(5) iFit/Math:

« One Class to do some Math »

God damn it ! Just compute it !

Math: perform mathematical operations seamlessly

As we have seen there is a **unique, simple way** to

- Import data into objects: load
- Plot object contents: plot
- Save objects: saveas

So we expect to also be able to do mathematical operations without hassle.
All **common Matlab math commands** have been ported for use with iData.

This includes:

- Unary operators (cosine, ln, ...),
- Binary operators (+-/*, combine, power, ...),
- Interpolation, union, intersection,
- FFT, convolution,
- Derivative, integrals,
- Subset indexing and object parts extraction, object appending

Math: Overview

The mathematical operators can handle objects of ***any dimensionality and size.***

The *Signal, Axes, Monitor, Error* bars are used. Check definitions, and adapt them with:

- setalias, getalias (alias definitions, including Signal, Error, Monitor)
- setaxis, getaxis (axis definitions)
- set, get (values)

The ***Axes* ranks** are defined as 1 → rows, 2 → columns, 3 → pages... 0 → *Signal/Monitor*.

All operators propagate the *Error* bar, and can handle objects with different sizes, binning and dimensions.

Math: Unary operators

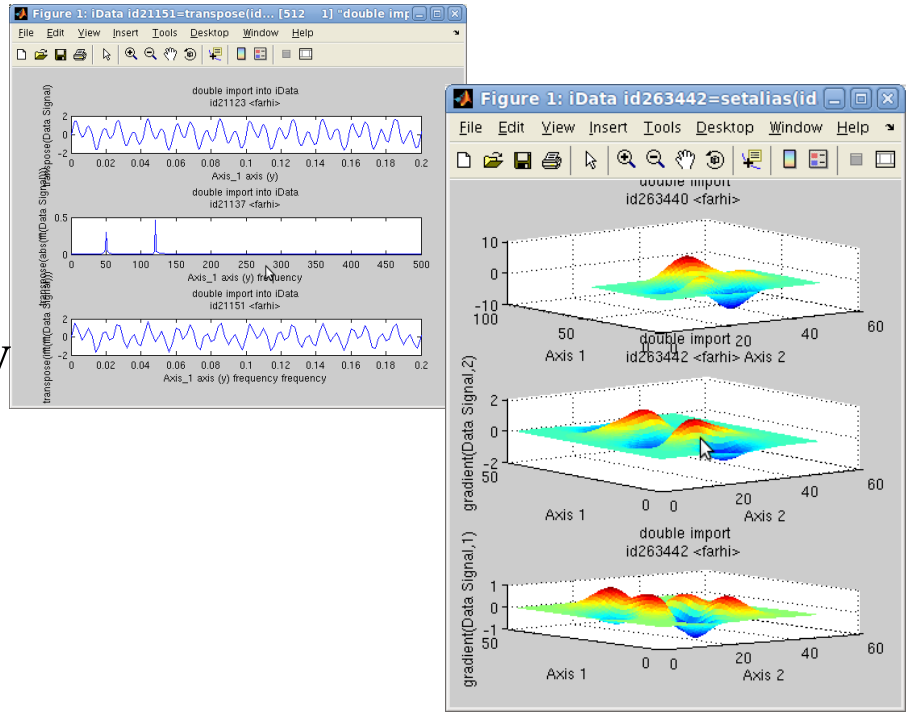
Unary operators take one iData object, and usually return a modified iData object.

```
>> b = cos(a)
>> c = -a; d= a';
```

The unary operators currently contain **27 methods**:

abs, acos, asin, atan, cos, sin, tan, exp, log (Neperian), log10, sqrt, conjugate transpose ('), transpose (.'), floor, ceil, round, sign, uminus (negate -), imag, real, fft, ifft, del2 (Laplacian), gradient, sum, prod, trapz

The **FFT operators** compute the *Signal* FFT, but also the reciprocal *Axes*.



The **gradient** and **Laplacian** operators allow to compute the local Taylor expansion of the Signal. The **Jacobian** uses gradients to perform coordinate frame changes automatically.

Math: Statistics, Projection, Integration

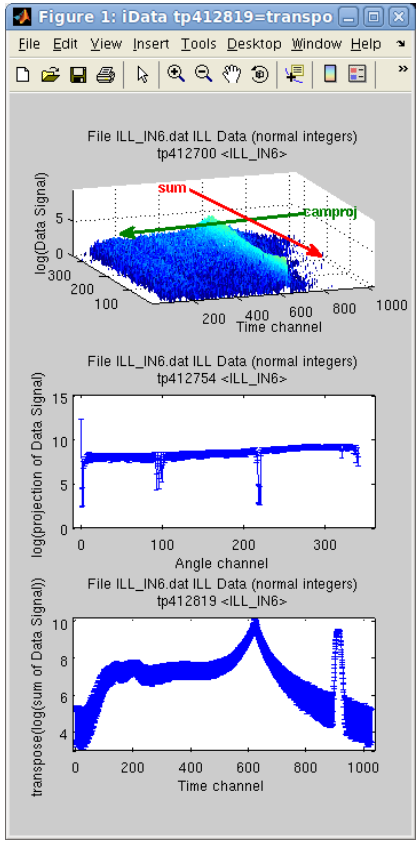
There are also methods to accumulate the object along axes (integrals)

