Scilab を利用した地質情報の数値解析について

藤井幸泰

深田地質研究所

Numerical Analysis of Geological Information using Scilab

FUJII Yukiyasu

Fukada Geological Institute

要旨：Scilab は理工学分野でのデータ解析，モデリング，シミュレーション，可視化などをプログラミングできるソフトである．フリーウェアでありながら高機能で，作り方次第で高速に大規模な計算を実行できる．写真測量で取得した三次元情報を解析し，可視化まで行った例を二つ紹介する．一つは破断面モデルがベストフィットする平面計算，格子状データへの変換，および粗度（ラフネス）計算である．もう一つはフィルダム模型モデルの格子状データへの変換，時間変化による差分の可視化である．

キーワード：写真測量，三次元点群，データ解析，可視化

Abstract: Scilab is the free software for numerical computation providing a powerful computing environment for engineering and scientific applications. Two attempts, which are related to data analysis and visualization of 3-D information gained by photogrammetry, are introduced here. One is calculations of fracture roughness with the best fit planes and translation to grid data. Second is visualization of topographic difference by fill dam shaking test.

Keywords: photogrammetry, 3-D points, data analysis, visualization

1. はじめに

Scilab は，主に INRIA（フランス国立コンピュータ科学・制御研究所）で開発されたソフトウェアであり，理工学分野でのデータ解析，モデリング，シミュレーション，可視化などをプログラミングできる．2003 年に Scilab の開発は Scilab コンソーシアムに移管され，現在は更に Digiteo に統合され，その 1 部門として Scilab コンソーシアムの活動を行っている（Scilab Web Site）（図 1）．フリーウェアでありながら高機能で，市販ソフトウェア MATLAB のクローンとも呼ばれている．

図 1 Scilab のロゴ

MATLAB も含めて，データの基本構造を行列として取り扱えるために容易であり，コンパイルも不要なインタープリタ方式で動作し，作り方次第では高速に大規模な計算を実行できる．

著者が MATLAB に出会ったのは，スタンフォード大学で 7～8 年前である．応力やひずみ

などの解析・可視化を地球科学の構造地質の授業に利用していた（後にこの授業の内容が教科書 (Pollard & Fletcher, 2005) になっている）．また，MATLAB を利用した宿題も課されていた．当時は英語に四苦八苦しており余裕がなかったのだが，何とか日本語の入門書を入手して MATLAB の基礎勉強だけを行った．著者は学生時代からプログラミング（英語ではたしかタグやコードを組むという表現をしていた）が苦手で，英語との二重苦であった．

帰国後しばらくは MATLAB のお世話にならなかったのだが，写真測量を利用して多量の三次元座標を取得するようになってから，コンピューターによるデータ解析や可視化が必要になった．最初は Microsoft Excel で何とか頑張っていたのだが，多数の地質対象物に，多量のデータ解析を行うには限界があった．そこで MATLAB を利用しようと思ったのだが，予想以上に高価であり，行き着いた先が Scilab である．

この報告では写真測量で取得した多量の三次元情報を解析し，可視化まで行った例をいくつか紹介する．ただし独学で行った内容であるため，教科書のようにはいかない．その点をご了承のうえ読み進めて頂ければ幸いである．

2. 写真測量解析について

2.1 写真測量について

写真測量は写真上で対象物の計測を行う技術である．１枚の写真に含まれるのは二次元情報である．同一対象物を異なる方向から撮影した２枚の写真，すなわち１組の立体写真は三次元情報を含んでいる．写真測量は航空写真による地形測量と共に発達した技術といえる (Linder, 2003)．10 年ほど前からデジタルカメラの普及が急速に進んだ．コンピューターの処理能力も格段に向上し，市販のデジタルカメラで撮影した画像をコンピューターで解析できる，近接デジタル写真測量が盛んに利用されている．実際に地質対象物への適用例もいくつかみられ，様々な成果を上げている (Fujii et al., 2007, 藤井ほか，2006)．

