t)dt\right] = 0$$

② cos どうしの掛け算の積分（$f_1 = f_2$）

$\cos^2 A = \dfrac{\cos(2A)+1}{2}$ から

$$\int_0^T a_1 \cdot \cos(\omega_0 t) \times \cos(\omega_0 t) dt = \frac{a_1}{2}\left[\int_0^T \cos(2\omega_0 t) dt + \int_0^T dt\right] = \frac{a_1 T}{2}$$

となる。

③ sin どうしの掛け算の積分（$f_1 \neq f_2$）

同様に、$\sin A \cdot \sin B = -\dfrac{\cos(A+B)-\cos(A-B)}{2}$ を利用して、

$$\int_0^T b_1 \cdot \sin(\omega_0 t) \times b_2 \cdot \sin(2\omega_0 t) dt = -\frac{b_1 b_2}{2}\left[\int_0^T \cos(3\omega_0 t) dt - \int_0^T \cos(-\omega_0 t) dt\right] = 0$$

④ sin どうしの掛け算の積分（$f_1 = f_2$）

$\sin^2 A = \dfrac{1-\cos(2A)}{2}$ から

$$\int_0^T b_1 \cdot \sin(\omega_0 t) \times \sin(\omega_0 t) dt = \frac{b_1}{2}\left[\int_0^T dt - \int_0^T \cos(2\omega_0 t) dt\right] = \frac{b_1 T}{2}$$

となる。

⑤ sin×cos の掛け算の積分（$f_1 \neq f_2$）

$$\sin A \cdot \cos B = \frac{\sin(A+B)+\sin(A-B)}{2}$$

$$\int_0^T b_1 \cdot \sin(\omega_0 t) \times a_2 \cdot \cos(2\omega_0 t) dt = \frac{b_1 a_2}{2}\left[\int_0^T \sin(3\omega_0 t) dt + \int_0^T \sin(-\omega_0 t) dt\right] = 0$$

⑥ sin×cos の掛け算の積分（$f_1 = f_2$）

$$\sin A \cdot \cos A = \frac{\sin(2A)}{2}$$

$$\int_0^T b_1 \cdot \sin(\omega_0 t) \times a_1 \cdot \cos(\omega_0 t) dt = \frac{b_1 a_1}{2}\int_0^T \sin(2\omega_0 t) dt = 0$$

まとめると、cos どうしの掛け算の積分（$f_1 = f_2$）と sin どうしの掛け算の積分（$f_1 = f_2$）の場合だけ、積分値が 0 にならない。

	$b_1 \cdot \sin\omega_0 t$	$b_2 \cdot \sin 2\omega_0 t$	$a_1 \cdot \cos\omega_0 t$	$a_2 \cdot \cos 2\omega_0 t$
$\sin\omega_0 t$	$b_1 T / 2$	0	0	0
$\cos\omega_0 t$	0	0	$a_1 T / 2$	0

これらのことを考慮して、(3・2) 式の両辺に cos2 ω_0t をかけて積分すると、

$$\int_0^T f(t)\cos(2\omega_0 t)dt = \int_0^T a_0\cos(2\omega_0 t)dt + \int_0^T a_1\cos(\omega_0 t)\cos(2\omega_0 t)dt + \int_0^T a_2 \cdot \boxed{\cos(2\omega_0 t)\cos(2\omega_0 t)d}t + \cdots$$

$$+ \int_0^T b_1\sin(\omega_0 t)\cos(2\omega_0 t)dt + \int_0^T b_2\sin(2\omega_0 t)\cos(2\omega_0 t)dt + \cdots$$

$$= \frac{a_2 T}{2}$$

$$\boxed{a_2 = \frac{2}{T}\int_0^T f(t)\cos(2\omega_0 t)dt}$$

一般に、次のような関係が求められる。

$$a_n = \frac{2}{T}\int_0^T f(t)\cos(n\omega_0 t)dt \qquad (3・4)$$

$$b_n = \frac{2}{T}\int_0^T f(t)\sin(n\omega_0 t)dt \qquad (3・5)$$

結局、(3・3) 式、(3・4) 式、(3・5) 式によって、フーリエ級数の係数の求め方がわかった。

3.4　数値積分

測定データに対して、積分をしないとフーリエ係数が求まらないので、まずは、数値積分のプログラムを作る。実際の測定データは離散データになるので、積分から和になる。

$$\int f(x)dx \rightarrow \sum_n f(x_n)\times\Delta x \qquad (3・6)$$

Δ x が小さくなれば連続関数の積分値に近づいていく。

実際に、 sin θ ($0 \leqq \theta \leqq \pi/2$) の離散データ数値積分を行ってみる。図 3・3 のように 2 通りの近似の仕方がある。さらに近似度を上げるには、棒状近似から台形近似（シンプソンの台形近似）にする。

$$\sum_n f(x_n)\times\Delta x \rightarrow \sum_n \frac{f(x_{n-1}) + f(x_n)}{2}\times\Delta x \qquad (3・7)$$

3 つの場合の計算プログラムは次のようになる。変数 Sum0, Sum1, Sum2 に計算結果が代入される。シンプソンの台形公式 (Sum2) が真の値に近いことがわかる。また、dX を小さくしていくと、Sum0, Sum1, Sum2 の差が小さくなっていく。離散データの 3 点を使って 2 次関数近似計算もあるが、少し複雑になるので、ここでは述べない。

