

# (1) iFit : a Matlab-based data analysis framework

*E. Farhi* <[farhi@ill.fr](mailto:farhi@ill.fr)>

***iFit***  
***v1.0.4***

*Use scientific data in a glimpse of an 1/2 eye*

**Import** any data set from most file formats.

**Display** its content without hassle, in camera-ready plots.

**Manipulate** the data set like a piece of ... clay.

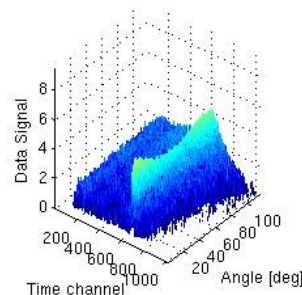
**Apply mathematical** operators seamlessly.

**Save** the data sets in many usual file formats.

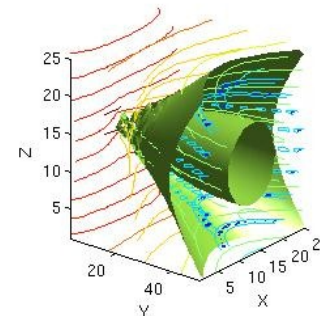
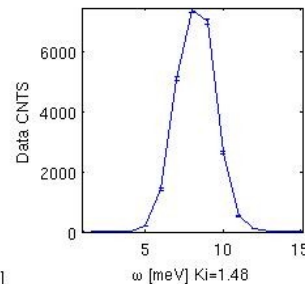
**Fit** the data to any model function, including your own.

***Find bugs and win a free license !***

File ILL\_IN6.dat ILL Data (normal integers)



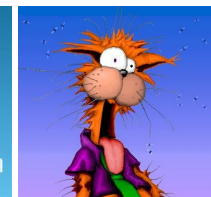
UPI3;File sv1850.scn ILL TAS Data



**Cost:** 0€ + smiles (free-style).

**Contact:** <farhi@ill.fr>

<ifit.mccode.org> ILL/DS/CS



The existing data analysis software I know do not satisfy me.

## What I do not like:

- Heavy user interface.
- Hard to maintain (short self life-time).
- Monolithic software design.
- Too complex software design.
- Limited functionalities.
- Non-extensible functionalities.
- Not generic software.
- Improper/unknown mathematical operator handling.
- Incomplete documentation.