この報告では以下二点の事例を示し，三次元情報の取得に関して簡単に説明を行う．

2.2 一軸引張試験で形成された破断面の写真測量解析

これは藤井ほか（2006）で実施した写真測量解析である．稲田花崗岩を用いた一軸引張試験後の破断面を対象に，化石の接写撮影と同じシステムを用いて立体写真を撮影した（図 2）．写真測量解析には左右カメラの位置と方向の情報が必要である．このため立体写真に標定点と呼ばれる座標既知のポイントを写し込み，標定点から光線が収束する位置を最小二乗法で求めてカメラ情報を計算する．カメラ情報の取得後，左右画像の視差差から破断面上の三次元情報を取得する．この際，破断面の三次元モデルは約 2500 点のポイントで形成され，それらポイントを頂点とする三角形群（Triangle Irregular Network）で構成される．ところでカメラ情報計算時の誤差は 0.15mm 程度であり，三次元点群の精度も同程度と考えられる．

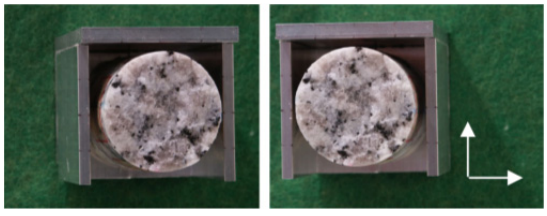

図 2　稲田花崗岩の一軸引張破断面の立体写真
破断面まわりのフレームに標定点が設置されている．矢印は座標を表している．紙面に平行で右方向が X 軸，上方向が Y 軸，紙面に垂直方向が Z 軸である．

2.3　フィルダム振動試験における形状の写真測量解析

これは藤井ほか（2011）で実施した写真測量解析である．振動試験のような移動体の計測には，左右写真（立体写真）の同期撮影が必要である（図 3）．これは藤井ほか（2010）に述べる手法でも可能であるが，今回はストロボを用いた手法を利用した．カメラのリモートスイッチからケーブルを伸ばし，途中でリレーを中継してケーブルを二股にして２台のデジタルカメラに接続した．すなわちリレーの中継により，２台のカメラに同時に撮影信号を送れるようにした．この装置に 1/350 秒間発光のストロボを接続し，左カメラはストロボとシンクロさせ，右カメラはシャッタースピードをやや長くして，ストロボ発光による同期撮影を約 5 秒間隔で行い，合計 300 組程度の立体写真を撮影した．立体写真には破断面と同様に標定点を写し込み，標定点から光線が収束する位置を最小二乗法で求めてカメラ情報を計算した．カメラ情報の取得後，左右画像の視差差からフィルダム模型の三次元情報を取得する．この際，フィルダム模型の三次元モデルは約 10000 点のポイントで形成され，それらポイントを頂点とする三角形群（Triangle Irregular Network）で構成される．合計 300 組の立体写真から厳選し，このような三次元モデルを 10 モデル作成した．ところでカメラ情報計算時の誤差は 3㎜ 程度であり，三次元点群の精度も同程度と考えられる．

3.　破断面の三次元情報の数値解析

前章で取得した稲田花崗岩の一軸引張試験後破断面の三次元情報を数値解析し，破断面の粗度（ラフネス）計算を試みた．

写真測量によって取得された三次元点群はランダムな並びである．またTINによる面構造の平均的な方向が水平を向いているとも限らない．そこではじめに①TINの平均方向が水平を向くように三次元点群を回転移動させた．そして②ランダムな三次元点群から補間法を利用してXY面における格子状の三次元点群へと変換を行い，③格子状データから粗度計算を試みた．

3.1　TINの平均方向が水平を向くように三次元点群を回転移動

三次元点群にZ方向でベストフィットする平面を計算し，これをTINの平均方向とした．そしてこの平面がXY面と一致するように，三次元点群を回転移動させた．さらに三次元点群の中心点が原点になるように，点群を平行移動させた．

