

付 録

A データ処理の手順

- 1 **area.f90** データ処理の前に、area で P1, P2 の ROI を確認しながら L_0 , $2\theta \pm 2\theta$, $\pm\varphi$ を決定します。
- 2 **move_files.f90** 試験片番号フォルダーに置いて、0 から 1671 までの全.tif ファイルを一気に P1h, P1r, P2h, P2r のフォルダーに仕分けします。
- 3 **MakeTextImage.ijm** P1h, P1r, P2h, P2r のフォルダーの各.tif ファイルを TextImage フォルダーに 0~417 番号ごとの画像ファイル x.i.txt を作成します。これにより fortran90 で処理できるようになります。また、P1h, P1r, P2h, P2r のフォルダを一括処理するマクロも用意しています。
- 4 **p_spots.f90** TextImage フォルダーの下に置き、各位置番号ごとのフォルダーの下に p.txt: ピーク像, x.txt: 処理斑点像, bi.txt: 二値化像, L.txt: ラベリング像, centroid.txt: 重心マーク像および各反転の重心のリスト spot_list.txt が作成されます。
- 5 **viewer.ijm** 作成された画像の p.txt: ピーク像, x.txt: 処理斑点像, bi.txt: 二値化像, L.txt: ラベリング像, centroid.txt: 重心マーク像などのスタックを作成するマクロ。処理結果を確認するのに便利です。一括処理するマクロ (L, org, centroid の 3 像) も用意されています。
- 6 **experimental.f90** TextImage フォルダー下の検出器座標フォルダの spot_list.txt から P1, P2 の選択に対応して、P1, P2 フォルダー下に実験室座標の spot.i.dat を作成します。TextImage フォルダに本プログラムを置くこと、P1, P2 のフォルダは自動で作成されます。
- 7 **DEM_spots.f90** まず、手動で hoop および radial のフォルダーを作成し、それぞれに該当する P1 および P2 フォルダーを移動して hoop および radial のフォルダーに格納してください。コンパイルしたオブジェクト (DEM_spots) は、hoop または radial のフォルダーの下に置いてください。P1 と P2 から各位置の spot.i.dat から対になる直線を求め、回折角、回折座標の結果は、hoop.txt, radial.txt として保存されます。ひずみの計算については、strain.h.dat, strain.r.dat として出力されます。
- 8 **vectors.f90** フォルダー vector を作成し、その中に **vectors.f90** をコンパイルして、実行型ファイル (vector) と、hoop.txt および radial.txt をコピーしておく。vector を起動すると、残留応力のベクトル表示用データとして **e_hp.txt**, **e_hm.txt**, **e_rp.txt**, **e_rm.txt** が作成されます。 $\varepsilon = 10^{-4}$ が、各座標において 1 の目盛になるようにベクトルの長さを決めています。出力したデータは、gnuplot のマクロ **vectors.plt** にて描画できます。
- 9 **map_stress.f90** 手動で map フォルダーを作成し、前述で作成された strain.h.dat および strain.r.dat を map フォルダーに移動します。次に、オブジェクト (map_stress) を起動します。計算された応力ファイル stress_??.dat が作成されます。??には方位と回折位置により、h, r, m(-), p(+) などの文字が当てられます。
- 10 **gnuplot_map.bat** フォルダ map をルートとして、そこに gnuplot_map.bat を用意します。gnuplot を起動し、load "gnuplot_map.bat" を入力すると、各方位、残留

応力マップが一括して作成されます。各残留応力マップをpngなどの汎用フォーマットで保存できます。

B Fortran プログラム

B.1 p_spots.f90

```
!*****
!*                斑点抽出プログラム      p_spots.f90
!   Kenji Suzuki, Niigata University
!   for PILATUS-300K
!*****

program p_spots
  implicit none

  character:: filename*50,dir1*30,file2*50,mkdir*30
  integer:: i,j,k,l,m,n,w,h,cnt,np,px(200),py(200),sm,nspot
  real*8:: x(0:2000,0:2000),xx(0:2000,0:2000)=0.0
  real*8:: p(0:2000,0:2000)=0.0 ! ピークフィルタの画像
  integer:: bi(0:2000,0:2000)=0 !2 値化画像
  integer:: pi(0:2000,0:2000)=0 ,istart, istop, loop
  real*8:: dx(0:2000,0:2000)=0.0,dI(0:2000,0:2000)=0.0,xf,yf,xyf,yxf
  real*8:: s,threshold

  w=486;h=618 ! 画像サイズ                w=0:486 h=0:618
  m=3 ! ピーク抽出セグメントのサイズ全幅 奇数 一般には m=5
  n=(m-1)/2 ! 差分幅

  threshold=100.0 ! ピーク検出閾値   この値で斑点検出の閾値を下げると、斑点検出の数が増える

  call SYSTEM('ls')

  print *, "File x_???.txt の開始番号と停止番号 istart, istop を入力"
  read(5,*) istart, istop

  do loop=istart, istop
    write(dir1,'(I5)') loop
    mkdir="mkdir "//trim(adjustl(dir1))
    print *, mkdir
    call SYSTEM(mkdir) ! ファイル番号の dir の作成

    filename="x_"//trim(adjustl(dir1))//".txt"
    print *,filename

    !原画像の読み込み
    open(unit=11,file=filename,status='old')
    do j=0,h
```