創価大学

Discover your potential
自分力の発見

DCT符号化による画像圧縮実験補助スライド

創価大学　理工学部　情報システム工学科

今村　弘樹

簡単な画像処理プログラミング

Windowsで必要なアプリ

minGW(gccのコンパイラ),　　Irfanview(画像ビューア)

Macでのテキストエディタと画像ビューアの起動方法

open –a textedit imageio.c[Enter key],　open –a preview lenna.pgm[Enter key]

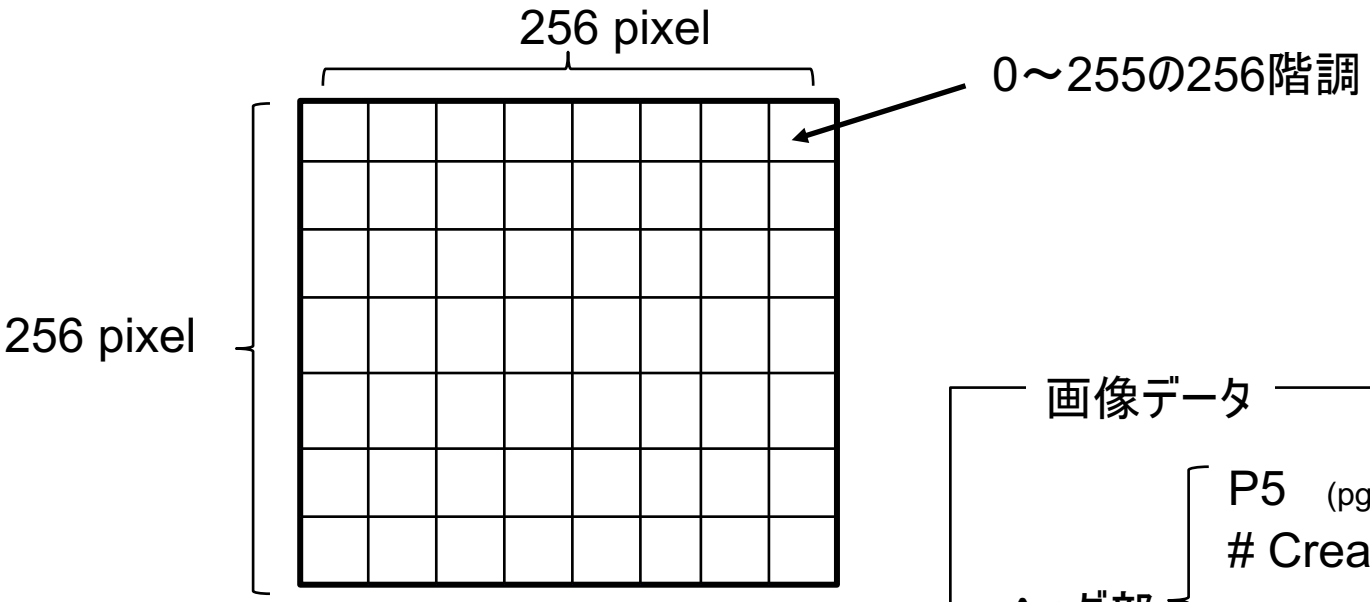
ソースコードのコンパイル方法

gcc imageio.c –o imageio –lm[Enter key]

実行方法

./imageio lenna[Enter key]