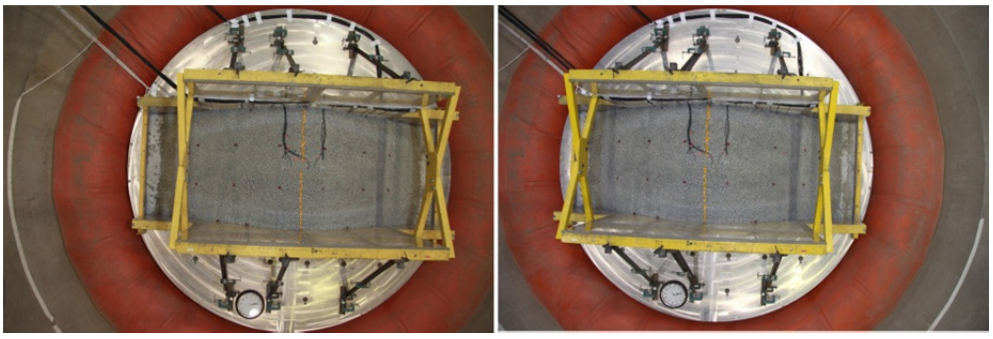

図 3　同期撮影立体写真の例（実験開始前）

3.2　ランダムな三次元点群から補間法を利用してXY面における格子状の三次元点群へと変換

TINによる面構造の平均的方向は水平となったが，三次元点群はランダムな並びのままである．TINの数はおよそ5000個であり，平均的な三角形一辺の長さは0.6mmである．そこでXY面に平行な0.5mm間隔の格子状データを作成し（X：-7～+7mm, Y：-7～+7mm），すべてのZ値は0とした．そして三次元点群のZ値を参考に，シャパード補間法を用いて格子状データのZ値を計算した．三次元点群（○）とシャパード補間法による格子状データを図4に示す．

3.3　格子状データからラフネス計算

破断面の格子状データを用いて，次式で表され粗度を計算した．

二乗平均平方根粗度（*RMS*）

$$RMS = \left(\frac{1}{N} \iint_N Z^2 dXdY \right)^{1/2}$$

中心面平均粗度（*CLA*）

$$CLA = \frac{1}{N} \iint_N |Z| dXdY$$

最大粗度（*MAX*）

格子状データの最大Z値と最小Z値の差

ここで*dX*, *dY*は格子状データの間隔（0.5mm），Zは格子状データのZ値，Nは三次元ポイントの数である．計算結果を以下に記す．

RMS　=　0.46 mm

CLA　=　0.39 mm

MAX　=　2.15 mm

ここまでの一連のプログラムをAppendix 1として論文の最後に添付する．

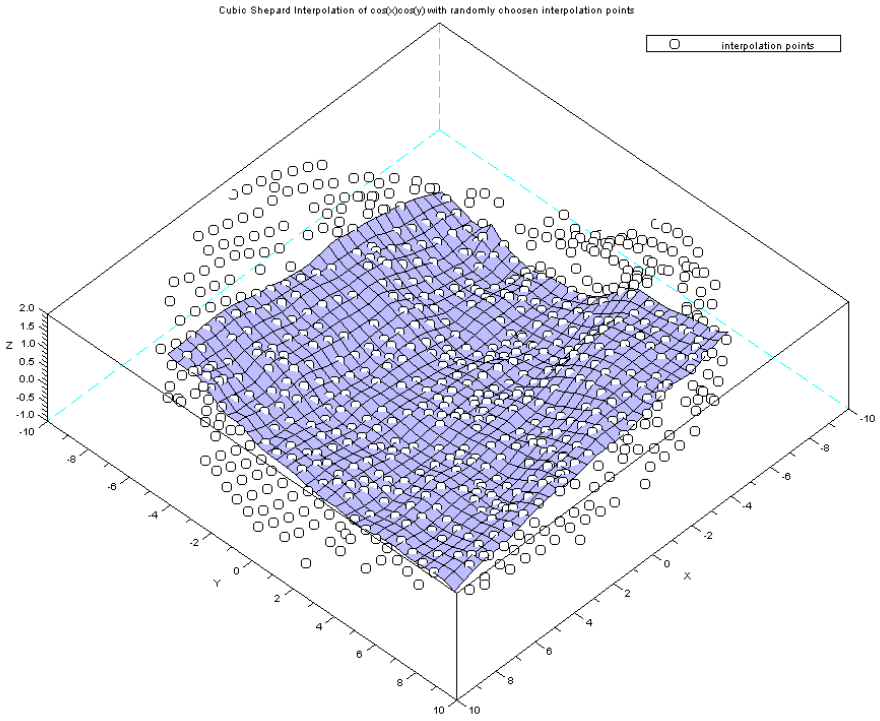

図4　補間法を利用した格子状データの作成
○が三次元点群であり，このデータを参考値にして青い格子状データを計算した．

4.　形状変化の三次元情報の数値解析

前章で取得したフィルダム模型の三次元情報を数値解析し，撮影時間の異なるモデル間の変化解析を試みた．

写真測量によって取得された三次元点群はランダムな並びである．そこではじめに①ランダムな三次元点群から補間法を利用して XY 面における格子状の三次元点群へと変換を行い，②撮影時間の異なる格子状データの差分を計算して可視化を試みた．