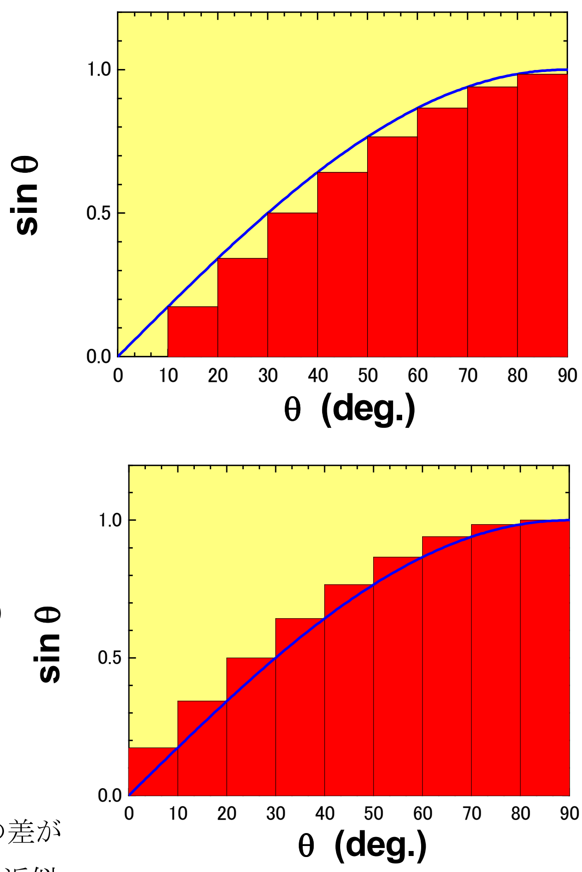


図　3・3　sin θ の数値積分

```
Dim pi As Double
Sub integ()
  pi = 4# * Atn(1#)      ' π 円周率
  X_max = 90             ' 積分範囲（°）
  dX = 10                '
  X0 = 0#: X1 = dX
  I = 0:    Sum0 = 0#: Sum1 = 0#: Sum2 = 0#
  Do
    I = I + 1
    Y0 = Sin(X0 * pi / 180#)
    Y1 = Sin(X1 * pi / 180#)
    Sum0 = Sum0 + dX * Y0
    Sum1 = Sum1 + dX * Y1
    Sum2 = Sum2 + dX * (Y0 + Y1) / 2#
    Cells(I, 1) = X0
    Cells(I, 2) = Y0
    Cells(I, 3) = X1
    Cells(I, 4) = Y1
    X0 = X0 + dX: X1 = X1 + dX
  Loop While X1 <= X_max
  Cells(I + 2, 1) = X_max / dX
  Cells(I + 2, 2) = Sum0 * pi / 180#
  Cells(I + 2, 3) = Sum1 * pi / 180#
  Cells(I + 2, 4) = Sum2 * pi / 180#
End Sub
```

3.5　離散フーリエ変換（DFT: Discrete Fourier Transform）

（3・2）式を書き直すと、

$$f(t) = a_0 + \sum_{n=1}^{\infty}\{a_n \cos(n\omega t) + b_n \sin(n\omega t)\} \qquad (3・8)$$

オイラーの公式より、$e^{i\theta} = \cos\theta + i\sin\theta$、$\cos\theta = \dfrac{e^{i\theta} + e^{-i\theta}}{2}$、$\sin\theta = \dfrac{e^{i\theta} - e^{-i\theta}}{2i}$ を（3・8）式に代入して、

$$f(t) = a_0 + \sum_{n=1}^{\infty}\left\{\frac{a_n\left(e^{in\omega t} + e^{-in\omega t}\right)}{2} + \frac{b_n\left(e^{in\omega t} - e^{-in\omega t}\right)}{2i}\right\}$$

$$f(t) = a_0 + \sum_{n=1}^{\infty}\left\{\frac{(a_n - ib_n)e^{in\omega t}}{2} + \frac{(a_n + ib_n)e^{-in\omega t}}{2}\right\}$$

$c_n = \dfrac{a_n - ib_n}{2}$、$c_{-n} = \dfrac{a_n + ib_n}{2}$ とすると、

$$f(t) = c_0 + \sum_{n=1}^{\infty}\left\{c_n e^{in\omega t} + c_{-n} e^{-in\omega t}\right\} = c_0 + \sum_{n=1}^{\infty} c_n e^{in\omega t} + \sum_{n=-1}^{-\infty} c_n e^{in\omega t}$$

$$\boxed{f(t) = \sum_{n=-\infty}^{\infty} c_n e^{in\omega t}} \qquad (3・9)$$

また、$c_n = \dfrac{a_n - ib_n}{2}$、(3・4) 式と (3・5) 式より、

$$
\begin{aligned}
c_n &= \frac{1}{2}\frac{2}{T}\int_0^T f(t)\{\cos(n\omega t) - i\cdot\sin(n\omega t)\}dt \\
&= \frac{1}{T}\int_0^T f(t)e^{-in\omega t}dt
\end{aligned}
\tag{3・10}
$$

プログラムで、f = 100 Hz の複雑な波形を作り、フーリエ変換してみる。(3・8) 式の n が 4 までの級数でそれぞれの係数の値を以下のようにすると、図 3.4 のようになる。