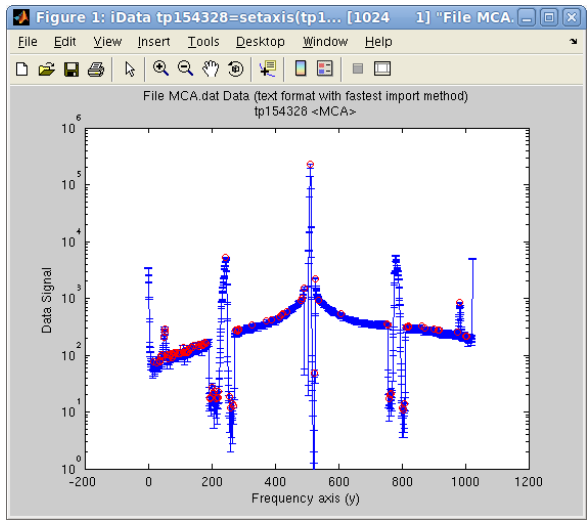
sum, camproj, trapz

prod, cumsum, cumprod

Statistical methods allow to compute the minimum, maximum, mean, median, Gaussian width and centre, as well as an automatic peak search.

min, max, mean, median, std, peaks





Math: Binary operators

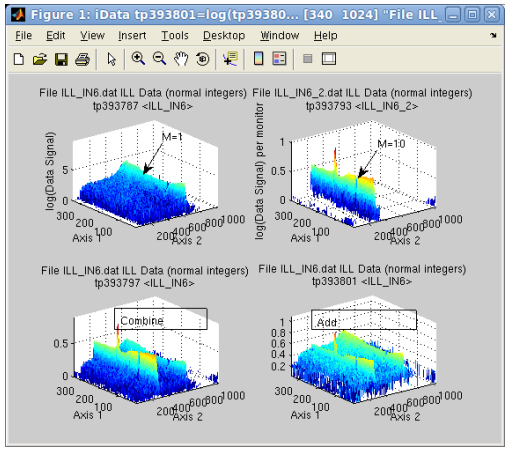
The binary operators take at least two iData objects.
The currently defined methods are:

plus, minus, divide, multiply, power, comparisons,
convolution, correlation, interpolation, combine

They all start by **finding the intersection** of objects along Axes, then check the **binning** and possibly re-sample on a common axis frame, then perform the operation.

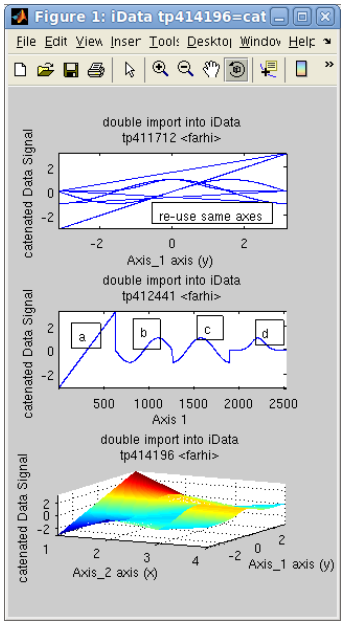
When objects of **different dimensionality** are used, the one of lower rank is expanded along the most sensible missing dimension, so that 1D row/columns objects can be added onto 2D ones.

The **combine** operator is an *un-weighted addition* to e.g. improve a measurement counting time.



Math: Appending and catenation

The **cat** operator can assemble an array of objects by appending their Signal along existing or new dimensions. This way a set of 1D objects can be assembled into a surface (side-by-side). Alternatively objects can be put one after the other when appending along an existing dimension.



Slicing an object can be done by simply extracting an index subset:

```
>> b = a(2:40, :);
```

Math: Exercise

- Import some data.
- Plot it.
- Set the axis to log-scale (contextual menu)
- Plot the *log10* of the object. Compare rendering.
- Load the [ifitpath 'Data/ILL_IN6*'] files.
- Set the Monitor to 1000 and 3000 resp. for the 1st and 2nd objects.
- Add the objects and plot the log scaled one
- Combine the objects and plot the log scaled one. Compare.
- Project one of the objects along the axis rank 1 and 2.
- Catenate the objects along the 2nd rank axis and plot the result.
- Experiment the *linspace*, and other operators.