PROGRAM: Laue_Orient

Author: J. Rodriguez-Carvajal (ILL)

Created: July-August 2010, Updated: November 2014

This program performs a peaks search and rough orientation of a series of Laue patterns.
The program needs as input files either the Laue images (TIFF files) or peak files (generated by Esmeralda). In the case the user provides Laue's images, the program performs a peaks search before trying to get the orientation matrix. The first image serves for obtaining the initial orientation matrix and from the spindle angles of the following images (or peaks files) the orientation of each image is set up approximately.

The input CFL file (see **Esmeralda** manual for more information on CFL files) should contain at least the following keywords: **`CELL`**, **`SPGR`**, **`IMAGE_FILES`** or **`PEAKS_FILES`**, **`PEAKFIND`** (needed for searching peaks in images), **`INDEXING`** and **`INSTRM`**. An optional instruction **`ADJUST_SPINDLE`** instructs the program that the output spindle in the indexed peaks files is slightly modified with respect to the nominal spindle value. This is needed when a large number of images have to be treated; a sequential order of the spindle angles is suggested in this case.
As usual, the items are not case sensitive (except name of files in UNIX systems), free format, blank lines are ignored, lines starting with "!" are comments, etc.
The program needs additional input files: TIFF files for images (or peaks files) and the instrument file containing the instrument parameters and excluded regions in the detector.
An appropriate CFL file can be automatically generated by **Esmeralda** after getting the orientation matrix from a single image. **Laue_Orient** can be invoked from Esmeralda to treat all the images contained in a directory.
The program can be invoked in the command line followed by the code of the CFL file:

```
My_Prompt> Laue_Orient my_CFL_file  <cr>
```

The file `my_CFL_file.cfl` should exist in the directory in which the program is invoked.
If no argument is provided, the program asks for the code of the CFL file.
If the run is successful, three output files are generated.

`my_CFL_file.idx`	:	General output file containing information about indexing
`my_CFL_file_peaks.inf`	:	File with the whole set of indexed peaks of each image (input for refinement)
`my_CFL_file_ref.cfl`	:	Template file for refinement of orientation, offsets and distortions. The VARY instructions are commented. The file is generated only if it does not exist. This allows further runs without destroying user's modifications.

Input file: Examples of CFL files adapted for this program are given below

Example 1 (TIFF files provided)

```
Title  fel_data
!        a         b        c     alpha    beta   gamma
Cell   17.2596  11.989  10.077    90.0   116.140   90.0
!     Space Group
Spgr  C 2/m
IMAGE_FILES  3
fel_23.tif
fel_22.tif
fel_24.tif
END_IMAGE_FILES
PEAKFIND   !Selection of Peak_Find_Threshold (modified SNAIL subroutine, by Clive Wilkinson)
!   (three set of numbers because we have three images) Peak-pixel=> I > Cutoff*Sigma(bgk)
! CutOff BlockSz  Min.Area Min.Dist
     15     30         9        5
     15     30        9        5
     15     30        9        5
INDEXING                             !Orient_cond_type:
!ang_Type coprime  nmax_ref nmax_sol norient maxind exhaust angtol  dmmtol ang_min ang_max
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0
INSTR  vivaldi.inf
```

Example 2 (Set of peaks files provided)

```
Title  fel_data
!        a        b        c     alpha    beta   gamma
Cell   17.2596  11.989  10.077   90.0   116.140   90.0
!     Space Group
Spgr  C 2/m

PEAKS_FILES  3
Ex_pk_22.dat
Ex_pk_23.dat
Ex_pk_24.dat
END_PEAKS_FILES

INDEXING                             !Orient_cond_type:
!ang_Type coprime  nmax_ref nmax_sol norient maxind exhaust angtol  dmmtol ang_min ang_max
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0

INSTR  vivaldi.inf
```

Example 3 (Set of image files provided with LFM algorithm for peak search)

```
Title  fel_data
!        a        b        c     alpha    beta   gamma
Cell   17.2596  11.989  10.077   90.0   116.140   90.0
!     Space Group
Spgr  C 2/m


IMAGE_FILES  3    ! Min.Dist  Min.Slope Min.Area
fel_23.tif           5          0        6
fel_22.tif           5          0        6
fel_24.tif           5          0        6
END_IMAGE_FILES

PEAKFIND   !Selection of Peak_Find_Threshold (modified SNAIL subroutine, by Clive Wilkinson)
!  (Three set of numbers because we have three images) Peak-pixel=> I > Cutoff*Sigma(bgk)
! CutOff BlockSz  Min.Area Min.Dist
     15     30        9        5
     15     30        9        5
     15     30        9        5

INDEXING                             !Orient_cond_type:
!ang_Type coprime  nmax_ref nmax_sol norient maxind exhaust angtol  dmmtol ang_min ang_max
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0
XYZ       .false.      80       20      5      3       0     2.5      1.5    30.0   150.0

INSTR  vivaldi.inf
```