画像データ



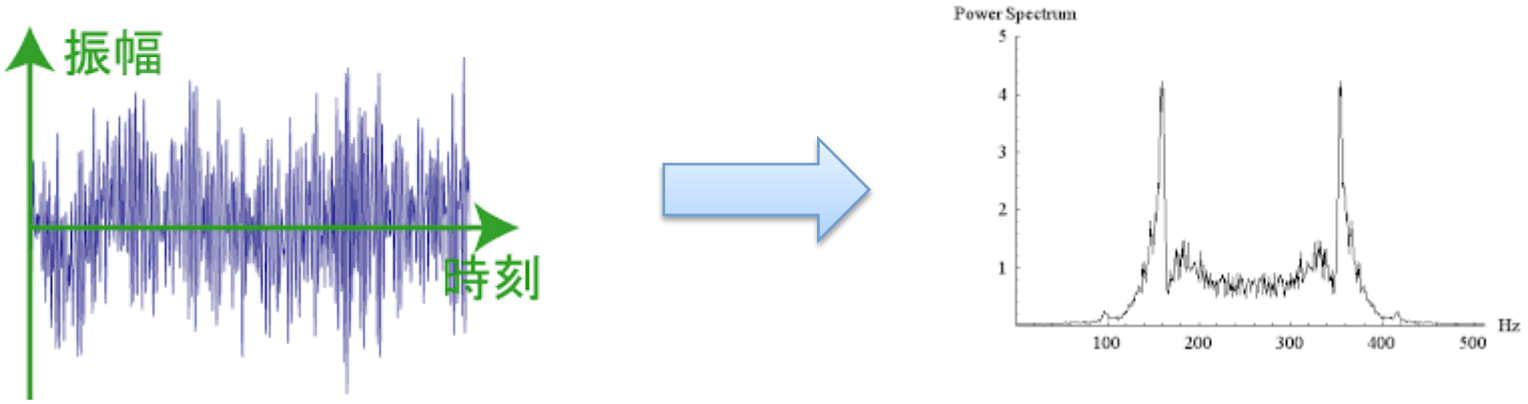
画像データ

ヘッダ部
- P5 (pgm形式のグレースケール画像)
- # Created by Irfanview
- 256 256 (画像の縦横の画素数)
- 255 (各画素の階調数)

