

# 格子点データの解析

## 1 月平均全球客観解析データの解析

客観解析データや衛星観測データのような格子点データは、バイナリ形式のデータファイルに記録されることが多いです。バイナリ形式のデータファイルは、テキスト形式の場合とは異なり、直接中身を見ることができません。プログラムを書いてデータを読み出して解析するのが普通です。ここでは、全球客観解析データを用いてバイナリ形式のファイルに記録された格子点データの解析について学びたいと思います。

はじめに、月平均海面気圧のデータファイルを開いて、特定の月のデータを読み出してみます。データファイルには1979年1月からの月別データが書きこまれています。ここでは、2011年1月のデータを読み出すプログラムを作成します。

FORTRAN では、OPEN 文でバイナリ形式のファイルを開きます。テキスト形式の場合とは引数の指定が異なります。バイナリ形式の場合には、FORM='UNFORMATTED' と指定し、さらに、ACCESS='DIRECT' とします。また、RECL=...で、レコード長をバイト数(整数値)で指定します。実数型の配列の場合、各要素は4バイトで、要素の数はLMAX\*MMAX(LMAXは経度方向の格子数、MMAXは緯度方向の格子数)なので、レコード長は4\*LMAX\*MMAXバイトになります。C では、fopen 関数を使ってファイルを開きます。モードが"rb" であることに注意します。

FORTRAN では、READ 文でデータを読みこみます。書式の指定はありませんが、REC=...でレコード番号を指定します。2011年1月のデータは385番目のデータですので、レコード番号は385です。C では、fseek 関数を使ってデータファイル上の所定の位置に移動、つまり384個のレコードの分だけ前に移動した後で、fread 関数でデータを読みこみます。書き出しの場合は、FORTRAN の場合はWRITE 文、C の場合はfwrite 関数を使います。

プログラム：

FORTRAN

```
C PARAMETER文で定数LMAXとMMAXを宣言する。
C 定数LMAXは経度方向の格子数を、MMAXは緯度方向の格子数を表す。
  PARAMETER (LMAX=144, MMAX=73)

C 配列Aを宣言する。
  REAL A(LMAX, MMAX)

C データを読みこむ。

C ファイルを開く。
C 機番は10以降の番号を指定する。
C STATUSは読みこみの場合は'OLD'を指定する。
```

C バイナリファイルの場合は、FORMには' UNFORMATTED' を指定する。

C さらに、ACCESSには' DIRECT' を指定し、

C RECLにはレコード長をバイト数(整数値)で指定する。

C 各要素は4バイトなので、レコード長は4×要素数である。

```

OPEN(10, FILE=' SLP. dat' , STATUS=' OLD' ,
+   FORM=' UNFORMATTED' ,
+   ACCESS=' DIRECT' , RECL=4*LMAX*MMAX)

```

C READ文でデータを読みこむ。

C 機番10を指定する。REC=... でレコード番号を指定する。

C ここでは、385番目のレコードを読みこむ。

C 読みこんだデータを配列Aに格納する。

```

READ(10, REC=385) A

```

C ファイルを閉じる。

```

CLOSE(10)

```

C データを書き出す。

C ファイルを開く。

C 機番は10以降の番号を指定する。

C STATUSは書き出しの場合は' UNKNOWN' を指定する。

C バイナリファイルの場合は、FORMには' UNFORMATTED' を指定する。

C さらに、ACCESSには' DIRECT' を指定し、

C RECLにはレコード長を整数値で指定する。

```

OPEN(10, FILE=' output. dat' , STATUS=' UNKNOWN' ,
+   FORM=' UNFORMATTED' ,
+   ACCESS=' DIRECT' , RECL=4*LMAX*MMAX)

