# Review of Mathematica 4.2

**Nigel Backhouse**
**University of Liverpool**

sx52@liv.ac.uk

Release 4.2 is the latest version of the mathematical software package Mathematica, published by Wolfram Research (WR). It competes directly with the only other currently marketed high end integrated computer algebra system (CAS), Maple 8.

I shall concentrate in this review on highlighting the new features concerned with optimization which I think have most relevance to the teaching/learning applications. Although there are very many improvements and new features in Mathematica 4.2, just as for the last upgrade, not many will have immediate impact in the classroom. In fact, WR have introduced a number of related products, including CalculationCenter, The Mathematical Explorer, webMathematica and Mathematica: Teacher's Edition, aimed more directly at the teaching and fun sector.

I installed the software on a PC with a 2.4 GHz Pentium IV processor and 512 Mb of RAM, running under Windows XP – a specification which far exceeds the minimum recommended. It occupied about 220 Mb of hard disk space.

***Optimization***

Optimization is a topic that the symbolic packages have only flirted with; its algorithms are rather hard to programme and their implementations often computationally intensive in practise. Mathematica 4.2 offers new tools for various kinds of optimization, both linear and non-linear, and also in the context of data fitting.

The new implementation of the *LinearProgramming* function is in the spirit of the optimisation of a linear functional subject to linear constraints which can either be inequalities or equalities – of course an equality can always be replaced by a pair of inequalities, albeit an inelegant alternative. In earlier versions, this function had a more limited syntax.

Here is an example:

Minimise $x-2y+3z$, subject to

$$3x+5y-z \le 4;\ 6x-7y=5;\ 4x+2y+9z \ge 12;$$
$$-13 \le x \le 2;\ -1 \le y \le 1;\ -1 \le z < \infty.$$

The new command for this is

```
LinearProgramming[{1, -2, 3}, {{3, 5, -1}, {6, -7, 0},
{4, 2, 9}}, {{4, -1}, {5, 0}, {12, 1}}, {{-13, 2}, {-
1, 1}, {-1, Infinity}}]
```

with output

$$\left\{\frac{571}{499}, \frac{133}{499}, \frac{382}{499}\right\}$$

In the command, the first set of pairs represents the right-hand sides of the constraints and the appropriate Boolean test: -1 for "less than or equal to", 0 for "equal to", 1 for "greater than or equal to". The rest should be self-explanatory.

*Supplier's contact details*

Wolfram Research Europe Ltd.
10 Blenheim Office Park
Lower Road
Long Hanborough
Oxfordshire OX8 8LN

T: 01993-883400
F: 01993-883800
E: info@wolfram.co.uk
W: www.wolfram.co.uk

The same result can be obtained using the *ConstrainedMin* function, whose objective function and constraints are expressed symbolically:

```
ConstrainedMin[x-2*y+3*z, {3*x+5*y-
z<=4, 6*x-7*y<=5, 6*x-7*y>=5,
4*x=2*y+9*z>=12, x>=-13, x<=2, y>=-1,
y<=1, z>=-1}, {x, y, z}]
```

with output

$$\left\{\frac{1451}{499}, \left\{x \to \frac{571}{499}, y \to \frac{133}{499}, z \to \frac{382}{499}\right\}\right\}$$

Both of these functions are only applicable to linearly expressed optimization problems.

For non-linear problems, the new *NMinimize* function in the *Numerical Math* package offers a variety of methods for global optimization. The following examples support the claim of its superiority over the sometimes hazardous application of the *FindMinimum* function and of the *NonlinearFit* function from the *Statistics* package.

Suppose we try to locate the minimum of the function

$$f(x) = x^2 + 10\sin(x)$$

The Mathematica command

```
FindMinimum[f[x], {x, 2}]
```

has output

$$\{8.31559, \{x \to 3.83747\}\}$$

whereas

```
FindMinimum[f[x], {x, 20}]
```

has output

$$\{-7.94582, \{x \to -1.30644\}\}$$

The number paired with the variable in the command is the starting value for the algorithm used. Note that the minimum obtained is not necessarily the closest to the starting point.

A plot of the function shows that it has two minima.

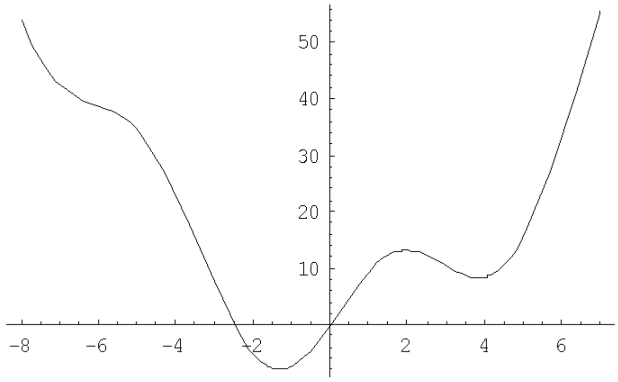


**Fig 1 Function with two minima**

The following yields the global minimum:

```
<<NumericalMath‘NMinimize‘
NMinimize[f[x], {x}]
```