$a_0 = 0.5$

$a_1 = 0.5$　　$b_1 = 0.8$

$a_2 = 0.2$　　$b_2 = -0.4$

$a_3 = -0.7$　　$b_3 = 0.1$

$a_4 = -1.2$　　$b_4 = 0.3$

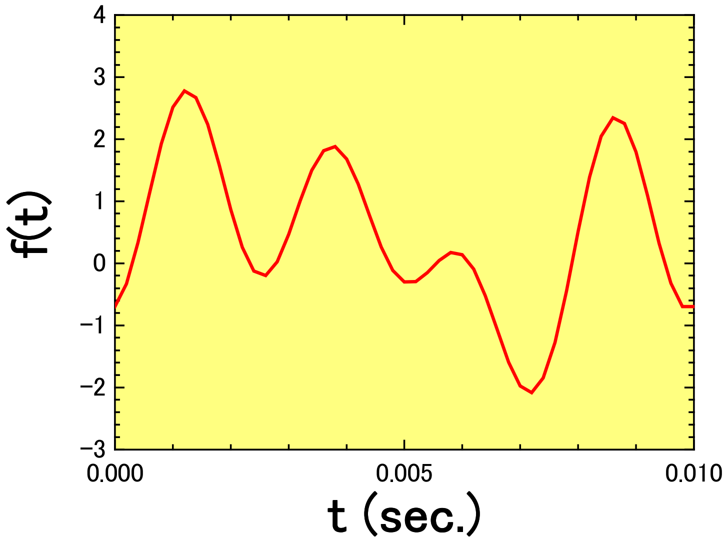


図 3・4　複雑な波形 (100Hz)

$$
\begin{aligned}
f(t) = 0.5 &+ 0.5\cos(\omega t) + 0.2\cos(2\omega t) - 0.7\cos(3\omega t) - 1.2\cos(4\omega t) \\
&+ 0.8\sin(\omega t) - 0.4\sin(2\omega t) + 0.1\sin(3\omega t) + 0.3\sin(4\omega t)
\end{aligned}
$$

計算結果は次のようになる。

初期値

N	a_n	b_n	c_n
0	0.5		0.5
1	0.5	0.8	0.471699
2	0.2	−0.4	0.223607
3	−0.7	0.1	0.353553
4	−1.2	0.3	0.618466

計算値

N	a_n	b_n	c_n
0	0.486	0	0.486
1	0.236	−0.4	0.464431
2	0.086	0.2	0.217706
3	−0.364	−0.05	0.367418
4	−0.614	−0.15	0.632057

```
Dim pi As Double, nD As Integer, nDFT As Integer, f1, T1, X(500) As Double, Y(500) As Double
Dim a(2, 10) As Single

Sub init_wave()
   pi = 4# * Atn(1#)      ' π 円周率
   f1 = 100                 ' Frequency (Hz)
   T1 = 1# / f1             ' 周期 (sec)
   dt = T1 / 50             ' Δt
   t_max = T1               ' 積分範囲 (1 周期) 0～t～T1
   nDFT = 4                 '
   a0 = 0.5                 ' 初期値
   a(1, 1) = 0.5: a(1, 2) = 0.2: a(1, 3) = -0.7: a(1, 4) = -1.2
   a(2, 1) = 0.8: a(2, 2) = -0.4: a(2, 3) = 0.1: a(2, 4) = 0.3
'------------------------------------------------------------
   t = 0#: I = 0
   Do
      I = I + 1
      X(I) = t
      Y(I) = a0
      For K = 1 To nDFT
         Y(I) = Y(I) + a(1, K) * Cos(K * 2# * pi * f1 * t) + a(2, K) * Sin(K * 2# * pi * f1 * t)
      Next K
      Cells(I, 1) = X(I)
      Cells(I, 2) = Y(I)
      t = t + dt
   Loop While t < t_max + dt
   nD = I
   Cells(1, 4) = 0
   Cells(1, 7) = a0
   For I = 1 To nDFT
      Cells(I + 1, 4) = I
      Cells(I + 1, 5) = a(1, I)
      Cells(I + 1, 6) = a(2, I)
      Cells(I + 1, 7) = Sqr(a(1, I) ^ 2 + a(2, I) ^ 2) / 2#
   Next I
'****************
   DFT
'****************
End Sub

Sub DFT()
   dt = X(2) - X(1)
   f = f1
   For N = 0 To nDFT
      c = 0#: s = 0#
      For I = 1 To nD
         c = c + Y(I) * dt * Cos(-N * 2# * pi * f * X(I))
         s = s + Y(I) * dt * Sin(-N * 2# * pi * f * X(I))
      Next I
      c = c / T1
      s = s / T1
      Cells(N + 8, 4) = N
      Cells(N + 8, 5) = c
      Cells(N + 8, 6) = s
      Cells(N + 8, 7) = Sqr(c ^ 2 + s ^ 2)
   Next N
End Sub
```

3.6　周期のわからないフーリエ変換

　これまでは周期のある波形のフーリエ変換であったが、実際の波形は周期がはっきりしていないものが多い（図 3・5)。それならば周期を無限大と考えれば、周期があいまいな複雑な波形の周期を決める必要がなくなる。