```

C 配列Aの各要素の値を機番10で指定されたファイルに書き出す。

C REC=1と指定し、1番目のレコードとして出力する。

```

WRITE(10, REC=1) A

```

C ファイルを閉じる。

```

CLOSE(10)

STOP
END

```

(参考) C

```

#include <stdio.h>

int main(void)
{

/* 定数 mmax と lmax を宣言する。
   定数 mmax は緯度方向の格子数を、lmax は経度方向の格子数を表す。 */
   int mmax=73, lmax=144;

/* 配列 a を宣言する。 */
   float a[mmax][lmax];

/* ファイルポインタ fp を宣言する。 */
   FILE *fp;

/* データを読みこむ。 */

/* ファイルを開く。
   モードは"rb"(バイナリファイルの読みこみ)を指定する。 */
   fp = fopen( "SLP.dat", "rb" );

/* ファイルポインタ fp で指定されたファイルの所定の位置に移動する。
   第3引数に SEEK_SET を指定し、ファイルの先頭を位置の基準とする。
   ここでは、385 番目のレコードを読みこむので、
   まず、ファイルの先頭から
   (385-1)*mmax*lmax*sizeof(float)バイトだけ前方に移動する。 */
   fseek( fp, (385-1)*mmax*lmax*sizeof(float), SEEK_SET );

/* 配列 a の各要素の値をファイルポインタ fp で指定されたファイルから読みこむ。
   項目のサイズは浮動小数点のサイズ sizeof(float)を指定する。
   項目の数は mmax*lmax を指定する。 */
   fread( a, sizeof(float), mmax*lmax, fp );

/* ファイルを閉じる。 */
   fclose( fp );

/* データを書き出す。 */

/* ファイルを開く。
   モードは"wb"(バイナリファイルへの書き出し)を指定する。 */

```

```

fp = fopen( "output.dat", "wb" );

/* 配列 a の各要素の値をファイルポインタ fp で指定されたファイルに書き出す。
   項目のサイズは浮動小数点のサイズ sizeof(float) を指定する。
   項目の数は mmax*lmax を指定する。 */
fwrite( a, sizeof(float), mmax*lmax, fp );

/* ファイルを閉じる。 */
fclose( fp );

return 0;
}

```

計算結果は output.dat というファイルに出力されましたが、このファイルの形式をコントロールファイルに記述しておきます。以下のようなテキスト形式のファイルを作成し、output.ct1 というファイル名で保存しておきます。ここでは、変数名を“A”としています。GrADS では、output.ct1 を開いて、“a”という名前の変数として作図できます（各種設定をした後、“d a”とすればよい）。

コントロールファイル：

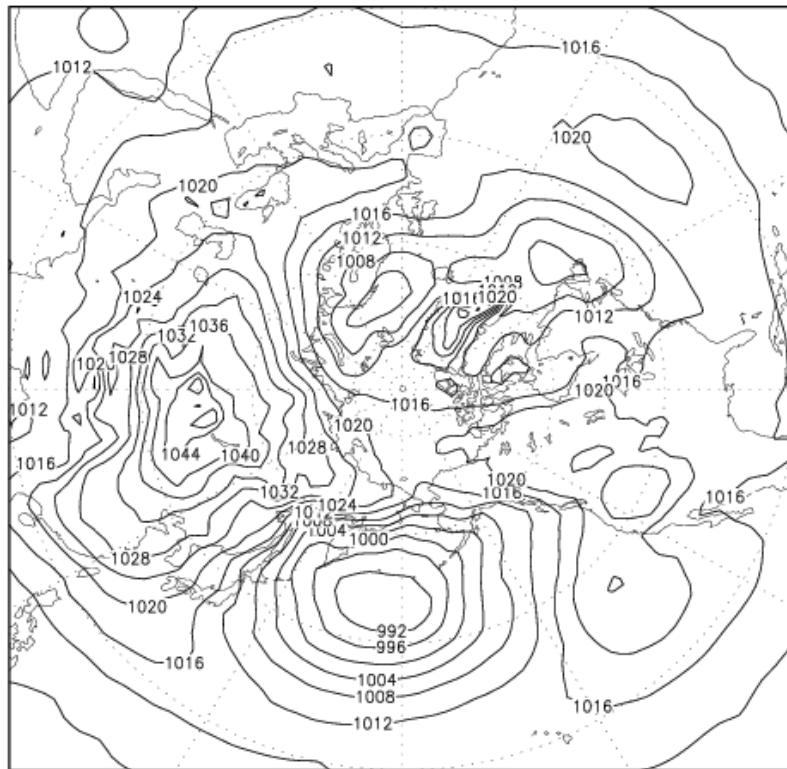
```

DSET  output.dat
UNDEF  -9.99E33
OPTIONS  yrev
XDEF  144  LINEAR  0.0  2.5
YDEF  73  LINEAR  -90.0  2.5
ZDEF  1  LEVELS  1000
TDEF  1  LINEAR  jan1979  1mo
VARS  1
A      0  99  Output
ENDVARS

