

## (7) iFit/fits:

*The ultimate step in data analysis*

Once we have efficient optimizers and a data set, we may fit a model function to the data.

The syntax is about the same as the optimizers:

```
>> [pars,criteria,message,output] = fits(a, model, p0, options, constraints)
```

The model has the syntax:

```
function y=model(p, x1, x2, ...) %for instance gauss(p,x)
```

The model can also be entered as an expression

```
model = 'p(1)*x+p(2)'
```

The *fits* results are the same as the one from the optimizers.

The optimizer is set with `options.optimizer`

The model evaluation after the fit is in `output.modelValue`

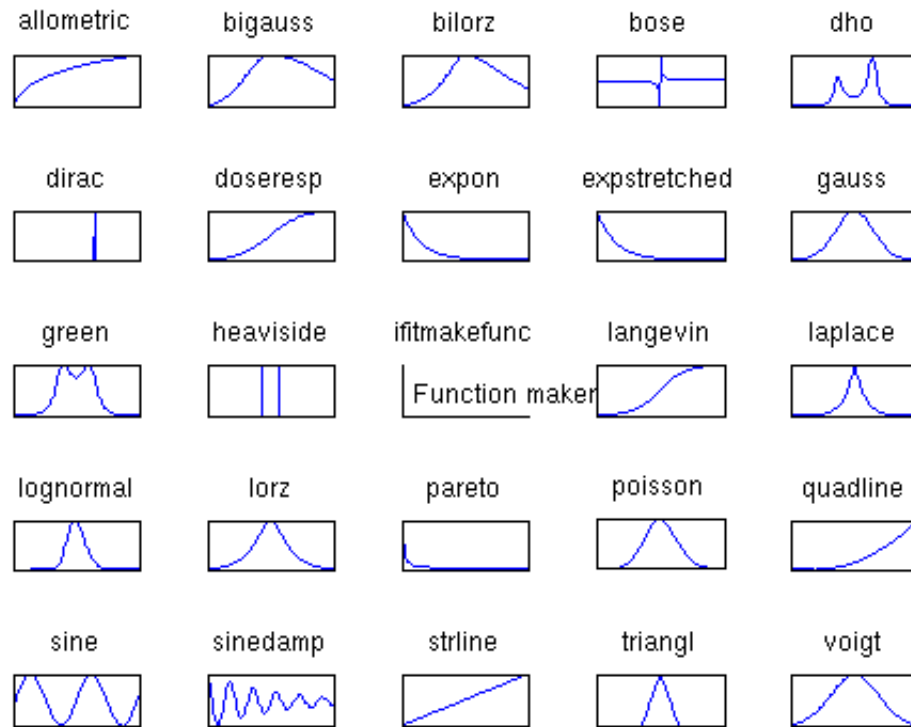
# iFit/fits: predefined model functions

All model functions can be displayed with e.g:

`gauss('plot')`

They all can guess starting parameters from the data set, so that p0 is not really needed.

`fits(a, model)`



When data set dimensionality is higher than the model, the model is expanded by perpendicular product along missing dimensions:

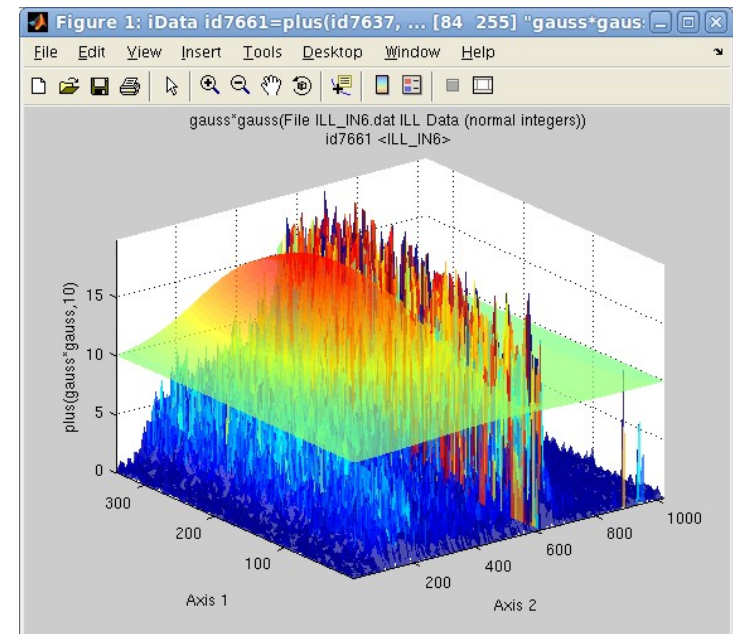
A 1D gauss may form 2D, 3D... gaussian model, automatically

```
>> fits(iData3D, 'gauss',...)
```