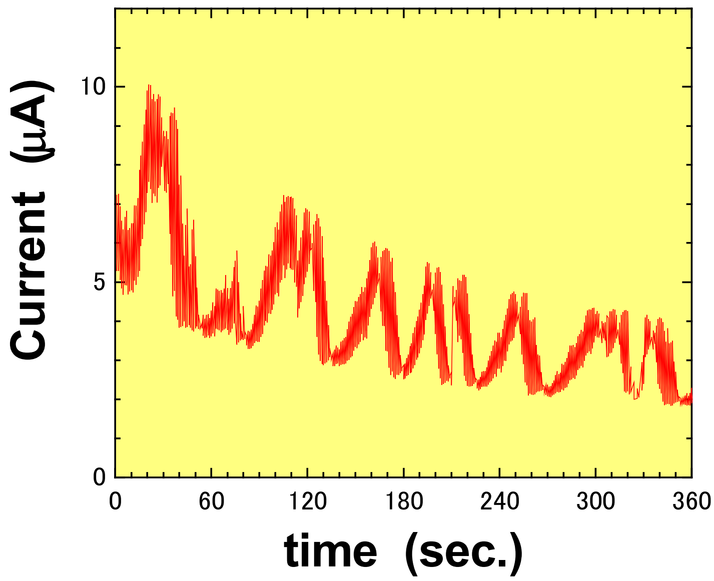


図 3・5　実際の波形

　(3・10）式は 1 周期分の積分だから次式のように書き換えられる。

$$c_n = \frac{1}{T}\int_{-T/2}^{T/2} f(t) e^{-in\omega t} dt \qquad (3・11)$$

この式を（3・9）式に代入すると、

$$f(t) = \sum_{n=-\infty}^{\infty} \left\{ \frac{1}{T}\int_{-T/2}^{T/2} f(t) e^{-in\omega t} dt \right\} e^{in\omega t}$$

T→∞とすると、1 / T → Δ f となる。 f の連続関数（nω→ω）になるので和（ΣΔ f ）は積分（∫df）になる。

$$f(t) = \lim_{T\to\infty} \sum_{n=-\infty}^{\infty} \left\{ \Delta f \int_{-T/2}^{T/2} f(t) e^{-in\omega t} dt \right\} e^{in\omega t} = \int_{-\infty}^{\infty} \left\{ \int_{-\infty}^{\infty} f(t) e^{-2\pi i f t} dt \right\} e^{2\pi i f t} df$$

$$G(f) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i f t} dt \qquad (3・12)$$

とすると、

$$f(t) = \int_{-\infty}^{\infty} \{G(f)\} e^{2\pi i f t} df \qquad (3・13)$$

　(3・12）式をフーリエ変換、(3・13）式を逆フーリエ変換と呼ぶ。実際に簡単な関数（図 3・6）のフーリエ変換を計算してみる。

f(t)=1　　　(-Δt/2 ≦ t ≦ Δt/2)

f(t)=0　　　(t < -Δt/2,　Δt/2 < t)

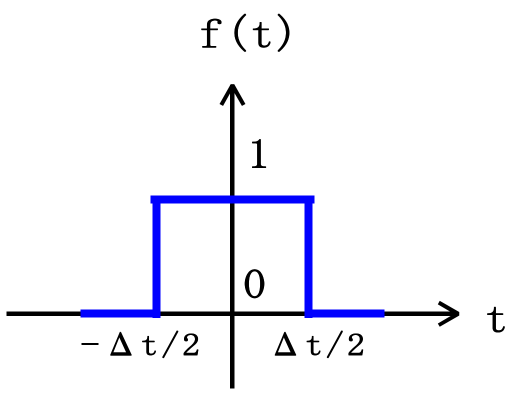


図 3・6

この f(t)を（3・12）式に代入すると、(t < -Δt/2,　Δt/2 < t）の範囲で 0 になるので、(-Δt/2 ≦ t ≦ Δt/2)の範囲だけを考えればよい。

$$G(f) = \int_{-\Delta t/2}^{\Delta t/2} 1 \cdot e^{-2\pi i f t} dt = \left| -\frac{1}{2\pi i f} e^{-2\pi i f t} \right|_{-\Delta t/2}^{\Delta t/2} = \frac{1}{2\pi i f} \left\{ e^{\pi i f \Delta t} - e^{-\pi i f \Delta t} \right\}$$

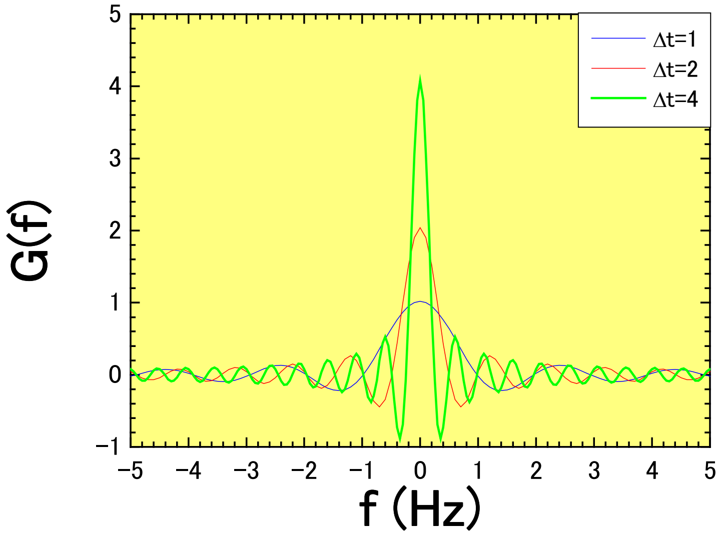
$\sin\theta = \dfrac{e^{i\theta} - e^{-i\theta}}{2i}$ なので、結局、

$$G(f) = \frac{\sin \pi f \Delta t}{\pi f} = \frac{\sin \pi f \Delta t}{\pi f \Delta t} \cdot \Delta t$$

となる。また、$\lim_{\theta \to 0} \dfrac{\sin\theta}{\theta} = 1$ の関係を使うと、$f \to 0$ のとき、$G(f) \to 1 \cdot \Delta t = \Delta t$ となる。実は、$\Delta t \to \infty$ にしたとき、$G(f) \to \infty$ となるデルタ関数である。つまり、一定関数のフーリエ変換はデルタ関数となる。