0～255の輝度データ
○ ○ ○ ○ ○ …

フーリエ変換

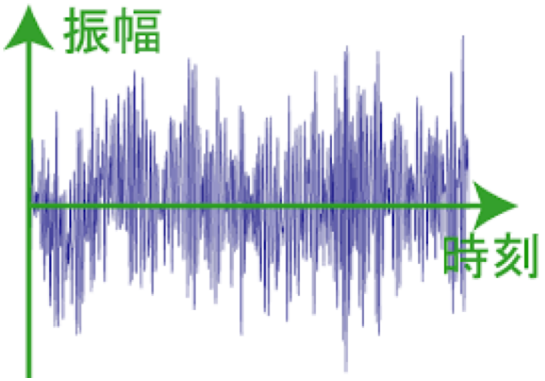
信号をコンピュータで扱う



フーリエ変換

フーリエ変換

信号をコンピュータで扱う



フーリエ変換

$\times \quad w_0 \quad \times \alpha = c_0$

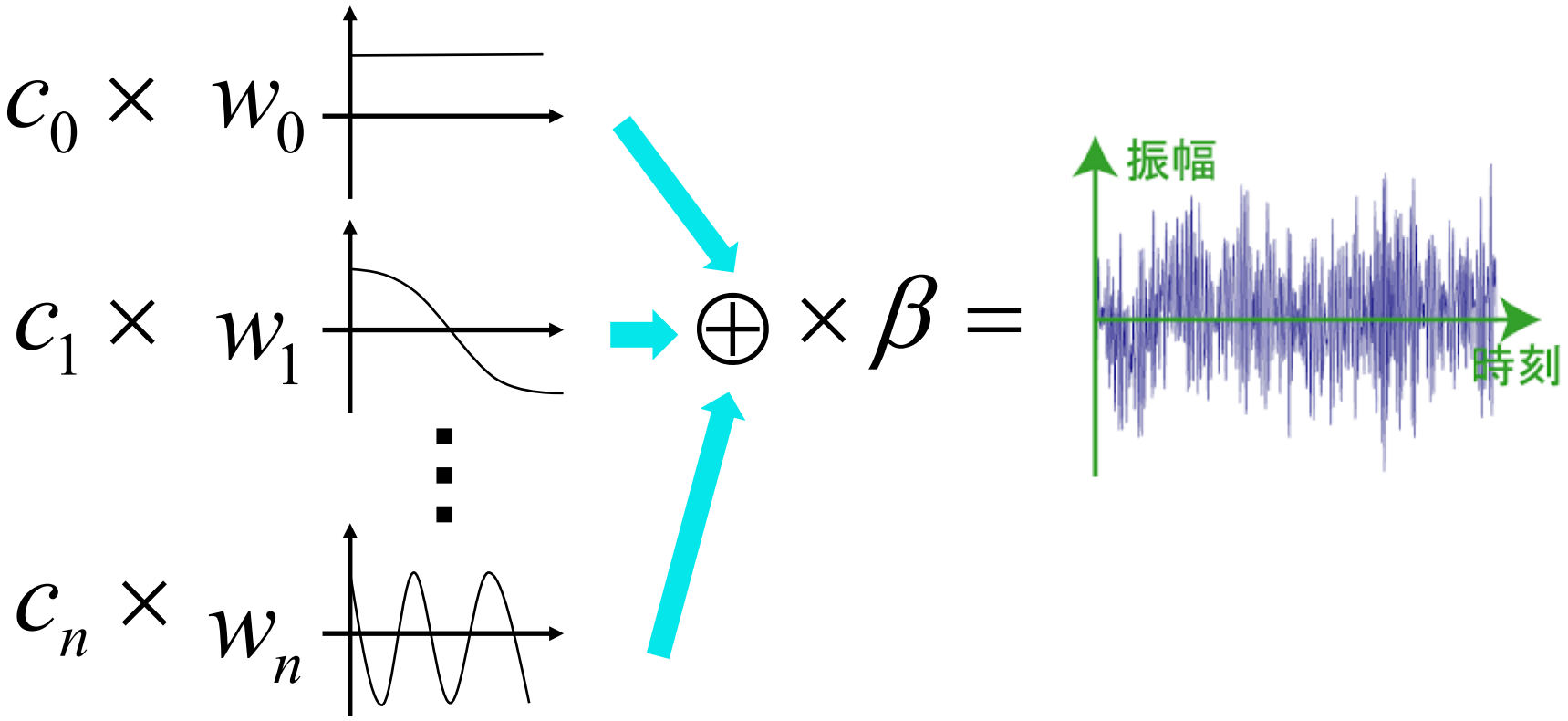
$\times \quad w_1 \quad \times \alpha = c_1$

$\times \quad w_n \quad \times \alpha = c_n$

フーリエ変換

信号をコンピュータで扱う

フーリエ逆変換



DCT変換・逆変換の実装

創価大学

DCT変換
逆変換の
実装場所

```
write_pgm_file(fp, 1);

for(k=0;k<y_size;k++){                    // Main Process
  for(j=0;j<x_size;j++){
        image_out[j][k] = image_in[j][k];
  }
}

for(k=0;k<y_size;k++){
 for(j=0;j<x_size;j++){
   fputc(image_out[j][k]  ,fp[1]);
 }
}
```

ここに実装する。

DCT変換・逆変換の実装

創価大学

DCT変換処理

```
for(x=0;x<x_size;x+=8)
{
  for(y=0;y<y_size;y+=8)
  {
    for(u=0;u<N;u++)
    {
      for(v=0;v<N;v++)
      {
```

Cu,Cvの場合分け

(5)式のΣの計算

（5）式全体を計算し、c[x+u][y+v]に格納

```
      }
    }
  }
}
```

DCT逆変換処理

```
for(x=0;x<x_size;x+=8)
{
  for(y=0;y<y_size;y+=8)
  {
    for(j=0;j<N;j++)
    {
      for(k=0;k<N;k++)
      {
```

(6)式のΣの計算

Σの計算の中でCu,Cvの場合分け

（6）式全体を計算し、image_out[x+j][y+k]に格納

```
      }
    }
  }
}
```

DCT変換の実装

創価大学

Cu, Cvの場合分け

```
double Cu,Cv;
.
.
.
if(u==0) Cu=1/sqrt(2); else Cu =1;
if(v==0) Cv=1/sqrt(2); else Cv =1;
```

DCT変換の実装

創価大学

(5)式のΣの計算

```
double sum;
int N=8;
…
sum=0;
for(j=0;j<N;j++)
{
  for(k=0;k<N;k++)
  {
    sum += image_in[x+j][y+k]*cos((2*j+1)*u*M_PI/16.0)
                *cos((2*k+1)*v*M_PI/16.0);
  }
}
```

DCT変換の実装

創価大学

(5)式全体を計算し、c[x+u][y+v]に格納

```
double c[X_SIZE][Y_SIZE];
…
c[x+u][y+v]=0.25*Cu*Cv*sum;
```

