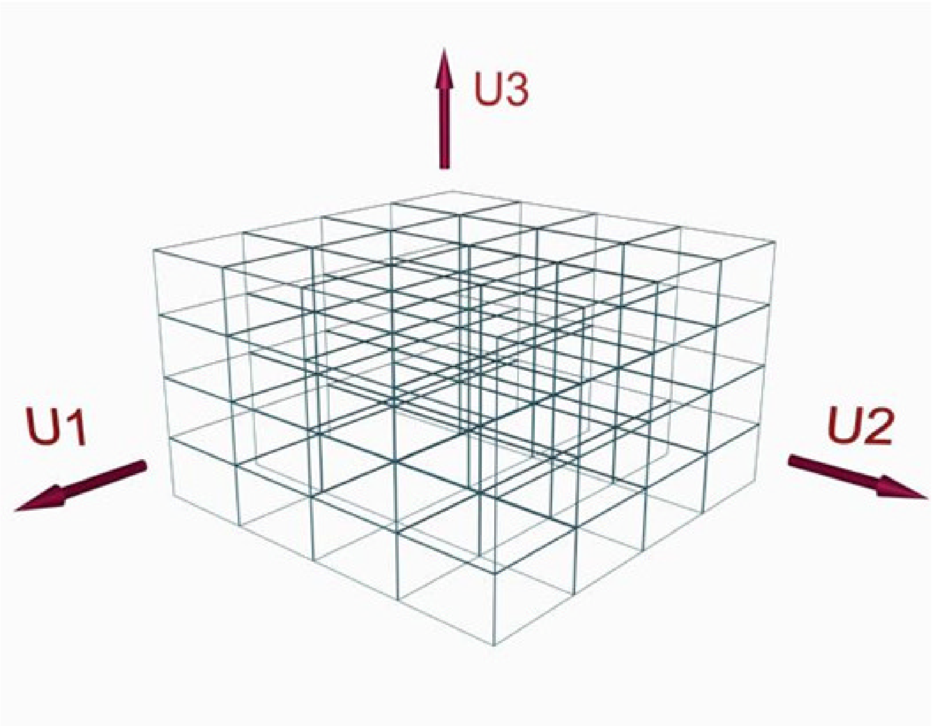
# Multiresolution Algorithm

Mslice Visualization Software add-on for Single Crystal with Position Sensitive Detectors.




Ibon bustinduy

14-11-2005

**Installation:**

Programme files are a collection of Matlab functions (source code ASCII *.m files). The .m source code should be portable between windows, UNIX/Linux platforms (such as different Linux OS, and Apple MacOs X).

```
ms_MApow.m
multires2d.m
MApowfromwindow.m
micall_MApow.m
ms_updatelable_MApow.m
msmapow.m
plot_MApow.m
```

1. Copy all files to a directory on the hard disk, for example *'c:\mprogs\mslice\MApowder'* or `'`*/home/ibon/mprogs/mslice/MApowder*`' in case of Unix OS.`

2. Include this directory to the Matlab path: **`File > Set Path… > Add Folder…`** and select the folder where the .m files are stored ( for example: *'c:\mprogs\mslice\MA'* in windows, or *'/home/ibon/mprogs/mslice/MA'* in Unix OS.). If you do not want to change path you also can change matlab directory **`cd('c:\mprogs\mslice\MA')`** or **`cd('`*/home/ibon/mprogs/mslice/MA*`')`**and therefore routines will be visible from Matlab.

**How to use it:**

The author of this document, considers that the reader is a Mslice user, and therefore familiarized with mslice manual, therefore basic concepts will not be revised. The package usage is (by now) centred in single crystal using an instrument equipped with Position Sensitive Detectors. The process of visualization will require the user to

1. Load parameters from Mslice control window,
2. Load Data
3. Calculate Projections.

Once these steps had been taken, instead of plotting the colour map, by means of clicking on "Display" button, the user can type in Matlab prompt the command:

```
>> ms_MApow
```

Resulting in a new control window, where the data set constructed from Mslice has been loaded, as well as information concerning slice limits parameters; such as limits. Where two new parameters: number of Levels "NLEVEL" and noise to signal ratio "ERR/S" are added.

NLEVEL: Specifies the number of "levels" the algorithm will cover, that is, if we state, that the step in $U_1$ axis is given by dx, the $U_2$ axis step by dy and the $U_3$ axis step by dz. The algorithm will go collecting counts from a bin scheme of [dx, dy] = [1, 1]. Up to a bin scheme given by [Dx, Dy]. Where Dx = dx.*(2.^(NLEVEL-1)) , Dy = dy.*(2.^( NLEVEL -1)), doubling the size of the bins as the Level grows. The reason is because we are going to perform this multiresolution technique in the detectors-energy space, to make sure we don't have problems with the boundaries of the detectors, we simply consider intensity matrix as the input data therefore the maximum resolution is when we move in steps of 1 'pixel'. After this we will be able to plot our data sets in the reciprocal space.