4.1　ランダムな三次元点群から補間法を利用して XY 面における格子状の三次元点群へと変換

三次元点群はランダムな並びである．TIN の数はおよそ 20000 個であり，平均的な三角形一辺の長さは 12mm である．そこで XY 面に平行な 10 mm 間隔の格子状データを作成し（X：+0.7～2.8 m, Y:-0.3～+0.7 m），すべての Z 値は 0 とした．そして格子状データの各点から最も近い３点を三次元点群から選択し，格子状データの点からの距離に応じた３点の Z 値を用いて格子状データの Z 値を計算した．すなわち，ある格子状データ点から最も近い三次元点群中の３点の Z 値を（Z_1，Z_2，Z_3），最も近い３点までの XY 面上での距離を（XY_1，XY_2，XY_3）とした場合，ある格子状データ点の Z 値は以下の式で表される．

$$Z = \frac{Z_1/XY_1 + Z_2/XY_2 + Z_3/XY_3}{1/XY_1 + 1/XY_2 + 1/XY_3}$$

これは三次元におけるスプライン補間法と呼べるが，計算結果は破断面の図 4 と似たようなものとなる．

4.2　撮影時間の異なる格子状データの差分計算と可視化

撮影時間の異なる XY 面格子状データの Z 値の差分を計算し，マイナス値（浸食された部分）を寒色系で，プラス値（堆積した部分）を暖色系で示したものを図 5 に示す．どの部分で浸食や堆積が生じているかが明瞭に現れている．

ここまでの一連のプログラムを Appendix 2 として論文の最後に添付する．

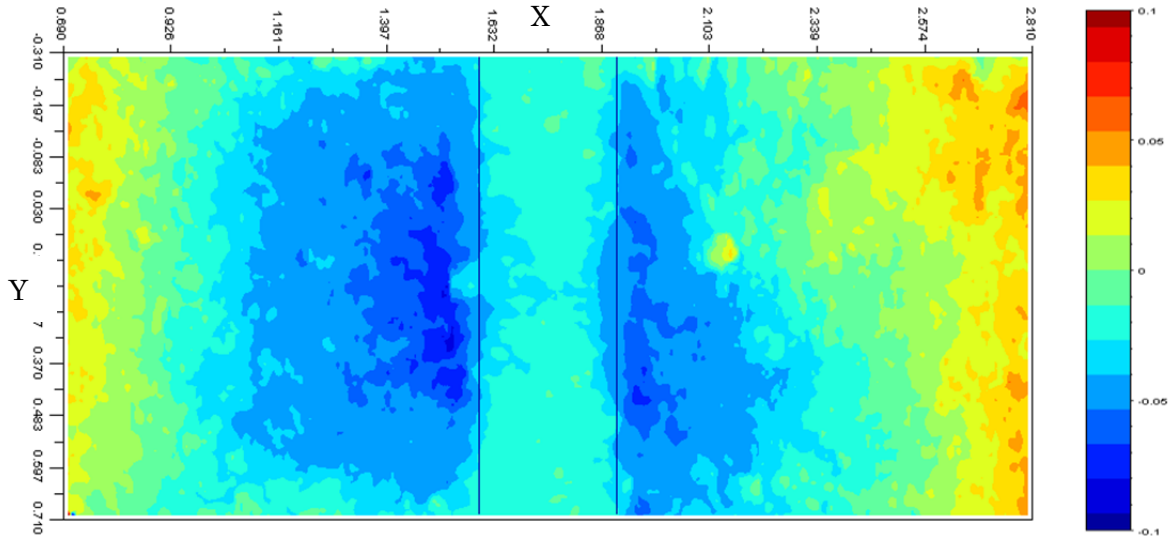

図 5　実験前後のデジタル格子モデルの差分

5. おわりに

地質対象物に写真測量を適用して取得した三次元情報を用いて，Scilab を利用した数値解析と可視化に関して簡単な紹介を行った．Scilab はフリーソフトでありながら，作り方次第で様々な利用ができ，地質工学の分野でも有用である．