```

上のコントロールファイルでは、DSET はデータファイル名、UNDEF は欠損値、“OPTIONS yrev”はデータの並び順が南から北ではなく北から南であることを示しています。XDEF (YDEF) は、東西 (南北) 方向の格子点の数、始点の経度 (緯度)、経度 (緯度) 間隔、ZDEF は各等圧面の気圧の値 (hPa)、TDEF は時刻 (この場合は 1979 年 1 月から 1 か月間隔) を表します。VARS は変数の種類の数を示し、そのあとで、各変数を列挙します。この場合は、変数は 1 種類で、変数名は“A”です。ひとつめの数字“0”は 2 次元であることを示します。3 次元の場合は鉛直方向の格子数を入れます。ふたつめの数字“99”は (特殊な書式ではなく) 通常の見出しであることを意味します。そのあと、同じ行に続けて変数名を書きます。

作図例：



次に、2010年12月から2011年2月までの3か月間の海面気圧の平均値を計算してみます。ここでは、3つの月別値の単純な平均を3か月間の平均とみなします。384番目から386番目までのレコードを順に読みこんで、足していき、最後にレコードの個数3で割ります。

プログラム：

FORTRAN

```
C PARAMETER文で定数LMAXとMMAXを宣言する。
C 定数LMAXは経度方向の格子数を、MMAXは緯度方向の格子数を表す。
  PARAMETER (LMAX=144, MMAX=73)

C 配列A、Bを宣言する。
  REAL A(LMAX, MMAX), B(LMAX, MMAX)

C 配列Aの各要素にゼロを代入する。
  DO 11 M=1, MMAX
  DO 12 L=1, LMAX
    A(L, M) = 0.
12 CONTINUE
11 CONTINUE

C データを読みこむ。
```

- C ファイルを開く。
  - C 機番は10以降の番号を指定する。
  - C STATUSは読みこみの場合は' OLD' を指定する。
  - C バイナリファイルの場合は、FORMIには' UNFORMATTED' を指定する。
  - C さらに、ACCESSには' DIRECT' を指定し、
  - C RECLにはレコード長をバイト数(整数値)で指定する。
  - C 各要素は4バイトなので、レコード長は4×要素数である。
- ```

OPEN(10, FILE=' SLP. dat' , STATUS=' OLD' ,
+   FORM=' UNFORMATTED' ,
+   ACCESS=' DIRECT' , RECL=4*LMAX*MMAX)

```

- C ここからデータを読みこむためのDOループが始まる。
  - C 読みこみたいレコードの個数だけ反復する。
  - C ここでは、384番目から386番目までのレコードを読みこむ。
- ```

DO 21 I=384, 386

```

- C READ文でデータを読みこむ。
  - C 機番10を指定する。REC=... でレコード番号を指定する。
  - C 読みこんだデータを配列Bに格納する。
- ```

READ(10, REC=I) B

```

- C 配列Bの各要素を配列Aの各要素に加える。
- ```

DO 31 M=1, MMAX
DO 32 L=1, LMAX
A(L, M) = A(L, M) + B(L, M)
32 CONTINUE
31 CONTINUE

```

- C ここでDOループ21が終了する。
- ```

21 CONTINUE

```

- C ファイルを閉じる。
- ```

CLOSE(10)

```

- C 平均値を計算する。

- C 配列Aの各要素をレコードの個数で割る。
- ```

DO 41 M=1, MMAX
DO 42 L=1, LMAX

```

```
      A(L, M) = A(L, M) / REAL(3)
```

```
42  CONTINUE
```

```
41  CONTINUE
```

C データを書き出す。

C ファイルを開く。

C 機番は10以降の番号を指定する。

C STATUSは書き出しの場合は' UNKNOWN' を指定する。

C バイナリファイルの場合は、FORMIには' UNFORMATTED' を指定する。

C さらに、ACCESSには' DIRECT' を指定し、

C RECLにはレコード長を整数値で指定する。

```
      OPEN(10, FILE=' output. dat' , STATUS=' UNKNOWN' ,  
+       FORM=' UNFORMATTED' ,  
+       ACCESS=' DIRECT' , RECL=4*LMAX*MMAX)
```