with output

$$\{-7.94582, \{x \to -1.30644\}\}$$

For the second example, let us generate a data set from function values with random perturbations. We then try to reconstruct the original function.

```
SeedRandom[13];
model=a+Sin[b*2*Pi*t+c]+Sin[d*2*Pi*t+e];
parameters={a,b,c,d,e};
values=Table[Random[],{5}];
data=Table[Evaluate[{t,model /.
Thread[parameters->values]}]
+Random[Real, {-0.1, 0.1}], {t, 0,
0.05*2*Pi}]
```

whose output is a data set of which the first few values are:

```
{{-0.0118779, 0.693023}, {0.253306,
1.76439},{0.610906, 0.829856}, …,
```

Invoking NonlinearFit :

```
<<Statistics‘NonlinearFit‘
nlf=NonlinearFit[data,model, {t}, {#, 0,
1} &/@ parameters],
```

with output

```
0.169377 – Sin[0.373292 – 2.64238 t] +
Sin[1.19996 + 4.08387 t].
```

This can be plotted together with the original data set:

```
Lp=ListPlot[data, PlotJoined -> False,
DisplayFunction -> Identity,
PlotStyle -> {Pointsize[0.02]}];
nlfplot=Plot[nlf, {t, 0, 2*Pi},
DisplayFunction -> Identity];
Show[lp, nlfplot, DisplayFunction ->
$DisplayFunction];
```

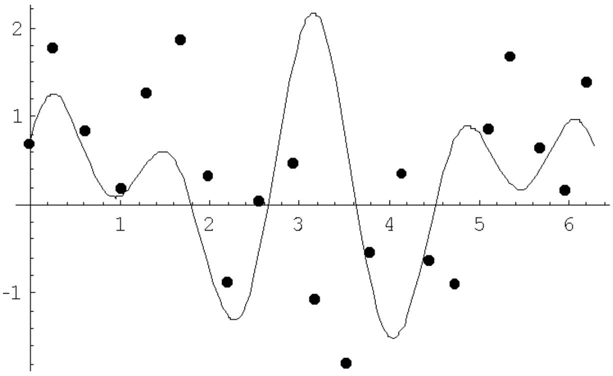

**Fig 2 Failed attempt to fit model to data**

So NonlinearFit gets trapped by a local minimum.

For a better fit construct a sum of squares function, which is then globally minimized:

```
SumSqrs=Sum[((model /.  {t ->
data[[I,1]]}) – data[[I,2)^2, {I,
Length[data]}];
Nmin=NMinimize[SumSqrs, {#, 0, 1} &/@
parameters,Methos->
"DifferentialEvolution"]
```

with output values

```
{0.550636, {a -> 0.201929, b ->
0.191519, c -> 0.387394, d ->
0.802503, e -> 0.0205679}}
```

The results can be plotted

```
Nminplot=Plot[model /.  Last[nmin],
{t, 0, 2*Pi}, DisplayFunction ->
Identity];
Show[lp, nminplot, DisplayFunction ->
$DisplayFunction];
```

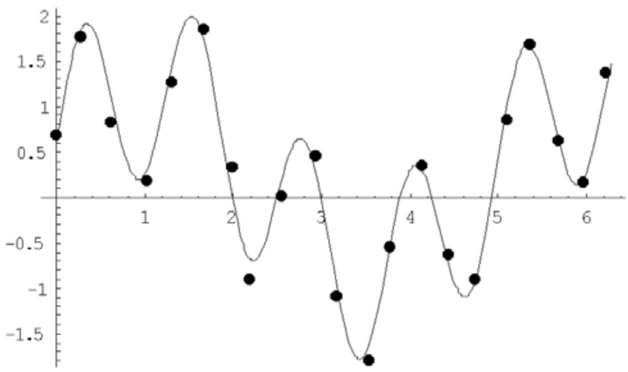

**Fig 3 Good fitting of model to data**

There is a choice of four methods to achieve the global optimization: Differential evolution, Nelder-Mead, simulated annealing and random search.  They give different values of the model parameters, but the final plots are indistinguishable to my eye.

### *Other enhancements*

Mathematica is a mature software package in respect of the areas of mathematics which the developers have chosen to address.  On the whole the mathematical improvements incorporated in version 4.2 are incremental. For example, the tools for the simplification of expressions now allow for the use of transformations involving *ProductLog* (also know as the Lambert function), *Mod*, *Log,* the error function and its relatives.  There are more improvements to the already very useful *Combinatorica* package, and a new addition to the *Statistics* package for the analysis of variance (ANOVA). Although there are also speed and robustness improvements, the main thrust is directed towards presentational, publishing and web issues.