プログラムを作って確かめてみる。

図 3・7



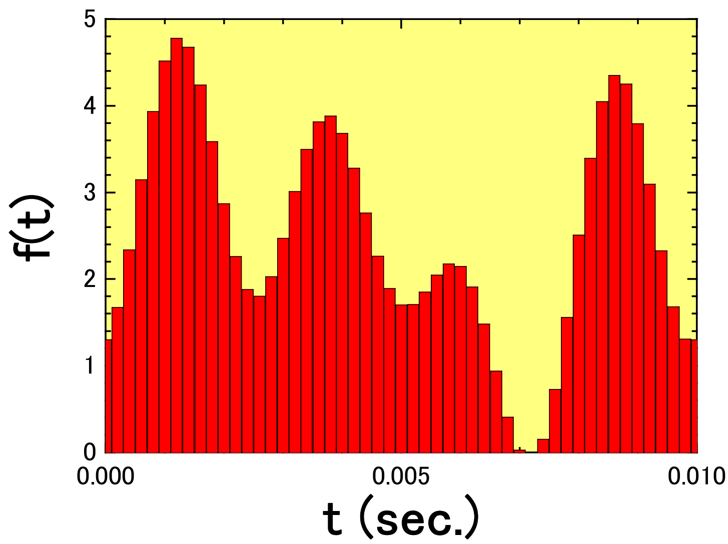

```
Dim pi As Double
Sub init_wave()
   pi = 4# * Atn(1#)       ' π 円周率
'-------------------------------
   dt = 2                       ' Δt (sec)
'-------------------------------
   div = dt / 50             '
   f_max = 5                   ' f (Hz)
   df = f_max / 50
   f = -f_max
   I = 0
   Do
      I = I + 1
      t = -dt / 2
      c = 0: s = 0
      Do
         c = c + div * Cos(-2# * pi * f * t)
         s = s + div * Sin(-2# * pi * f * t)
         t = t + div
      Loop While t < dt / 2 + div
      Cells(I, 1) = f
      Cells(I, 2) = c
      f = f + df
   Loop While f < f_max + df
End Sub
```

3.7　周期がわからない離散フーリエ変換（DFT: Discrete Fourier Transform）

（3・12）式の積分でフーリエ変換の計算が行われるが、プログラムを使っての数値計算では図 3・8 のような短冊の和で計算する。短冊の幅を τ とすると、$t = k\tau$ となる（k は整数）。

$$G(f) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i f t} dt \to \sum_{k=0}^{N-1} f(k\tau) e^{-2\pi i f k \tau} \tau$$

短冊が合計 N 本あるとすると、$T = N\tau$ となる。基本周波数は 1/T なので、n 倍周波数は



$$f_n = \frac{n}{N\tau}$$

$$G(f_n) = G\left(\frac{n}{N\tau}\right) = \sum_{k=0}^{N-1} \tau \cdot f(k\tau) e^{-2\pi ikn/N} \qquad (3・14)$$

実際に図 3・4 の基本周波数(f_1=100Hz)の n=4 の計算をプログラムで計算すると、図 3・9 のようになる。G(f)は、f_1, f_2, f_3, f_4 で値を持ち、f(t)の c_1, c_2, c_3, c_4 の値に比例している。

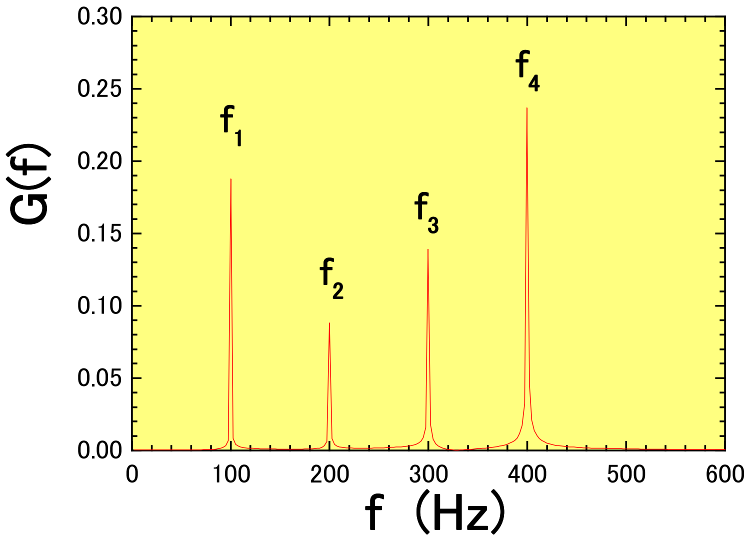


図 3・9

```
Dim pi As Double, nD As Integer, nDFT As Integer
Dim Y(5000) As Double
Dim a(2, 10) As Single, tau As Double
Sub init_wave()
   pi = 4# * Atn(1#)      ' π  円周率
   f1 = 100                ' Frequency (Hz)
   T1 = 1# / f1           ' 周期 (sec)
   dt = T1 / 50           ' Δt
   t_max = T1 * 40        ' 積分範囲
   nDFT = 4                 '
'------------------------------------------------------------
   a0 = 0.5               ' 初期値
   a(1, 1) = 0.5: a(1, 2) = 0.2: a(1, 3) = -0.7: a(1, 4) = -1.2
   a(2, 1) = 0.8: a(2, 2) = -0.4: a(2, 3) = 0.1: a(2, 4) = 0.3
'------------------------------------------------------------
   t = 0#: I = 0
   Do
      I = I + 1
      X(I) = t
      Y(I) = a0
      For K = 1 To nDFT
         Y(I) = Y(I) + a(1, K) * Cos(K * 2# * pi * f1 * t) + a(2, K) * Sin(K * 2# * pi * f1 * t)
      Next K
      Cells(I, 1) = X(I)
      Cells(I, 2) = Y(I)
      t = t + dt
   Loop While t < t_max + dt
   nD = I
   tau = t_max / (nD - 1)
   Cells(1, 3) = nD
'****************
   DFT
'****************
End Sub
```

```
Sub DFT()
   For N = 1 To 300
      c = 0#: s = 0#
      f = N / nD / tau
      For K = 0 To nD - 1
         c = c + Y(K) * tau * Cos(-2# * pi * K * N / nD)
         s = s + Y(K) * tau * Sin(-2# * pi * K * N / nD)
      Next K
      Cells(N, 4) = f
      Cells(N, 5) = c
      Cells(N, 6) = s
      Cells(N, 7) = Sqr(c ^ 2 + s ^ 2)
   Next N
End Sub
```