C 配列Aの各要素の値を機番10で指定されたファイルに書き出す。

C REC=1と指定し、1番目のレコードとして出力する。

```
      WRITE(10, REC=1) A
```

C ファイルを閉じる。

```
      CLOSE(10)
```

```
      STOP
```

```
      END
```

(参考) C

```
#include <stdio.h>  
  
int main(void)  
{  
  
/* 定数 mmax と lmax を宣言する。  
   定数 mmax は緯度方向の格子数を、lmax は経度方向の格子数を表す。 */  
   int mmax=73, lmax=144, i, m, l;  
  
/* 配列 a、b を宣言する。 */  
   float a[mmax][lmax], b[mmax][lmax];  
  
/* ファイルポインタ fp を宣言する。 */
```

```

FILE *fp;

/* 配列 a の各要素にゼロを代入する。 */
for (m=0; m<=mmax-1; m++)
{
    for (l=0; l<=lmax-1; l++)
    {
        a[m][l] = 0.;
    }
}

/* データを読みこむ。 */

/* ファイルを開く。
   モードは"rb"(バイナリファイルの読みこみ)を指定する。 */
fp = fopen( "SLP.dat", "rb" );

/* ここからデータを読みこむための for ループが始まる。
   読みこみたいレコードの個数だけ反復する。
   ここでは、384 番目から 386 番目までのレコードを読みこむ。 */
for (i=384-1; i<=386-1; i++)
{

/* ファイルポインタ fp で指定されたファイルの所定の位置に移動する。
   第 3 引数に SEEK_SET を指定し、ファイルの先頭を位置の基準とする。
   ファイルの先頭から i*mmax*lmax*sizeof(float) バイトだけ前方に移動する。 */
fseek( fp, i*mmax*lmax*sizeof(float), SEEK_SET );

/* 配列 b の各要素の値をファイルポインタ fp で指定されたファイルから読みこむ。
   項目のサイズは浮動小数点のサイズ sizeof(float) を指定する。
   項目の数は mmax*lmax を指定する。 */
fread( b, sizeof(float), mmax*lmax, fp );

/* 配列 b の各要素を配列 a の各要素に加える。 */
for (m=0; m<=mmax-1; m++)
{
    for (l=0; l<=lmax-1; l++)
    {
        a[m][l] = a[m][l] + b[m][l];
    }
}

```



```

    }

/* ここで for ループが終了する。 */
}

/* ファイルを閉じる。 */
fclose( fp );

/* 平均値を計算する。 */

/* 配列 A の各要素をレコードの個数で割る。 */
for (m=0; m<=mmax-1; m++)
{
    for (l=0; l<=lmax-1; l++)
    {
        a[m][l] = a[m][l] / 3.;
    }
}

/* データを書き出す。 */

/* ファイルを開く。
   モードは"wb"(バイナリファイルへの書き出し)を指定する。 */
fp = fopen( "output.dat", "wb" );

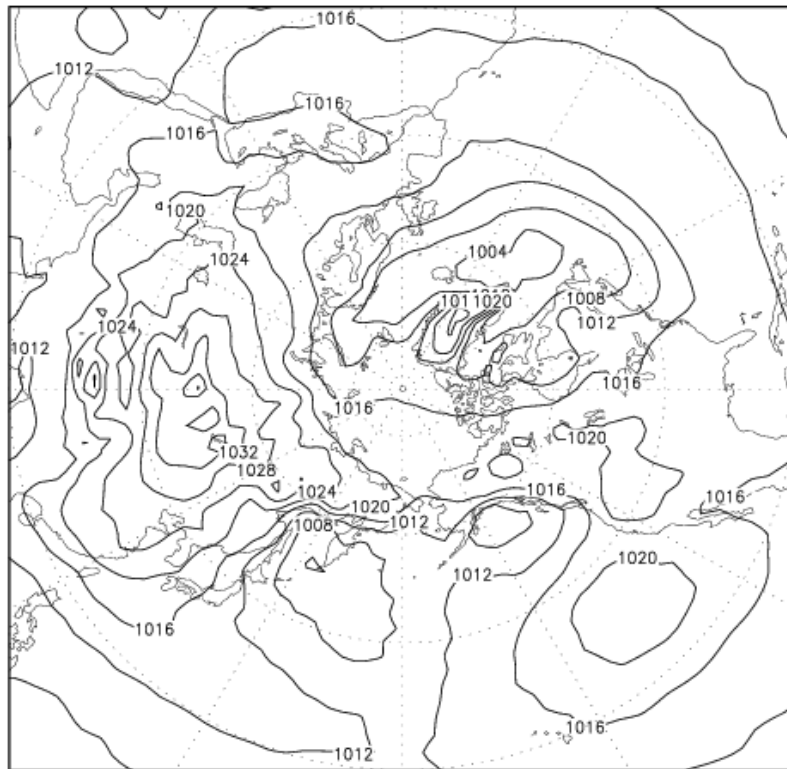
/* 配列 a の各要素の値をファイルポインタ fp で指定されたファイルに書き出す。
   項目のサイズは浮動小数点のサイズ sizeof(float) を指定する。
   項目の数は mmax*lmax を指定する。 */
fwrite( a, sizeof(float), mmax*lmax, fp );

/* ファイルを閉じる。 */
fclose( fp );

return 0;
}