DCT逆変換の実装

創価大学

(6)式のΣの計算

```
int BL=8
…
sum=0;
for(u=0;u<BL;u++)
{
  for(v=0;v<BL;v++)
  {
    Σの計算の中でCu,Cvの場合分け
    if(u==0) Cu=1/sqrt(2); else Cu=1;
    if(v==0) Cv=1/sqrt(2); else Cv=1;
    sum+=Cu*Cv*c[x+u][y+v]*cos((2*j+1)*u*M_PI/16.0)
                *cos((2*k+1)*v*M_PI/16.0);
  }
}
```

DCT逆変換の実装

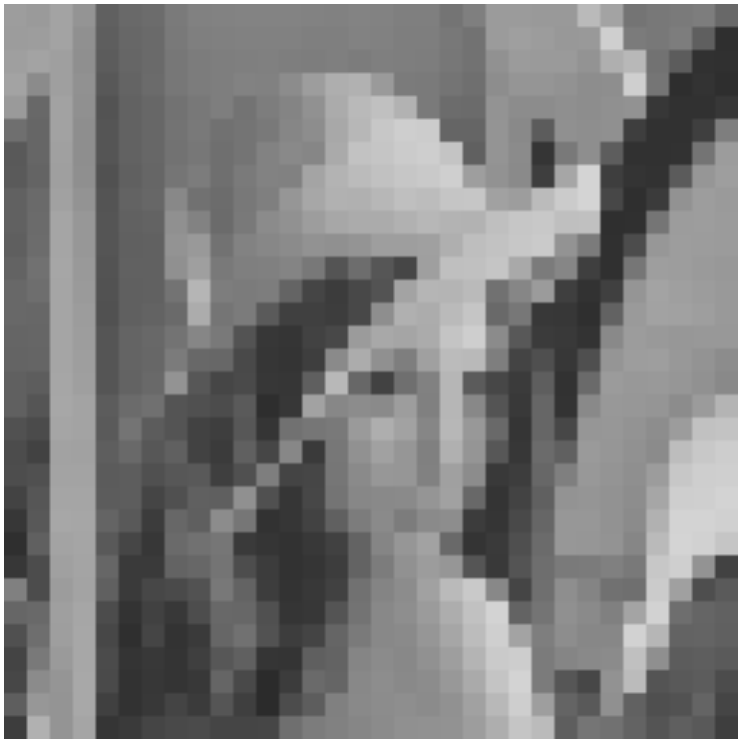
創価大学

(6)式全体を計算し、image_out[x+j][y+k]に格納

```
image_out[x+j][y+k]=0.25*sum;
```

DCT変換・逆変換の実装（確かめ）

(6)式のΣの計算の箇所のNをBLと変更し、BL＝1として、画像lenna.pgmに対して、DCT変換・逆変換し、以下のような画像が生成されれば、正しく実装されたこととなる。



客観的画質評価(PSNR)の実装

imageio2.cのコンパイルの仕方

gcc imageio2.c –o imageio2 –lm[Enter Key]

imageio2の実行の仕方

./imageio2 lenna lenna_output[Enter Key]

客観的画質評価(PSNR)の実装

実装のポイント

$log_{10} *$ は、C言語で$\log 10(*)$で計算できる。

$*^2$ は、$pow(*, 2)$で計算できる。

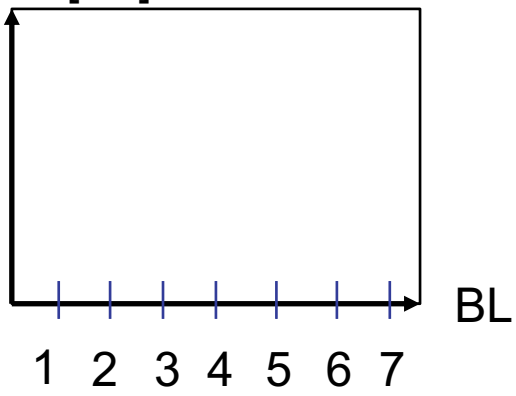
A/B の計算の際、Bは、実数型($double$か$float$)とする。
Bがint型の場合は、(double)Bと実数型にキャストする。