ERR/S: will be used as a threshold, where a bin will be considered as a valid collection of counts as long as its noise to signal ratio stays below the given threshold; varying it from 0 to 1. For example, if we choose a ERR/S = 0.1 we are being very restrictive, forcing counts to be collected in larger bins in order to be bellow the threshold.

Viewing axis as well as their limits can be established from the **Mslice Control Window.**

By default those two parameters will be 1. (Number of levels = 1, means no *multiresolution*, and noise to signal ratio = 1, means all bins which ERR/S is bellow 100% will be accepted). We also can fix any of the given axis, performing an equal=width binning in this axis, leaving the rest of the axis performing this *multiresolution* algorithm.

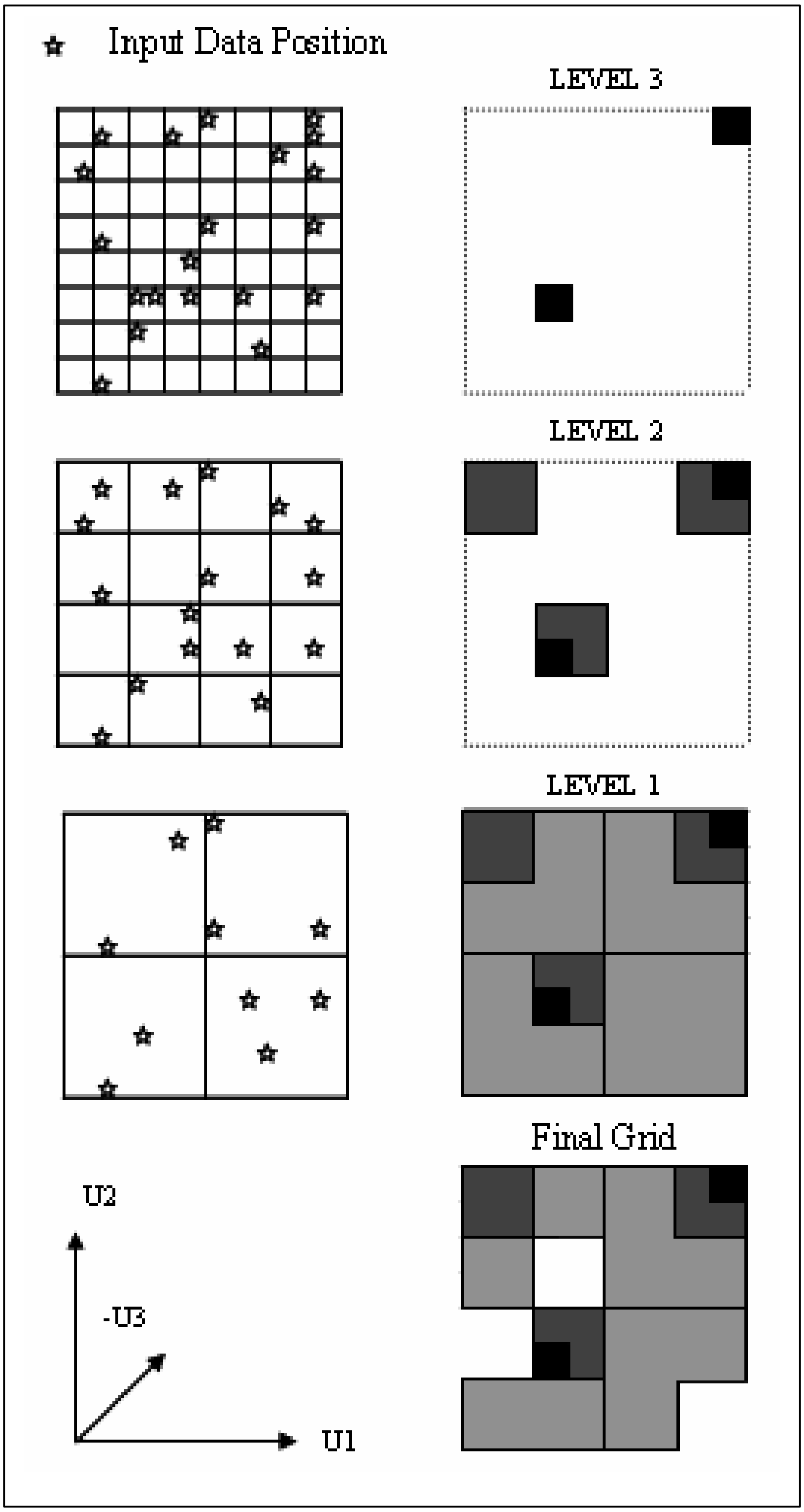



Fig.1. To illustrate the multiresolution algorithm consider the 2D case, where the number of levels (NLEVEL) is equal to 3, and the resolution pattern has chosen to be determined by the Level 2 binning size. In the multires3d implementation for the sake of simplicity the resolution pattern related to instrument detectors is considered to be fixed to the highest level, correspondent to