3.8　ＦＦＴ（FFT: Fast Fourier Transform）

まず、最低何個の点で波を表現できるか考えてみる。ω=1 では最低２個のデータ店が必要になる。

ω=1	2 個
ω=2	4 個
ω=3	6 個
ω=4	8 個
ω=5	10 個

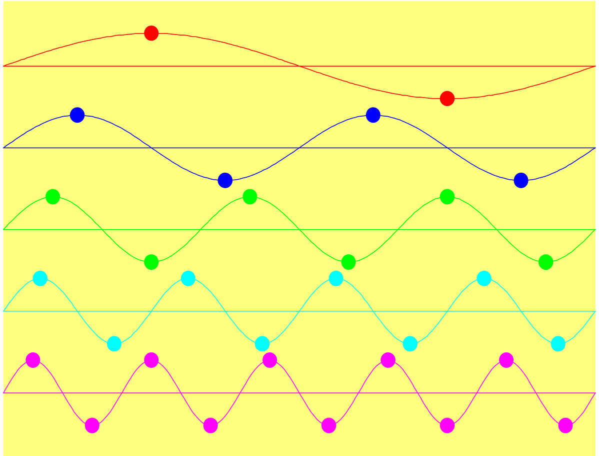


図 3・10　　データ点と波数

つまり、データ点がＮ個の場合、N/2 個の波数の波まで作れる。これをサンプリング理論と呼ぶ。サンプリング理論を考慮して周期のわからないデータを（3・14）式のＤＦＴで計算すると、莫大な計算量になる。例えば、周波数 f=4000Hz までの音のデータをとる。サンプリング理論から 1 秒間で N=8000 点のデータが必要になる。100m秒観測するならば 800 点のデータを取らなければならない（τ＝0.125m秒)。ひとつの周波数で 800 回 $\sum_{k=0}^{N-1} \tau \cdot f(k\tau) e^{-2\pi ikn/N}$ の和の計算を行う。また、図 3・1 より、800 個のデータに対して 400 個のωが作れる。結局、800×400＝320,000 回の計算をしないといけない。

そこで、計算回数を減らすためにＦＦＴが考え出された。ＤＦＴの計算では、もとの f(t)に cos(1ωt), cos(2ωt), cos(3ωt), cos(4ωt), cos(5ωt), ・・・をかけるが、図 3・11 を見るとある時間 t で同じ値をかけていることがわかる。この同じ値の計算を省略すれば計算回数を減らせることができる。(3・14）式のτ＝1 秒として、$W = e^{-2\pi i/N}$ を導入すると、

$$f_n = \frac{n}{N}$$

$$G\left(\frac{n}{N}\right) = \sum_{k=0}^{N-1} f(k) W^{kn} \qquad (3 \cdot 15)$$

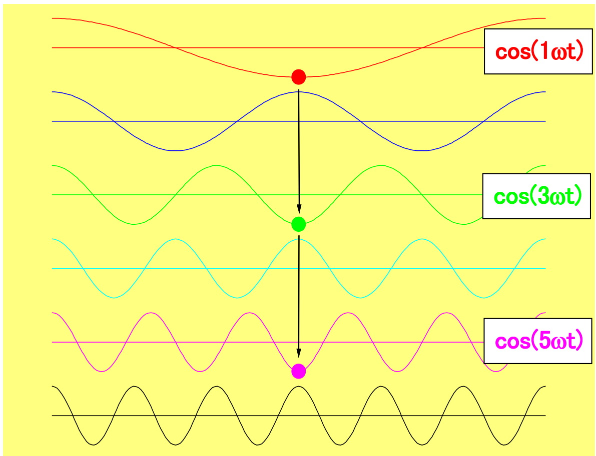


図 3.11

つぎに N=8 の場合の計算をすると、$W = e^{-2\pi i/8}$ となるので、$W^8=W^0$, $W^9=W^1$ ・・・の関係が求まる。

$$
\begin{aligned}
&n=0 \text{ のとき、} G(0) = f(0)W^0 + f(1)W^0 + f(2)W^0 + f(3)W^0 + f(4)W^0 + f(5)W^0 + f(6)W^0 + f(7)W^0 \\
&n=1 \text{ のとき、} G(1/8)=f(0)W^0 + f(1)W^1 + f(2)W^2 + f(3)W^3 + f(4)W^4 + f(5)W^5 + f(6)W^6 + f(7)W^7 \\
&n=2 \text{ のとき、} G(2/8)=f(0)W^0 + f(1)W^2 + f(2)W^4 + f(3)W^6 + f(4)W^8 + f(5)W^{10} + f(6)W^{12} + f(7)W^{14} \\
&n=3 \text{ のとき、} G(3/8)=f(0)W^0 + f(1)W^3 + f(2)W^6 + f(3)W^9 + f(4)W^{12} + f(5)W^{15} + f(6)W^{18} + f(7)W^{21} \qquad (3.16)
\end{aligned}
$$