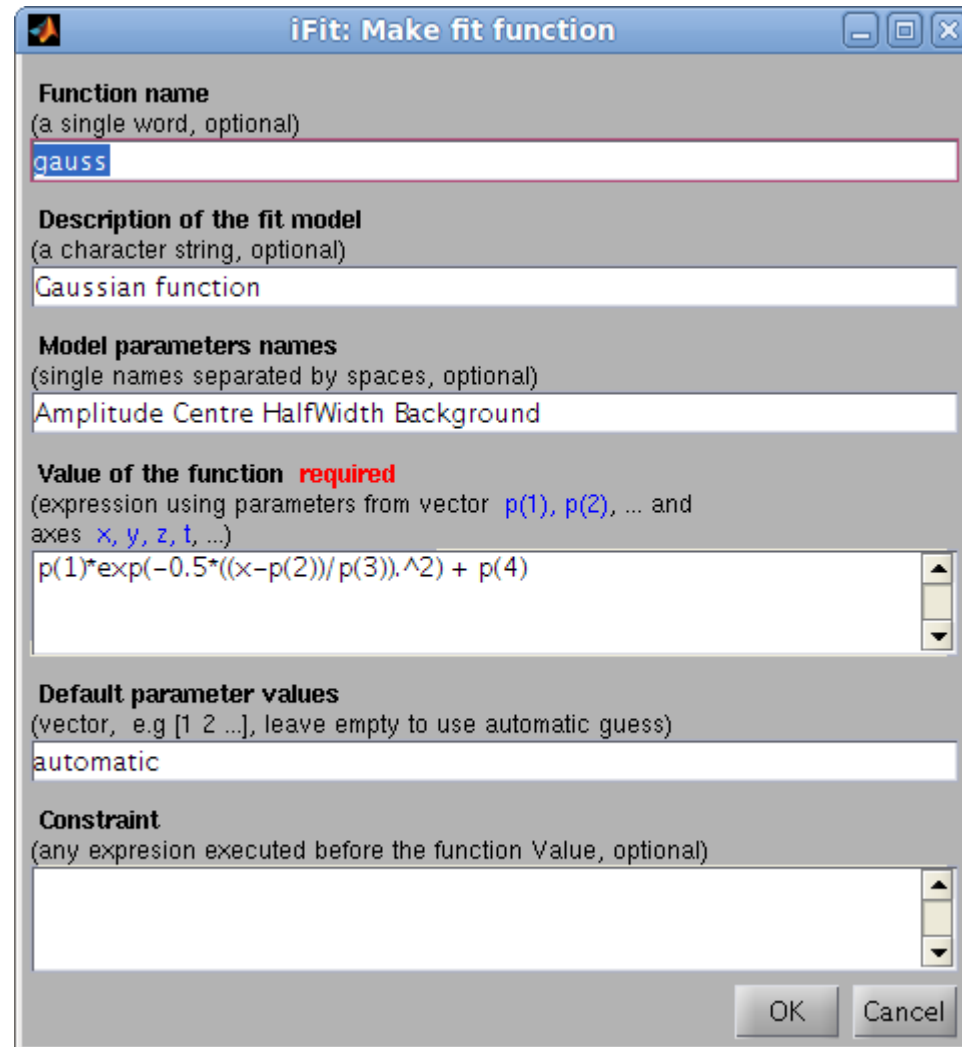
When the model is given as a cell of models, the models are multiplied perpendicularly

```
>> fits(iData2D, {'gauss','lorz'},...)
```

Fitting works in any dimensionality.



The *ifitmakefunc* command bring a dialogue to define a new fit model. It creates a function file, and returns a function handle that can be used for fitting.



The dialog box titled "iFit: Make fit function" contains the following fields:

- Function name** (a single word, optional): `gauss`
- Description of the fit model** (a character string, optional): `Gaussian function`
- Model parameters names** (single names separated by spaces, optional): `Amplitude Centre HalfWidth Background`
- Value of the function** **required** (expression using parameters from vector `p(1), p(2), ...` and axes `x, y, z, t, ...`): `p(1)*exp(-0.5*((x-p(2))/p(3)).^2) + p(4)`
- Default parameter values** (vector, e.g [1 2 ...], leave empty to use automatic guess): `automatic`
- Constraint** (any expression executed before the function Value, optional): (empty)

Buttons: OK, Cancel