Flatcone data treatment with MATLAB - The one-page quick reference guide

- To open a figure window and plot Flatcone data, use the command **`plotmultiple`.**
  Most operations are then accessible from the menu of the plot window.

  Example:

`plotmultiple 012345`	*to plot file 012345*
`plotmultiple 0123[45,46,47,50]`	*to combine files 012345, 012346, etc. and plot them*

  A number of options can be set in the file options.m.
  To make changes, copy this file to your directory and type `edit options`.

- To manipulate data from the command line, you need to store the data in a workspace variable:

  ```
  data = getfiguredata(fig);
  ```

  fig is the number of the figure window (takes active window, if left empty).

- To plot data from a workspace variable, use

  ```
  fcplot(data);
  ```

- To smooth the data, use

  ```
  smdata = smoothdata(data,range);
  ```

  where range defines the length scale to smooth (each point replaced by an average over a region of this size).
  Example: `smdata = smoothdata(data,[.5,.5]);` (See note below)

- To subtract one data set from another, use

  ```
  diffdata  = subtractdata (data1, data2, [mode], [opt]);
  ```

  Result: diffdata = data1-data2

  The parameter *mode* defines what happens if the coordinates in *data2* are not exactly the same as in *data1*.

  Example:

`subtractdata(data1,data2);`	*Takes nearest point in data2 and subtract from data1.*
`subtractdata(data1,data2,'nearest',[.1,.1]);`	*Same, but maximum allowed distance is [.1,.1]*
`subtractdata(data1,data2,'interpolate');`	*Interpolates data2 to determine the value to be subtracted.*
`subtractdata(data1,data2,'range',[.1,.1]);`	*Averages data2 over the range [.1,.1]*

  For `smoothdata` and `subtractdata`, please note:
  By default, the coordinates of the data points are the angles a4 and a3 (resp. psi). The ranges that you give refer to these angles. If you want to give the ranges in reciprocal Angstrom (Q-space), transform the data using

  ```
  datanew = coordtransform(data, ‘qxy’);
  ```

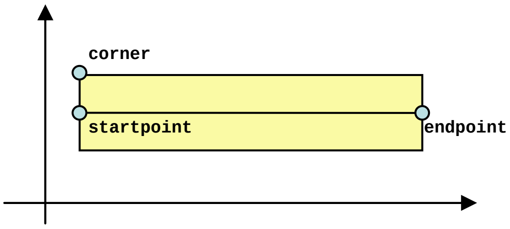
- To multiply the data by a constant:

  ```
  scaleddata = scaledata (data, factor);
  ```

- To integrate over a certain region:

  ```
  [x,y,dy,H,K,L] = integrateqxy(data, startpoint, endpoint, corner, npoints);
  ```

  The values for *startpoint*, *endpoint* and *corner* are interpreted as (H,K,L) if they are three-element vectors, and as (Qx,Qy) in Å^{-1} if they are two-element vectors. *npoints* defines the number of points in the output.


corner
startpoint
endpoint