・・・

$W^8=W^0$, $W^9=W^1$ ・・・の関係を利用すると、結局、

	f(0)	f(1)	f(2)	f(3)	f(4)	f(5)	f(6)	f(7)
G(0/8)	W^0	W^0	W^0	W^0	W^0	W^0	W^0	W^0
G(1/8)	W^0	W^1	W^2	W^3	W^4	W^5	W^6	W^7
G(2/8)	W^0	W^2	W^4	W^6	W^0	W^2	W^4	W^6
G(3/8)	W^0	W^3	W^6	W^1	W^4	W^7	W^2	W^5
G(4/8)	W^0	W^4	W^0	W^4	W^0	W^4	W^0	W^4
G(5/8)	W^0	W^5	W^2	W^7	W^4	W^1	W^6	W^3
G(6/8)	W^0	W^6	W^4	W^2	W^0	W^6	W^4	W^2
G(7/8)	W^0	W^7	W^6	W^5	W^4	W^3	W^2	W^1

表 3.1

これを偶数と奇数に分けると、

偶数

同じ

	f(0)	f(2)	f(4)	f(6)
G(0/8)	W^0	W^0	W^0	W^0
G(1/8)	W^0	W^2	W^4	W^6
G(2/8)	W^0	W^4	W^0	W^4
G(3/8)	W^0	W^6	W^4	W^2
G(4/8)	W^0	W^0	W^0	W^0
G(5/8)	W^0	W^2	W^4	W^6
G(6/8)	W^0	W^4	W^0	W^4
G(7/8)	W^0	W^6	W^4	W^2

表 3.2

奇数

	f(1)	f(3)	f(5)	f(7)
G(0/8)	W^0	W^0	W^0	W^0
G(1/8)	W^1	W^3	W^5	W^7
G(2/8)	W^2	W^6	W^2	W^6
G(3/8)	W^3	W^1	W^7	W^5
G(4/8)	W^4	W^4	W^4	W^4
G(5/8)	W^5	W^7	W^1	W^3
G(6/8)	W^6	W^2	W^6	W^2
G(7/8)	W^7	W^5	W^3	W^1

表 3.3

奇数の場合は、W^n をくくりだすと、

$n=0$ のとき、$G(0/8)^{odd} = W^0\{f(1)W^0 + f(3)W^0 + f(5)W^0 + f(7)W^0\}$

$n=1$ のとき、$G(1/8)^{odd} = W^1\{f(1)W^0 + f(3)W^2 + f(5)W^4 + f(7)W^6\}$

$n=2$ のとき、$G(2/8)^{odd} = W^2\{f(1)W^0 + f(3)W^4 + f(5)W^0 + f(7)W^4\}$

$n=3$ のとき、$G(3/8)^{odd} = W^3\{f(1)W^0 + f(3)W^6 + f(5)W^4 + f(7)W^2\}$

$n=4$ のとき、$G(4/8)^{odd} = W^4\{f(1)W^0 + f(3)W^0 + f(5)W^0 + f(7)W^0\}$

$n=5$ のとき、$G(5/8)^{odd} = W^1\{f(1)W^0 + f(3)W^2 + f(5)W^4 + f(7)W^6\}$

$n=6$ のとき、$G(6/8)^{odd} = W^2\{f(1)W^0 + f(3)W^4 + f(5)W^0 + f(7)W^4\}$

$n=7$ のとき、$G(7/8)^{odd} = W^3\{f(1)W^0 + f(3)W^6 + f(5)W^4 + f(7)W^2\}$

W^nをくくりだした奇数は次のようにまとめられる。

同じ

	f(1)	f(3)	f(5)	f(7)
G(0/8)	W^0	W^0	W^0	W^0
G(1/8)	W^0	W^2	W^4	W^6
G(2/8)	W^0	W^4	W^0	W^4
G(3/8)	W^0	W^6	W^4	W^2
G(4/8)	W^0	W^0	W^0	W^0
G(5/8)	W^0	W^2	W^4	W^6
G(6/8)	W^0	W^4	W^0	W^4
G(7/8)	W^0	W^6	W^4	W^2

表 3.4

偶数と奇数部分で上の部分と下の部分が共通なので計算回数が半分になることがわかる。但し偶数の部分でW^nを 4 回かけないといけないので、16 + (16+4) =36 回になる。

64 回　→　36 回

同じ操作をするとさらに回数が減る。f(k)の偶数部分の f(2k)を p(k)とする。2k → k となることに注意して、($W^{2k} = V^k$)、$W^8=W^0$ なので、 $V^4=V^0$ となる。

	p(0)	p(1)	p(2)	p(3)
G(0/8)	V^0	V^0	V^0	V^0
G(1/8)	V^0	V^1	V^2	V^3
G(2/8)	V^0	V^2	V^0	V^2
G(3/8)	V^0	V^3	V^2	V^1
G(4/8)	V^0	V^0	V^0	V^0
G(5/8)	V^0	V^1	V^2	V^3
G(6/8)	V^0	V^2	V^0	V^2
G(7/8)	V^0	V^3	V^2	V^1

表 3.5

が求まる。偶数と奇数の部分に分けると、

	p(0)	p(2)
G(0/8)	V^0	V^0
G(1/8)	V^0	V^2
G(2/8)	V^0	V^0
G(3/8)	V^0	V^2
G(4/8)	V^0	V^0
G(5/8)	V^0	V^2
G(6/8)	V^0	V^0
G(7/8)	V^0	V^2

