【離散コサイン変換　　Discrete Cosine Transform】

■ DCT

$$X_k = \sum_{n=0}^{N-1} x_n cos(\frac{\pi k(2n+1)}{2N})$$

■ IDCT

$$x_n = \frac{1}{N}X_0 + \frac{2}{N}\sum_{k=1}^{N-1} X_k cos(\frac{\pi k(2n+1)}{2N})$$

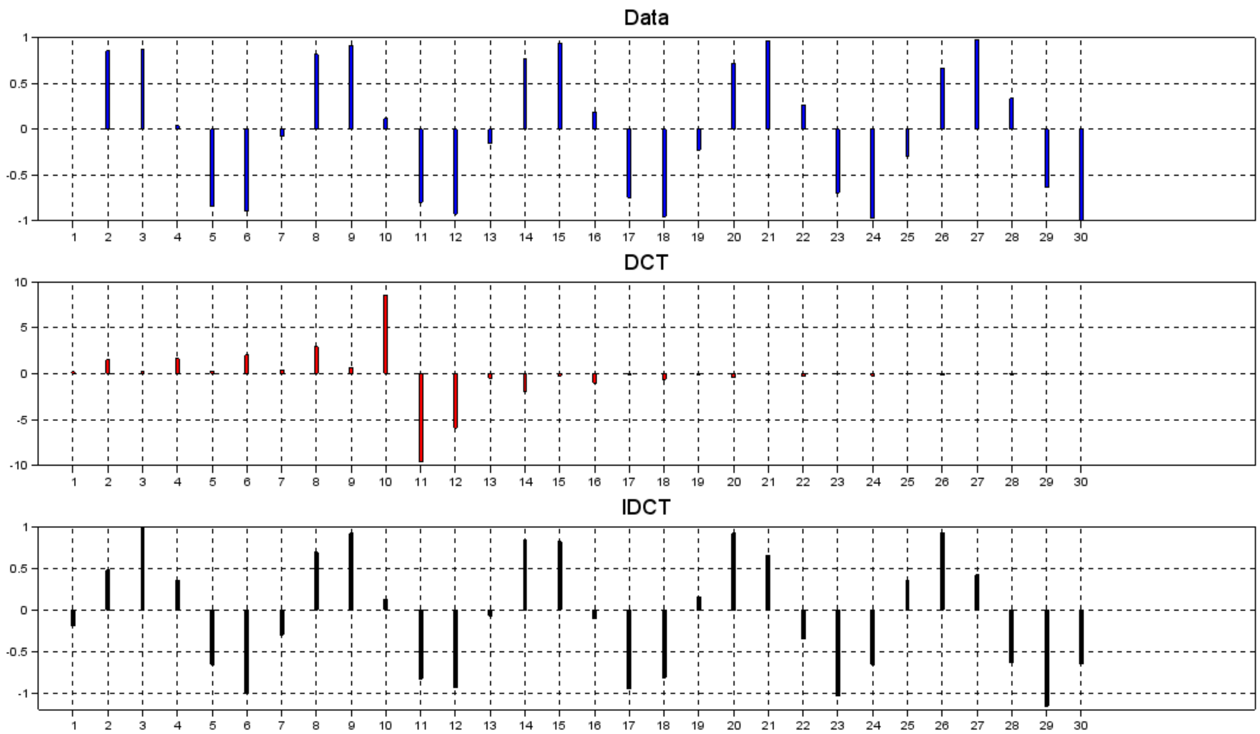

Figure 1: Scilab 実行結果

Source Code 1: Scilab

```
//////////////////////////////////
//      離散コサイン変換
//  Discrete Cosine Transform
//
//                         M.Tsutsui
//////////////////////////////////

clear all;

pi=%pi;

Samp_size=30;//サンプル数
sita=linspace(0,Samp_size,Samp_size);
Samp=sin(sita);//データ

[s_size1,s_size2]=size(Samp);

cos_vec=[];

for k=1:1:s_size2;//cos計算
    for n=1:1:s_size2;
        cos_vec(k,n)=cos(pi*(k-1)*(2*(n-1)+1)/(2*s_size2));
```

```
        end
end


//______________________DCT____________________//

DCT=cos_vec*Samp';//DCT Coefficient

//______________________IDCT____________________//

i_cos_va=[];
for J=2:1:s_size2;
        i_cos_va=[i_cos_va,cos_vec(:,J)];//DCT cosin matrix-> n=0 cut
end

X_k_1=DCT(2:s_size2);//Xk ->X0 cut

IDCT_p=2/s_size2*i_cos_va*X_k_1;
[size3,size4]=size(IDCT_p);
IDCT=1/s_size2*DCT(1,1)*ones(size3,1)+IDCT_p;

//__________________plot______________
subplot(311);
bar(Samp,0.1);
title('Data','fontsize',4);
xgrid();

subplot(312);
bar(DCT,0.1,'r');
title('DCT','fontsize',4);
xgrid();

subplot(313);
bar(IDCT,0.1,'k');
title('IDCT','fontsize',4);
xgrid();
```

Source Code 2: Python

```
#-----------------------------------------
# Module Name:Discrete Cosine Transform
# Author: m_tsutsui
#-----------------------------------------

#Library_Import##############################
from numpy import*
from scipy import*