```

作図例：



さらに、1月の月平均海面気圧の平年値（1981年から2010年までの30年間の平均値）を計算してみます。1981年1月のレコード（25番目）から2010年1月のレコード（373番目）を12個おきに順に読みこんで、足していき、最後にレコードの個数30で割ります。

プログラム：

FORTRAN

```
C PARAMETER文で定数LMAXとMMAXを宣言する。
C 定数LMAXは経度方向の格子数を、MMAXは緯度方向の格子数を表す。
  PARAMETER (LMAX=144, MMAX=73)

C 配列A、Bを宣言する。
  REAL A(LMAX, MMAX), B(LMAX, MMAX)

C 配列Aの各要素にゼロを代入する。
  DO 11 M=1, MMAX
  DO 12 L=1, LMAX
    A(L, M) = 0.
12 CONTINUE
11 CONTINUE

C データを読みこむ。
```

- C ファイルを開く。
  - C 機番は10以降の番号を指定する。
  - C STATUSは読みこみの場合は' OLD' を指定する。
  - C バイナリファイルの場合は、FORMには' UNFORMATTED' を指定する。
  - C さらに、ACCESSには' DIRECT' を指定し、
  - C RECLにはレコード長をバイト数(整数値)で指定する。
  - C 各要素は4バイトなので、レコード長は4×要素数である。
- ```

OPEN(10, FILE=' SLP. dat' , STATUS=' OLD' ,
+   FORM=' UNFORMATTED' ,
+   ACCESS=' DIRECT' , RECL=4*LMAX*MMAX)

```

- C ここからデータを読みこむためのDOループが始まる。
  - C 読みこみたいレコードの個数だけ反復する。
  - C ここでは、1981年から2010年までの1月のレコードを読みこむ。
- ```

DO 21 IY=1981, 2010

```

- C レコード番号を計算する。
  - C ここでは、25番目から373番目までのレコードを読みこむ。
- ```

I = 1 + 12 * (IY-1979)

```

- C READ文でデータを読みこむ。
  - C 機番10を指定する。REC=... でレコード番号を指定する。
  - C 読みこんだデータを配列Bに格納する。
- ```

READ(10, REC=I) B

```

- C 配列Bの各要素を配列Aの各要素に加える。
- ```

DO 31 M=1, MMAX
DO 32 L=1, LMAX
A(L, M) = A(L, M) + B(L, M)
32 CONTINUE
31 CONTINUE

```

- C ここでDOループ21が終了する。
- ```

21 CONTINUE

```

- C ファイルを閉じる。
- ```

CLOSE(10)

```

- C 平均値を計算する。

C 配列Aの各要素をレコードの個数で割る。

```
DO 41 M=1, MMAX
```

```
DO 42 L=1, LMAX
```

```
    A(L, M) = A(L, M) / REAL(30)
```

```
42 CONTINUE
```

```
41 CONTINUE
```

C データを書き出す。

C ファイルを開く。

C 機番は10以降の番号を指定する。

C STATUSは書き出しの場合は' UNKNOWN' を指定する。

C バイナリファイルの場合は、FORMIには' UNFORMATTED' を指定する。

C さらに、ACCESSには' DIRECT' を指定し、

C RECLにはレコード長を整数値で指定する。

```
    OPEN(10, FILE=' output. dat', STATUS=' UNKNOWN',
```

```
    +   FORM=' UNFORMATTED',
```

```
    +   ACCESS=' DIRECT', RECL=4*LMAX*MMAX)
```

C 配列Aの各要素の値を機番10で指定されたファイルに書き出す。

C REC=1と指定し、1番目のレコードとして出力する。

```
    WRITE(10, REC=1) A
```

C ファイルを閉じる。

```
    CLOSE(10)
```

```
    STOP
```

```
    END
```

(参考) C

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
/* 定数 mmax と lmax を宣言する。
```

```
   定数 mmax は緯度方向の格子数を、lmax は経度方向の格子数を表す。 */
```

```
int mmax=73, lmax=144, i, iy, m, l;
```

```

/* 配列 a、b を宣言する。 */
float a[mmax][lmax], b[mmax][lmax];

/* ファイルポインタ fp を宣言する。 */
FILE *fp;

/* 配列 a の各要素にゼロを代入する。 */
for (m=0; m<=mmax-1; m++)
{
    for (l=0; l<=lmax-1; l++)
    {
        a[m][l] = 0.;
    }
}

/* データを読みこむ。 */

/* ファイルを開く。
   モードは"rb"(バイナリファイルの読みこみ)を指定する。 */
fp = fopen( "SLP.dat", "rb" );