表 3.6

また、奇数部の V^n をくくりだしたものは、$V^4=V^0$ を考慮して、

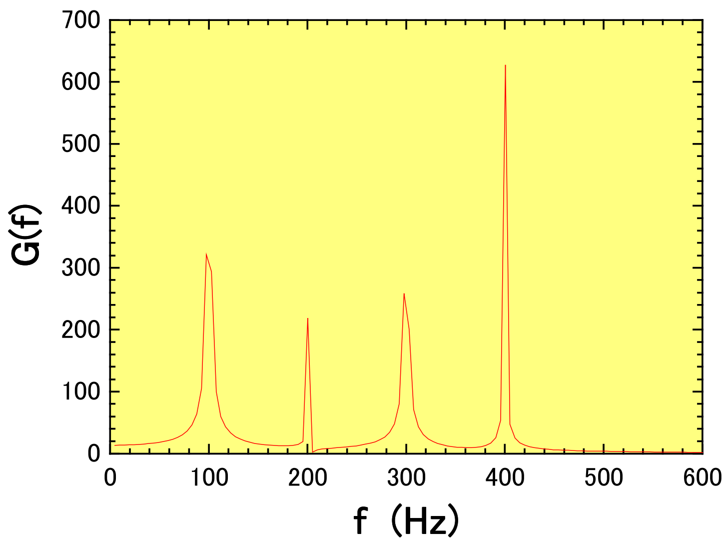
	p(1)	p(3)
G(0/8)	V^0	V^0
G(1/8)	V^0	V^2
G(2/8)	V^0	V^0
G(3/8)	V^0	V^2
G(4/8)	V^0	V^0
G(5/8)	V^0	V^2
G(6/8)	V^0	V^0
G(7/8)	V^0	V^2

表 3.7

問題

N=8 の DFT の計算（64 回）は、FFT の計算で何回に減らすことができるか？

FFT の計算では、半分、半分、・・・・と分けて考えているので、データは、2^n 個でなければならない。周期のわからない f(t)を DFT で計算したが（図 3・9）、同じ f(t)を FFT のプログラムで計算してみる。



```
Dim pi As Double, nD As Integer, nPoly As Integer
Dim Xr(5000) As Double, Xi(5000) As Double
Dim ce(2, 10) As Single, tau As Double
Sub init_wave()
   pi = 4# * Atn(1#)      ' π  円周率
   f1 = 100                ' Frequency (Hz)
   T1 = 1# / f1            ' 周期 (sec)
   tau = T1 / 50           ' τ     (sec)
   nD = 2 ^ 10               ' number of DATA
'-----------------------------------------------------------------
   nPoly = 4
   a0 = 0.5                ' 初期値
   ce(1, 1) = 0.5: ce(1, 2) = 0.2: ce(1, 3) = -0.7: ce(1, 4) = -1.2
   ce(2, 1) = 0.8: ce(2, 2) = -0.4: ce(2, 3) = 0.1: ce(2, 4) = 0.3
'-----------------------------------------------------------------
   t = 0#
   For I = 1 To nD
      t = (I - 1) * tau
      Xr(I) = a0            ' Real part         (Input DATA)
      Xi(I) = 0#            ' Imaginary part
      For K = 1 To nPoly
         Xr(I) = Xr(I) + ce(1, K) * Cos(K * 2# * pi * f1 * t) + ce(2, K) * Sin(K * 2# * pi * f1 * t)
      Next K
      Cells(I, 1) = t
      Cells(I, 2) = Xr(I)
      t = t + dt
   Next I
'****************
   FFT_1D
'****************
   For I = 1 To nD / 2
      f = I / nD / tau
      Cells(I, 4) = f
      Cells(I, 5) = Xr(I)
      Cells(I, 6) = Xi(I)
      Cells(I, 7) = Sqr(Xr(I) ^ 2 + Xi(I) ^ 2)
   Next I
End Sub '
```

```
Sub FFT_1D()
   Dim s(5000) As Double, c(5000) As Double
   Dim A As Double, B As Double, dA As Double
   Dim M As Integer, H As Integer, G As Integer, P As Integer, Q As Integer

   M = Log(nD) / Log(2)              ' N = 2^m
   A = 0#
   dA = 2# * pi / nD                 ' 2π/N
   For I = 0 To nD / 2
      s(I) = Sin(A)                  ' W^k = exp{ -i(2π/N)k }
      c(I) = Cos(A)
      A = A + dA
   Next I

   L = nD
   H = 1
   For G = 1 To M
      L = L / 2
      K = 0
      For Q = 1 To H
         P = 0
         For I = K To L + K - 1
            J = L + I
            A = Xr(I) - Xr(J)
            B = Xi(I) - Xi(J)
            Xr(I) = Xr(I) + Xr(J)
            Xi(I) = Xi(I) + Xi(J)
            If P = 0 Then
               Xr(J) = A
               Xi(J) = B
            Else
               Xr(J) = A * c(P) + B * s(P)
               Xi(J) = B * c(P) - A * s(P)
            End If
            P = P + H
         Next I
         K = K + L + L
      Next Q
      H = H + H
   Next G

   J = nD / 2
   For I = 1 To nD - 1
      K = nD
      If J < I Then
         Dummy = Xr(I): Xr(I) = Xr(J): Xr(J) = Dummy
         Dummy = Xi(I): Xi(I) = Xi(J): Xi(J) = Dummy
      End If
      Do
         K = K / 2
         If J >= K Then J = J - K Else Exit Do
      Loop
      J = J + K
   Next I
End Sub
```