謝辞

格子状データに関する補間法に関して，Enrico Segre 博士（Weizmann Institute of Science, Israel）のオープンデータを一部利用させて頂いた．稲田花崗岩の一軸引張破断面試料は高橋学博士（産業技術総合研究所）から拝借した．またフィルダム振動試験は電源開発株式会社の茅ヶ崎研究所で行われたものであり，江原昌彦氏(本社)をはじめ茅ヶ崎研究所の方々にお世話になった．また同期撮影に関しては堀伸三郎氏（防災技術株式会社）にお世話になった．ここに記して感謝申し上げます．

参考文献

藤井幸泰，堀伸三郎，高橋学，竹村貴人，林為人（2006）：デジタル立体写真測量による，稲田花崗岩の異方性と一軸引張破断面粗度のちがいについて，応用地質，47，252-258.

Fujii, Y., Takahashi, M., Hori, S. (2007): Three-dimensional topography of fracture surfaces obtained by a digital photogrammetric technique, International Journal of the JCRM, 3, 29-34.

藤井幸泰，堀伸三郎（2010）：高速移動体（マスムーブメント）の三次元動態計測を目指した同期立体写真撮影システムの開発について，深田地質研究所年報，11，95-103.

藤井幸泰，江原昌彦，堀伸三郎（2011）：フィルダム模型振動試験における同期立体写真撮影と写真測量解析, 第 46 回地盤工学研究発表会, 1095-1096.

Linder, W. (2003): Digital photogrammetry theory and applications, Springer, 189.

Pollard, D.D., Fletcher, R. C. (2005): Fundamentals of Structural Geology, Cambridge, 500.

