|  |  |
| --- | --- |
| **nnfData**: | Structure to save data and process data |
| **nnfPlot**: | Plot handle to produce |
| **nnfDataCol**: | Unit structure to save data |
| **nnfDataSet:** | Dataset to save data from file |

**Instructions**

**Introductions**

**nnfData** is a MATLAB class to save data from experimental results / processed data and can be used to manage plots.

**nnfPlot** is a MATLAB class to create publication quality figure, with features to change characteristics of figure, to save .fig file (reload next tiem), export .eps file.

**nnfDataSet** is a MATLAB class to create set of data directly read from file

**nnfDataCol** is the data structure unit to load data that can be further processed by other functions.

# **Installation**

copy nnfplot.m, nnfplot\_DataCol.m, nnfplot\_Data.m, nnfplot\_Plot.m to MATLAB path.

# **Usage**

### nnfData: comprised of (raw) data and processed results

|  |  |
| --- | --- |
| **Create new object** | hdata = nnfData(*filename*) |
| **Add data** | hdata.addData(*dataSetHandle*) |
| **Remove data** | hdata.removeData(*dataSetHandleNum*) |
| **Replace data** | hdata.replaceData(*dataSetHandleNum, dataHandle*) |
| **List data** | hdata.listData() |
| **Add results** | hdata.addRes(*dataColHandle*) |
| **Remove results** | hdata.removeData(*dataColHandleNum*) |
| **Replace results** | hdata.replaceData(*dataColHandleNum, dataColHandle*) |
| **List results** | hdata.listRes |
| **Save library** | hdata.saveLib() |

### nnfPlot: notice that you need to use nnfDataCol to create the data

|  |  |
| --- | --- |
| **Create new plot** | hPlot = nnfPlot(‘*filename.fig’*)  or nnfPlot(*yDataColHandle*)  or nnfPlot(*x1DataColHandle,*  *y1DataColHandle, …*  *x2DataColHandle, …*  *x1DataColHandle, …*)  or nnfPlot(tableData, …  xNum, … % # of column to  % be x  yNum) % # of column to  % be y, can be  % an array |
| **Add plot** | hPlot.addPlot(*xDataColHandle,*  *yDataColHandle, …*  *[optional] linewidth, …*  *[optional] lineStyle*) |
| **Add plot from table** | hPlot.addTable(*tableData, …*  xNum, … % # of column to  % be x  yNum) % # of column to  % be y, can be  % an array |
| **Add errorbar** | hPlot.addError(*plotNum*, …  [1, 2, 3, 4, …]) |
| **Add reference line at y=4** | hPlot.addRefLine(@(x) 4) |
| **Add fit line and plot reference line** | hPlot.addFitLine(*PlotNum, …*  *@fun(x, xdata), …*  *x0, …*  *[optional] xrange*)  % using lsqcurvefit  or  hPlot.addFitLine(*PlotNum, …*  *PolyfitNum, …*  *[optional] xrange*)  % using polyfit |
| **Set figure properties** | Setting of properties are realized by directly inputting:  hPlot.BoxDim = [6, 4]  hPlot.LineWidth = [1, 2]  hPlot.LineStyle = {‘-’, ‘--’}  hPlot.Markers = {‘none’, ‘\*^’, ‘s’}  % ‘\*’ prefix means filled  hPlot.XLim = [1, 10]  hPlot.YLim = [2, 20]  hPlot.XLabel = ‘x’  hPlot.YLabel = ‘y’  hPlot.Title = ‘Title’  hPlot.Legend = {‘a’, ‘b’}  hPlot.typeFig = ‘linear’ |
| **Export to eps** | hPlot.export(*fileName*) |
| **Save to fig** | hPlot.saveToFig(*[optional] fileName*)  The fig is saved to the same folder |

### nnfDataCol:

|  |  |
| --- | --- |
| **Create new data column**  where:   * vName: string of variable name, used in plot label * unit: string of unit, used in plot label * data: 1xN or Nx1 vector or table | hDataCol = nnfDataCol(vName, unit, data) |
| **Read data column** | Directly input   * hDataCol.vName * hDataCol.unit * hDataCol.data |

nnfDataSet: independent, of other programs, and will leave for later.

### Current problems to be solved (to be solved in next version)

* Line with skipped point, once reopened, cannot retrieve original data
* Reorder legend more smoothly
* Batch change option

Version information:

Alpha 1.0.0:

* General structure

Alpha 1.0.1

* Bugs fixed
* Add new functions
  + nnfPlot
    - allHideLines
    - markersRandom
    - allErrorProps(cap\_size, line\_width)
    - removePlot
    - reorderPlots(*permutation of 1-# of plots, e.g. [1, 4, 2, 3]*)
    - batchCol(plotNum, cols)
* Improve
  + nnfPlot
    - AddFitLine: report equation
    - LineCount: more information
  + nnfDataCol