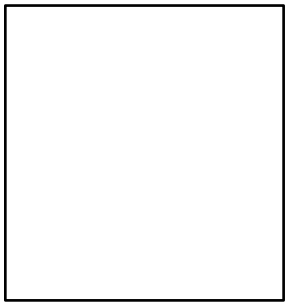
lennaのBL＝1と原画像との客観的画質評価の結果は、
PSNR=21.****となれば、正しく実装されている。

レポートの書き方

高周波成分が多い画像

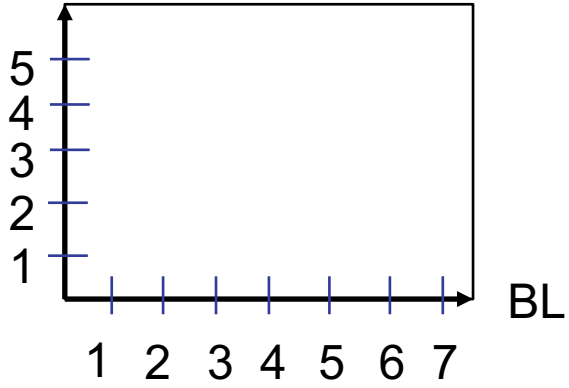
PSNR[dB]

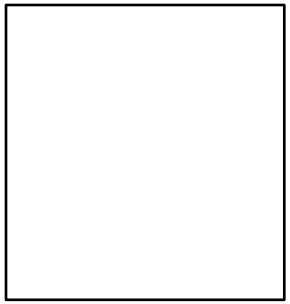




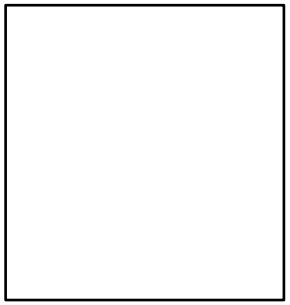
原画像

Score



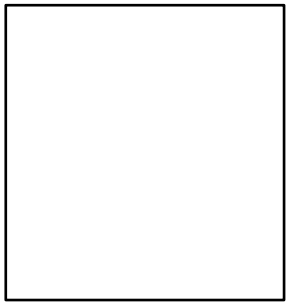


BL＝1



BL＝2

…

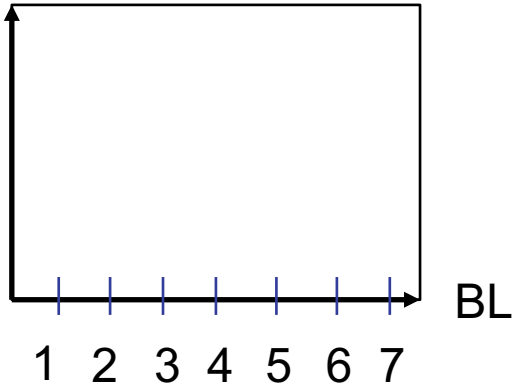


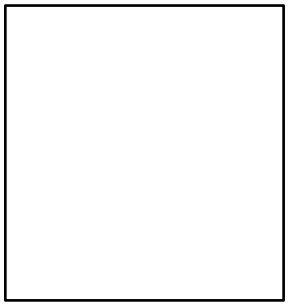
BL＝7

レポートの書き方

中間の周波数成分が多い画像

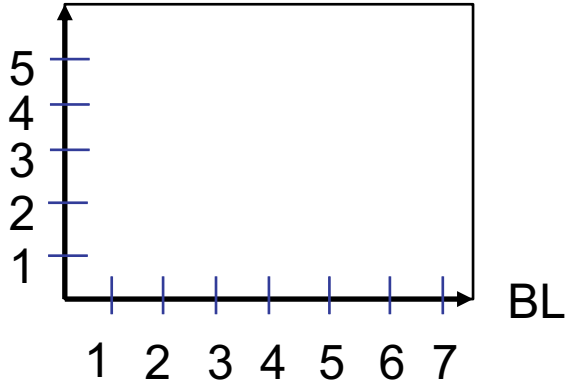
PSNR[dB]