Scilab Web Site, http://www.scilab.org/

Appendix 1

「破断面の三次元情報の数値解析」

```
chdir('D:¥Application¥SCILAB5¥least_3D')
//(X,Y)data の読み込み
A=read('D:¥Application¥SCILAB5¥least_3D¥te
st.txt',-1,3);
//X 値と Y 値を列ベクトルに変換
X=A(:,1);Y=A(:,2);Z=A(:,3);
l=length(X)
XYZ=[X,Y,Z];
clf(); //clear a graphics window and erase
the associated recorded graphics
subplot(4,1,1);plot(X,Y,'+')
xtitle('','X','Y');
subplot(4,1,2);plot(X,Z,'+')
xtitle('','X','Z');
subplot(4,1,3);plot(Y,Z,'+')
xtitle('','Y','Z');

//最小二乗計算
a0 =[1;1;1]; //a0 を 1 行 3 列の行列 (a1, a2,
a3) とし, これらの初期値を 1 とする
function z = zth1(x,y,a)
z = a(1)*x+a(2)*y+a(3) //z=a1*x+a2*y+a3 の
数式を zth1(x,y,a)と定義
endfunction
function e = myfun1(a,X,Y,Z)
e = ( zth1(X,Y,a) - Z ) //e の数式を
myfun(a,X,Y,Z)と定義
//初期値あるいは仮定値 a と実際の値 Z の差を
e としている
endfunction
// 実際の最小二乗計算
[f1, xopt1, gropt1] =
leastsq(list(myfun1,X,Y,Z),a0)
//list - Scilab object and list function
definition
//fun = list(myfun1,X,Y,Z), この値の二乗が
最小になるように a0 を計算してくれる
//f1 : value of the function f1 = ||fun||^2
at xopt
//xopt : best value of a0 found to minimize
||fun||^2
//grdopt : gradient of f1 at xopt
//xopt(1~3)=a(1~3)となって結果が出る

a1=xopt1(1);
a2=xopt1(2);
a3=xopt1(3);

//各平面に回帰平面の直線を表示
xx=[0:1:10];
yy=(-a1/a2)*xx-a3/a2;
subplot(4,1,1);plot(xx,yy)
zz=a1*xx+a3;
subplot(4,1,2);plot(xx,zz)
yy=[0:1:10];
zz=a2*yy+a3;
subplot(4,1,3);plot(yy,zz)

//行ベクトルを列ベクトルに変換
X=matrix(X,1,1); //1 行 1 列の行列
Y=matrix(Y,1,1);
Z=matrix(Z,1,1);

//座標変換
//z 軸で回転
d1=atan(a2/a1);
D1=[cos(-d1),-sin(-d1);sin(-d1),cos(-d1)];
XY=[X;Y];
XY2=D1*XY;
```

```
//列ベクトルを行ベクトルに
X2=XY2(1,:);Y2=XY2(2,:);
X22=matrix(X2,1,1);
Y22=matrix(Y2,1,1);

//z 軸で回転後のポイントのプロット(X-Z 面でほぼ直線に近くなる)
subplot(4,1,1);plot(X22,Y22,'o')
subplot(4,1,2);plot(X22,Z,'o')
subplot(4,1,3);plot(Y22,Z,'o')

//座標変換
//y 軸で回転->a1,a2 の符号で回転方向が変わる
//a1>0->, a1<0->

//座標変換
//y 軸で回転
ab=sqrt(a1*a1+a2*a2)
if a1 >= 0 then
  d2=atan(ab);
else d2=atan(-ab);
end

D2=[cos(-d2),-sin(-d2);sin(-d2),cos(-d2)];
XZ=[X2;Z];
XZ2=D2*XZ;

//列ベクトルを行ベクトルに
X3=XZ2(1,:);Z2=XZ2(2,:);
X33=matrix(X3,1,1);
Z22=matrix(Z2,1,1);

subplot(4,1,1);plot(X33,Y22,'*')
subplot(4,1,2);plot(X33,Z22,'*')
subplot(4,1,3);plot(Y22,Z22,'*')

//ここからは格子状データへの変換
xyz = [X33,Y22,Z22];
tl_coef = cshep2d(xyz);

// evaluation on a grid
m = 30;
xx = linspace(-7,7,m);
yy = linspace(-7,7,m);
[XX,YY] = ndgrid(xx,yy);
ZZ = eval_cshep2d(XX,YY, tl_coef);

//再び最小二乗計算 ->  この結果はほぼ平面になるはず
//a0 =[1;1;1]; //a0 を 1 行 3 列の行列(a1, a2, a3) とし, これらの初期値を 1 とする
//function z = zth2(x,y,a)
//z = a(1)*x+a(2)*y+a(3) //z=a1*x+a2*y+a3 の数式を zth2(x,y,a)と定義
//endfunction
//function e = myfun2(a,XX,YY,ZZ)
//e = ( zth2(XX,YY,a) - ZZ ) //e の数式を myfun2(a,X,Y,Z)と定義
//endfunction
// 実際の計算
//[f2, xopt2, gropt2] =
leastsq(list(myfun2,XX,YY,ZZ),a0)
//xopt(1-3)=a(1-3)となって結果が出る  ->
すなわち xopt(1)=xopt(2)=0
//aa3=xopt2(3);

//ラフネス計算
RMS=msd(ZZ)
CLA=mad(ZZ)
MAX=strange(ZZ)
```

```
l=1

//元のデータと格子状データのプロット
clf()
plot3d(xx,yy,ZZ)
param3d1(X33,Y22,list(Z22,-9))
xtitle("Cubic Shepard Interpolation of
cos(x)cos(y) with randomly choosen
interpolation points")
legends("interpolation points",-9,1)
xselect()


XYZ=[X33,Y22,Z22];
//write('test_result.txt',XYZ);
```