from pylab import *

import math, numpy as np
import matplotlib.pyplot as plt
#Library_Import_end##########################

Samp_size=30
sita=np.linspace(0,Samp_size,Samp_size*2)
Samp=sin(sita)
Samp_M=matrix(Samp)

SC1=np.size(Samp_M,0)#size_check1
SC2=np.size(Samp_M,1)#size_check2

cos_vec=[]
for K in range(0,SC2,1):
```

```
    for N in range(0,SC2,1):
        cos_vec.append(cos(pi*K*(2*N+1)/(2*SC2)))

SC3=np.size(cos_vec,0)#size_check3

cos_va=[]
for J in range(1,SC2+1,1):
    cos_va.append(cos_vec[SC2*J-SC2:SC2*J])

cos_va_M=matrix(cos_va)

SC4=np.size(cos_va_M,0)#size_check4
SC5=np.size(cos_va_M,1)#size_check5

DCT=cos_va_M*Samp_M.T #DCT_Coefficient
#_DCT_###########################################
x_def=np.linspace(0,SC2,SC2)
plt.figure(facecolor='w')#Backgroundcolor_white
plt.subplot(211)
plt.grid()#grid on
plt.bar(x_def,array(DCT),width=0.1,color='k')
plt.title('DCT')

#_IDCT_###########################################
i_cos_va1=cos_vec[10:Samp_size*Samp_size+1] #DCT cosin matrix ->n=0 cut

SC6=np.size(i_cos_va1,0)#size_check6

i_cos_va2=[]
for J_i in range(1,SC2+1,1):
    i_cos_va2.append(cos_vec[SC2*J_i-(SC2-1):SC2*J_i])

i_cos_va2_M=matrix(i_cos_va2)

SC7=np.size(i_cos_va2_M,0)#size_check7
SC8=np.size(i_cos_va2_M,1)#size_check8


X_k_1=DCT[1:SC2]#Xk ->X0 cut
X_k_1_M=matrix(X_k_1)

SC8=np.size(X_k_1,0)#size_check8

IDCT_p=2/SC2*i_cos_va2_M*X_k_1_M

IDCT_p_M=matrix(IDCT_p)

SC9=np.size(IDCT_p_M,0)#size_check9
SizC=np.size(IDCT_p_M,1)#size_check10

IDCT=1/SC2*(DCT[0])+IDCT_p_M

subplot(212)
plt.bar(x_def,array(IDCT),width=0.1,color='k')
plt.title('IDCT')
plt.grid()#grid on
plt.show()
```