*So I tried to avoid all these pitfalls ...*

## History and development

**First release** 1.0 in August 2011, but designed 2 prototypes since 2003.  
Actively developed since 2009.

**Development team:** me

Beta-testers and friendly users: *P. Willendrup, L. Udby, H. Jacobsen.*

**Licence:** GPL2. ILL software.

**Funding:** none (10% of my time).

**Project size:** Core is about 25 kLOC. Total 55 kLOC with docs and contrib.

**Stability:** good (most features are implemented and functional).

# Download and installation



**Requirements:** Matlab (any version since R13 - 6.5) – no need for toolboxes.

**Obtain** the package at <http://ifit.mccode.org> and extract it. It is free (GPL).

**Start** Matlab and type:

```
>> addpath(genpath('/location/to/iFit'))
```

**Check** that it works, and open the documentation:

```
>> doc(iData)
```

```
>> methods(iData)
```



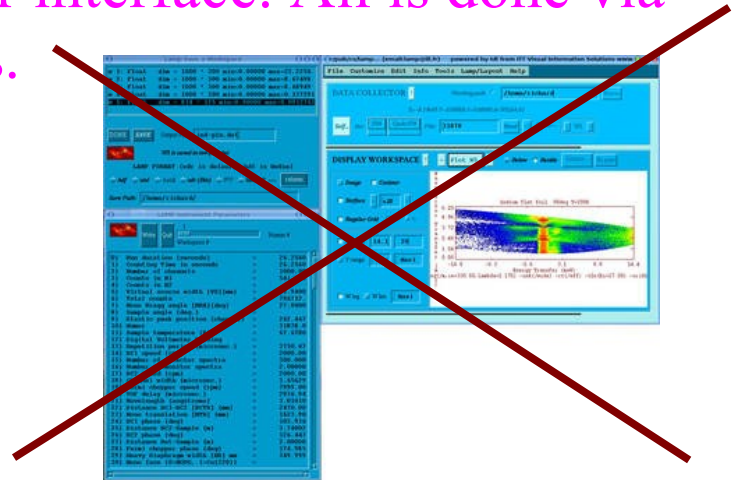
## What can iFit do for you

- ▶ Handle data sets typically up to ~100 Mb.
- ▶ Import/export any text-based data file (*load*).
- ▶ Import/export a variety of binary data files (*saveas*).
- ▶ Display data sets (*plot*).
- ▶ Apply mathematical operators (50+) directly on data sets (+-/\*).
- ▶ Carry signal, axes, error bars, ... along through operations
- ▶ Optimize any multi-parameter problem.
- ▶ Fit model function onto data sets (*fits*).

# What can iFit *NOT* do for you

iFit does not contain any advanced graphical user interface. All is done via scripts, except a few dialogues and plot windows.

Mfit/MView like interfaces may be designed in the future



iFit can hardly handle 1Gb+ files.

A faster file handling system may be added for large files.

iFit is not very fast.

Yes, but it does the job right (I hope) and this is more important to me.

## The future of iFit, what *YOU* can do for iFit

iFit does not yet support specific data analysis routines, adapted to e.g. neutron scattering, X-rays, light spectroscopy, NMR, ...

To be done by developers and the user community.

iFit uses Matlab, a commercial software.

Ports to Octave and Python are envisaged.

iFit has a wide, but still limited number of supported file formats.

Additional formats can be easily implemented.

iFit needs testing as it is a *baby* software.

The user community is essential. Please report bugs and suggestions.

Use [ifit-users@mccode.org](mailto:ifit-users@mccode.org)



## A reminder about Matlab: numerical data

Matlab variables are scalars, vectors and matrices usually containing numerics

Operator	Symbol	Syntax	Example
<i>Assignment</i>	=	a = b (assign b to a)	a = 5; b=[2,3,4];
<i>Addition</i>	+	a+b	c=b+5;
<i>Subtraction</i>	-	a-b	c=b-5;
<i>Multiplication</i>	*	a*b or a.*b	c=b*3 ; c=a.*b;
<i>Division</i>	/	a/b or a./b	c=b/3 ; c=a./b;
<i>Power</i>	^	a^b or a.^b	c=b^3 ; c=a.^b;

In addition, Matlab supports character arrays, 'structures' (name/value pairs), and 'cells' (array/list of other objects).

## A reminder about Matlab: special characters

The `'%'` character indicates comments.

The `' '` character is a separator, just like spaces.

The `'[ ... ]'` indicates matrices (of numbers, characters, ...).

The `'[]'` is the empty matrix

Character strings are specified with the single quote character: `'blah'`

The `';` indicated that we switch to the next line, without showing the content:

```
a = [ 1 ; 2 ]; % a column vector
```

Commands which do not end with `;` print their result.

The `':'` colon character indicates a range:

```
a = 2:10;
```

Subset of a matrix can be obtained using the `'()'` indexing:

```
a = [ 1,2,3,4,5 ]; b = a(2:4);
```

Matrix indexing is done with (rows, columns) formalism:

```
a = rand(5,5); b=a(2:4, 3);
```

The transposition operator is a single single-quote `'`

```
a = rand(5,5); b = a';
```

# A reminder about Matlab: common operators

Syntax	Description
load('file.mat'); openfig('file.fig')	Load a Matlab file or figure
save('file.mat');	Save all variables into a Matlab file
size(a)	Display the object size (rows x columns)
min, max, mean, std, log, exp, cos, fft, conv, sqrt, ...	Math stuff
sum, prod	Summation, product
Interp1, interp2	1D and 2D interpolation
plot, surf	1D and 2D plots

To get help about any function/command/method :

>> **help** *method*

>> **doc** *method*

## A reminder about Matlab: execution control

To control the code execution, one can use :

<b>for</b> index=vector ... <b>end</b>	Indexed loops
<b>while</b> condition ... <b>end</b>	Conditional loops
<b>if</b> condition ... <b>elseif</b> condition ... <b>end</b>	Tests

Conditions can use relational and logical operators:

&	AND
	OR
~	NOT ( <i>not</i> !)
< <= == ~= >= >	comparisons



Try it, find bugs and help us all