Appendix 2

「形状変化の三次元情報の数値解析」

```
nb=3
//グリッド xg,yg 各々の点における，周辺 nb 個の実測データ xyzp を利用
//3~10 くらいが良い？
xg=linspace(-0.3,0.7,101);
yg=linspace(0.7,2.8,211);
//xyzp=rand(100,3);
//0->__の値にしなくてはならない？
//(-10,10,100)では dx が零になって計算できない？
//カレントディレクトリの変更
chdir('D:¥Application¥Scilab5')
//(X,Y,Z)data の読み込み
xyzp=read('D:¥Application¥Scilab5¥2_155944_980gal.txt',-1,3);
xset("mark size",3)
clf();
param3d1(xyzp(:,1),xyzp(:,2),list(xyzp(:,3),-9))
//ランダムデータ xyzp のプロット
xp=xyzp(:,1); yp=xyzp(:,2); zp=xyzp(:,3);
np=size(xyzp,1);
if np<nb then error("not enough points to find neighbors!"); end

nx=length(xg); ny=length(yg);
zi=zeros(nx,ny); zj=zeros(nx,ny);

//size of a square where I have probability of finding nb
// datapoint, if datapoints are uniformly distributed over
//  the rectangle xg*yg:
dx=sqrt(nb*(xg($)-xg(1))*(yg($)-yg(1))/np);//xg($)=1,xg(1)=0
//xg-yg の区間 np×nb を sqrt で戻した，平均的距離？
//dx は下の ix のサイズを調整するための値らしい
//even if the grid is nonuniformly spaced, the points are non
//evenly distributed, etc,
//this is a starting size for the shell search

//ここから重要，データの選別，グリッドから近い３点を選んでいる
for i=1:nx
  ix=(xp-xg(i))/dx; nmx=max(abs(ix))
  //すべての xp とある xg(i)との距離，ix は xp と同じ 1 行 12 列
   for j=1:ny
     iy=(yp-yg(j))/dx;  nmy=max(abs(iy))
     //すべての yp とある yg(i)との距離，iy は yp と同じ 1 行 12 列
//search for nb neighbors of (xg(i),yg(j)) by looking in
//  larger and larger cell neighborhoods
      nnb=0;ns=0; nls=0; //(=> first macrocell -> 1x probability)
      while nnb<nb & nls<max(nmx,nmy)
        ns=ns+1; nls=sqrt(ns/4);
        inb=find(abs(ix)<nls & abs(iy)<nls)
        nnb=length(inb)
      end
      ixy=[ix,iy]
//compute the distanceszi between the gridpoint and all the
// candidate neighbors found in the macrocell
```

```
        d2=(xp(inb)-xg(i)).^2 +
(yp(inb)-yg(j)).^2
//選んだ xyzp３点と gridpoint との距離
        zb=zp(inb);//選んだ３点の zp 値
//select the nb closest data points
        [ds,k]=gsort(-d2); ds=-ds;//d2 を小さ
い順にならべた ds
        //k=1,2,3
        //選んだ nb>>nnb となっても（例 3 個以
上になっても), ここで 1～3 番になる
        zb=zb(k(1:nb)); ds=ds(1:nb)//ds と同
じ順番に zb(zp)をならべる
//the grid value is computed as a weighted
mean of the values
// at the neighbor datapoints, with inverse
square distance weights
        if or(ds==0) then
//one or more datapoints could fall exactly
on a gridpoint
           zi(i,j)=mean(zb(ds==0))
        else
          zi(i,j)=sum(zb./ds)./sum(1../ds)
          zbs=zb./ds
          ds1=1../ds
          //距離 ds の重みづけで zb から
zi(i,j)を算出
        end
    end
 end
[XX,YY] = ndgrid(xg,yg);
 //param3d1(XX,YY,list(zi,-9))
 mesh(XX,YY,zi)
 ZS=zi(51,:)
 ZS=matrix(ZS,ny,1);
 write('2_155944_980gal_section.txt',ZS);
clf();
chdir('D:¥201008¥Nenpou_2010¥Chigasaki_1_p
oint&section');
z_2_155944=read('2_155944_980gal_mesh.csv'
,101,211);
z_1_141224=read('1_141224_50gal_mesh.csv',
101,211);

xg=linspace(-0.3,0.7,101);
yg=linspace(0.7,2.8,211);
z_2_155944_141224 = z_2_155944 - z_1_141224;
 z_2_155944_141224(101,1)=0.1;
z_2_155944_141224(101,2)=-0.1;

rect=[-0.3,0.7,0.7,2.8];
isoview(-0.3,0.7,0.7,2.8);
contourf(xg,yg,z_2_155944_141224,16);
xset('colormap',jetcolormap(16));
Zmin=min(z_2_155944_141224);
Zmax=max(z_2_155944_141224);
colorbar(Zmin,Zmax);
Y=ones(101,1); Y16=1.6*Y; Y19=1.9*Y;
plot2d(xg,Y16); plot2d(xg,Y19);
title('2_155944(980gal) -
1_141224(50gal)'); xlabel('X');
ylabel('Y');
```