/* ここからデータを読みこむための for ループが始まる。
   読みこみたいレコードの個数だけ反復する。
   ここでは、1981 年から 2010 年までの 1 月のレコードを読みこむ。 */
for (iy=1981; iy<=2010; iy++)
{

/* レコード番号を計算する。
   ここでは、25 番目 (i=12) から 373 番目 (i=372) までのレコードを読みこむ。 */
i = 12 * (iy-1979);

/* ファイルポインタ fp で指定されたファイルの所定の位置に移動する。
   第 3 引数に SEEK_SET を指定し、ファイルの先頭を位置の基準とする。
   ファイルの先頭から i*mmax*lmax*sizeof(float) バイトだけ前方に移動する。 */
fseek( fp, i*mmax*lmax*sizeof(float), SEEK_SET );

/* 配列 b の各要素の値をファイルポインタ fp で指定されたファイルから読みこむ。
   項目のサイズは浮動小数点のサイズ sizeof(float) を指定する。
   項目の数は mmax*lmax を指定する。 */
fread( b, sizeof(float), mmax*lmax, fp );

```

```

/* 配列 b の各要素を配列 a の各要素に加える。 */
for (m=0; m<=mmax-1; m++)
{
    for (l=0; l<=lmax-1; l++)
    {
        a[m][l] = a[m][l] + b[m][l];
    }
}

/* ここで for ループが終了する。 */
}

/* ファイルを閉じる。 */
fclose( fp );

/* 平均値を計算する。 */

/* 配列 A の各要素をレコードの個数で割る。 */
for (m=0; m<=mmax-1; m++)
{
    for (l=0; l<=lmax-1; l++)
    {
        a[m][l] = a[m][l] / 30.;
    }
}

/* データを書き出す。 */

/* ファイルを開く。
   モードは"wb"(バイナリファイルへの書き出し)を指定する。 */
fp = fopen( "output.dat", "wb" );

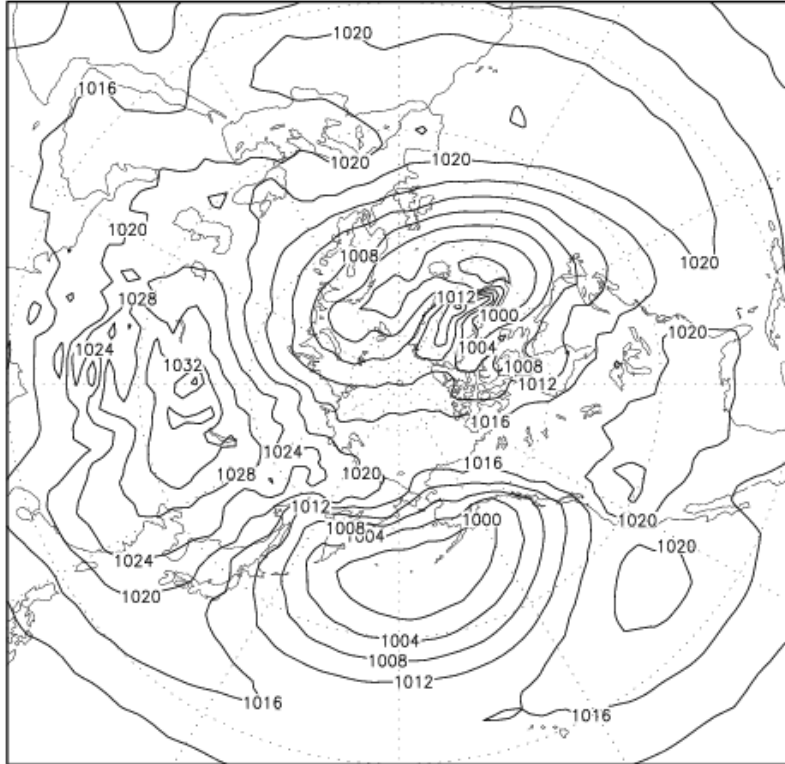
/* 配列 a の各要素の値をファイルポインタ fp で指定されたファイルに書き出す。
   項目のサイズは浮動小数点のサイズ sizeof(float)を指定する。
   項目の数は mmax*lmax を指定する。 */
fwrite( a, sizeof(float), mmax*lmax, fp );

/* ファイルを閉じる。 */
fclose( fp );

```

```
return 0;
}
```

作図例：



ここで、東西平均からのずれを計算してみます。まず、2011年1月のデータを読みこんで、それぞれの緯度において、東西平均（全経度のデータの平均）を計算します。その後で、各経度での値から東西平均の値を差し引きます。

プログラム：

FORTRAN