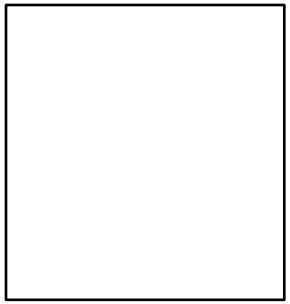




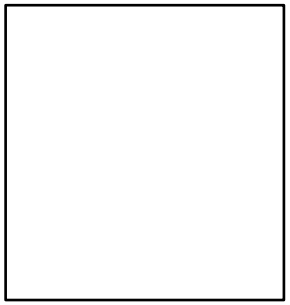
原画像

Score



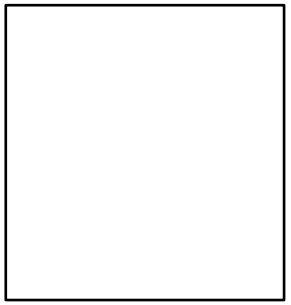


BL＝1



BL＝2

…

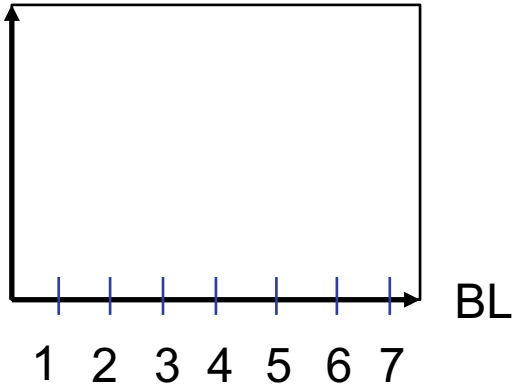


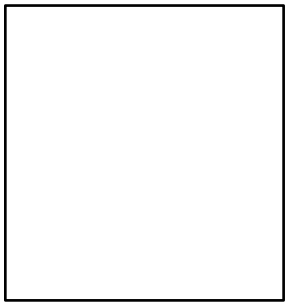
BL＝7

レポートの書き方

低周波成分が多い画像

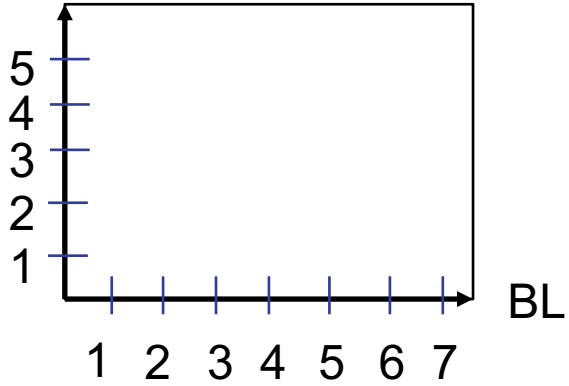
PSNR[dB]

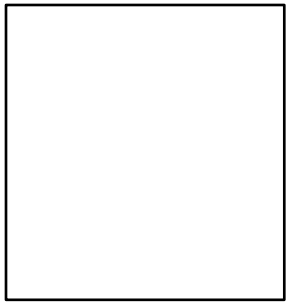




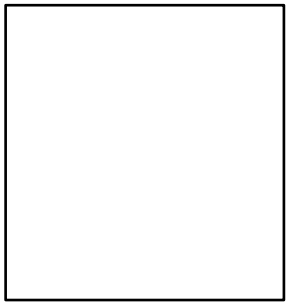
原画像

Score



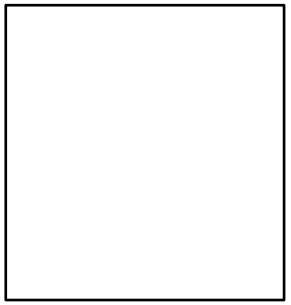


BL＝1



BL＝2

…



BL＝7

レポートの書き方

注意

・レポートの形式は、一般的なレポートの形式（表紙、実験の目的、実験の原理、課題の考察、参考文献）に沿って、作成すること。

・各グラフは、gnuplotを用いず、MicrosoftのExcelを用いて作成すること。

・画像は、レポート作成に用いるアプリ（Word等）で使用できる画像形式(png、jpeg、gif等)に変換して、用いること。

・実験で用いたプログラムのソースコードは、レポートに記載する必要は、無い。

・明らかにコピーであるレポートは、コピーした側、コピーされた側も、採点はしない。