Those parameters can be modified from the *'MSlice: MULTIRESOLUTION powder'* window:

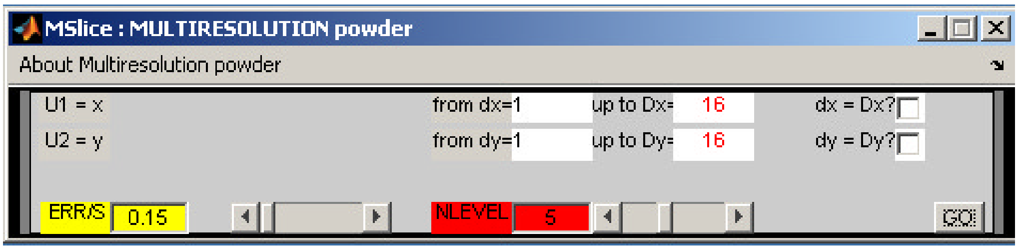



Fig. 2. Represents MULTIRESOLUTION powder, input window, we can type in all parameters to construct the required colour maps. Once we pushed GO button *a colour map* will show up.

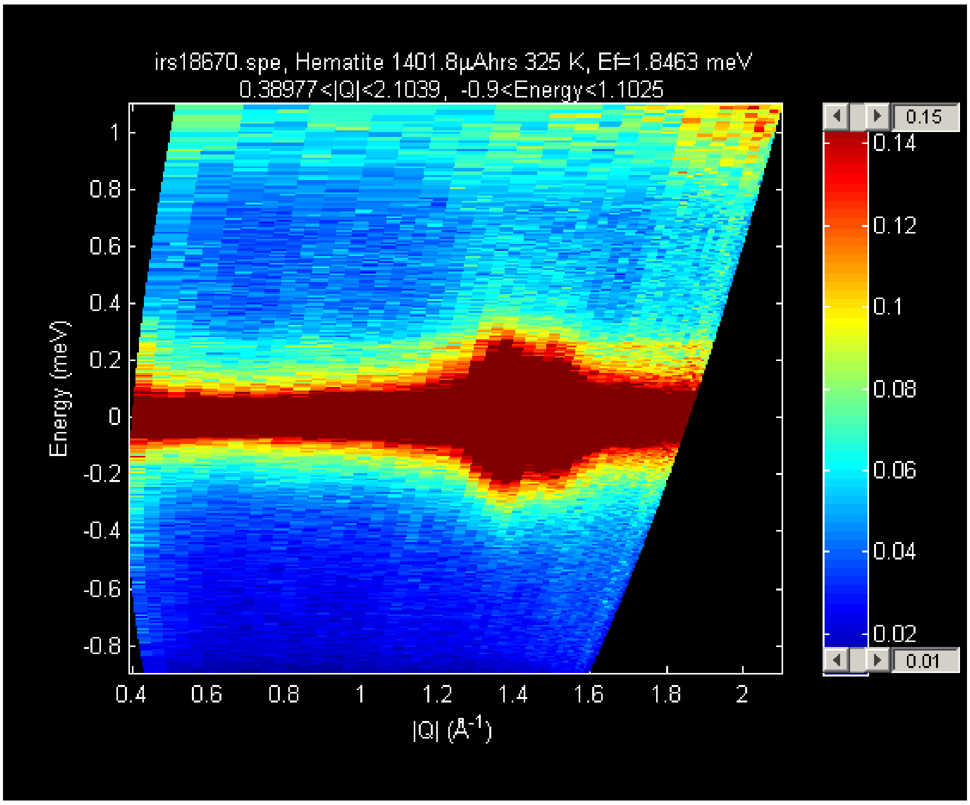



Fig. 3. Represents a colour map from IRIS instrument.

**Limitations**

- Main Question is what to do with those counts, that even in the Larger bins `LEVEL` they do not satisfy the imposed threshold by the user?, one solution is call them ‘rejected counts’, and plot them using the maximum bin size scheme. One way of distinguish them is by plotting them: like this:

  The total number of counts that have not been collected below the threshold will be given by:

  ```
  >> MA_d=MApowfromwindow;
  >> MA_d.L_T(1)
  ```

**Other remarkable functions**

**`MApowfromwindow.m`** extracts MA_d from `MULTIRESOLUTION 3D` Window into the command line.

Shape of the structure we will obtain, where the multiresolution volume is stored:

```
MA_d =

    intensity: [164x64 double]
        error: [164x64 double]
          L_T: [4x1 double]
```

For more information, type in the MATLAB command window:

```
>> doc multires2d
```