```

C PARAMETER文で定数LMAXとMMAXを宣言する。
C 定数LMAXは経度方向の格子数を、MMAXは緯度方向の格子数を表す。
  PARAMETER (LMAX=144, MMAX=73)

C 配列A、Bを宣言する。
  REAL A(LMAX, MMAX), B(MMAX)

C データを読みこむ。

C ファイルを開く。
C 機番は10以降の番号を指定する。

```

- C STATUSは読みこみの場合は' OLD' を指定する。
- C バイナリファイルの場合は、FORMIには' UNFORMATTED' を指定する。
- C さらに、ACCESSには' DIRECT' を指定し、
- C RECLにはレコード長をバイト数(整数値)で指定する。
- C 各要素は4バイトなので、レコード長は4×要素数である。

```
OPEN(10, FILE=' SLP. dat', STATUS=' OLD',
+   FORM=' UNFORMATTED',
+   ACCESS=' DIRECT', RECL=4*LMAX*MMA)
```

- C READ文でデータを読みこむ。
- C 機番10を指定する。REC=... でレコード番号を指定する。
- C ここでは、385番目のレコードを読みこむ。
- C 読みこんだデータを配列Aに格納する。

```
READ(10, REC=385) A
```

- C ファイルを閉じる。

```
CLOSE(10)
```

- C 東西平均値を計算する。

- C 配列Bの各要素にゼロを代入する。

```
DO 11 M=1, MMA
   B(M) = 0.
```

```
11 CONTINUE
```

- C ここから東西平均を計算するためのDOループが始まる。

```
DO 21 M=1, MMA
```

- C 配列Aの各経度での値を配列Bに加える。

```
DO 31 L=1, LMA
   B(M) = B(M) + A(L, M)
```

```
31 CONTINUE
```

- C 配列Bの各要素の値を経度方向の格子数で割る。

```
B(M) = B(M) / REAL(LMA)
```

- C ここでDOループ21が終了する。

```
21 CONTINUE
```

- C 東西平均値からの偏差を計算する。



C 配列Aの各要素の値から、配列Bの対応する要素の値を差し引く。

```
DO 41 M=1, MMAX
```

```
DO 42 L=1, LMAX
```

```
    A(L, M) = A(L, M) - B(M)
```

```
42 CONTINUE
```

```
41 CONTINUE
```

C データを書き出す。

C ファイルを開く。

C 機番は10以降の番号を指定する。

C STATUSは書き出しの場合は' UNKNOWN' を指定する。

C バイナリファイルの場合は、FORMIには' UNFORMATTED' を指定する。

C さらに、ACCESSには' DIRECT' を指定し、

C RECLにはレコード長を整数値で指定する。

```
    OPEN(10, FILE=' output. dat', STATUS=' UNKNOWN',
```

```
    +   FORM=' UNFORMATTED',
```

```
    +   ACCESS=' DIRECT', RECL=4*LMAX*MMAX)
```

C 配列Aの各要素の値を機番10で指定されたファイルに書き出す。

C REC=1と指定し、1番目のレコードとして出力する。

```
    WRITE(10, REC=1) A
```

C ファイルを閉じる。

```
    CLOSE(10)
```

```
    STOP
```

```
    END
```

(参考) C

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
/* 定数 mmax と lmax を宣言する。
```

```
   定数 mmax は緯度方向の格子数を、lmax は経度方向の格子数を表す。 */
```

```
int mmax=73, lmax=144, m, l;
```

```

/* 配列 a、b を宣言する。 */
float a[mmax][lmax], b[mmax];

/* ファイルポインタ fp を宣言する。 */
FILE *fp;

/* データを読みこむ。 */

/* ファイルを開く。
   モードは"rb"(バイナリファイルの読みこみ)を指定する。 */
fp = fopen( "SLP.dat", "rb" );

/* ファイルポインタ fp で指定されたファイルの所定の位置に移動する。
   第3引数に SEEK_SET を指定し、ファイルの先頭を位置の基準とする。
   ここでは、385 番目のレコードを読みこむので、
   まず、ファイルの先頭から
   (385-1)*mmax*lmax*sizeof(float)バイトだけ前方に移動する。 */
fseek( fp, (385-1)*mmax*lmax*sizeof(float), SEEK_SET );

/* 配列 a の各要素の値をファイルポインタ fp で指定されたファイルから読みこむ。
   項目のサイズは浮動小数点のサイズ sizeof(float)を指定する。
   項目の数は mmax*lmax を指定する。 */
fread( a, sizeof(float), mmax*lmax, fp );

/* ファイルを閉じる。 */
fclose( fp );

/* 東西平均値を計算する。 */

/* 配列 b の各要素にゼロを代入する。 */
for (m=0; m<=mmax-1; m++)
{
    b[m] = 0.;
}

/* ここから東西平均を計算するための for ループが始まる。 */
for (m=0; m<=mmax-1; m++)
{

/* 配列 a の各経度での値を配列 b に加える。 */

```

```

    for (l=0; l<=lmax-1; l++)
    {
        b[m] = b[m] + a[m][l];
    }

/* 配列 b の各要素の値を経度方向の格子数で割る。 */
    b[m] = b[m] / lmax;

/* ここで for ループが終了する。 */
}

/* 東西平均値からの偏差を計算する。 */

/* 配列 a の各要素の値から、配列 b の対応する要素の値を差し引く。 */
    for (m=0; m<=mmax-1; m++)
    {
        for (l=0; l<=lmax-1; l++)
        {
            a[m][l] = a[m][l] - b[m];
        }
    }

/* データを書き出す。 */

/* ファイルを開く。
    モードは"wb"(バイナリファイルへの書き出し)を指定する。 */
    fp = fopen( "output.dat", "wb" );

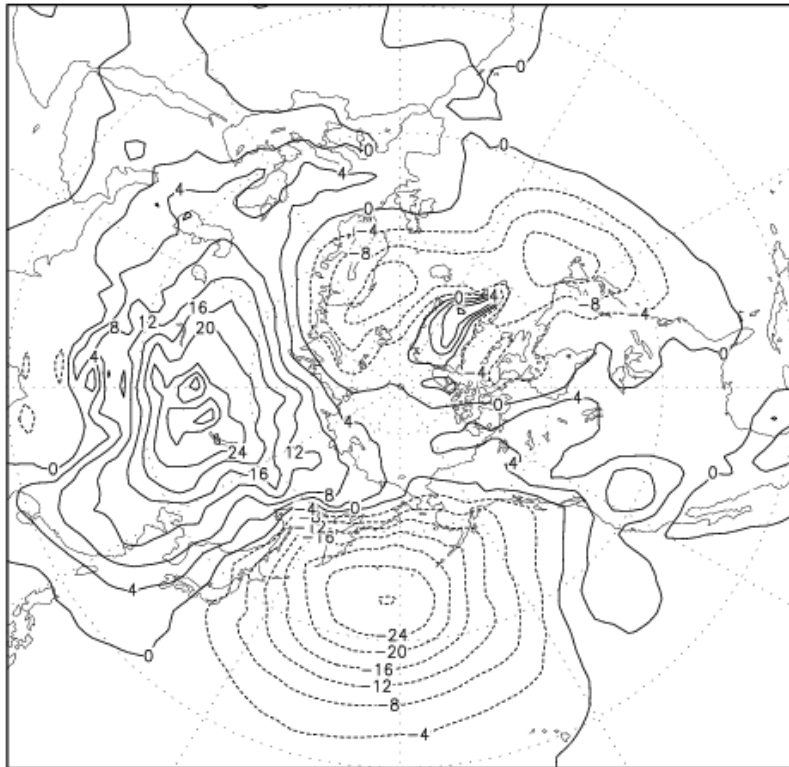
/* 配列 a の各要素の値をファイルポインタ fp で指定されたファイルに書き出す。
    項目のサイズは浮動小数点のサイズ sizeof(float)を指定する。
    項目の数は mmax*lmax を指定する。 */
    fwrite( a, sizeof(float), mmax*lmax, fp );

/* ファイルを閉じる。 */
    fclose( fp );

    return 0;
}

```

作図例：



客観解析データは、数値モデルに観測値を取りこみながら時間発展させて得られた格子点データです。現業の天気予報や学術研究では観測値とみなして利用することも多いですが、厳密には観測値ではない点に注意が必要です。

**課題B**：2011年1月の月平均海面気圧、500hPa面高度、850hPa面気温の偏差を計算せよ。ここでは、平年値（1981年から2010年までの30年間の平均値）からのずれを偏差と定義する。海面気圧の偏差の計算に用いたプログラム（`anomaly.f[c]`）と、海面気圧（SLP.ps）、500hPa面高度（Z500.ps）、850hPa面気温（T850.ps）の偏差を作図した図を提出せよ。なお、作図する領域は北緯20度以北の全経度域とし、ポーラステレオで作図せよ。

※この演習では NCEP/NCAR（米国環境予測センター/米国大気研究センター）による客観解析データを用いています。