

# MuPAD feels the benefits of going commercial

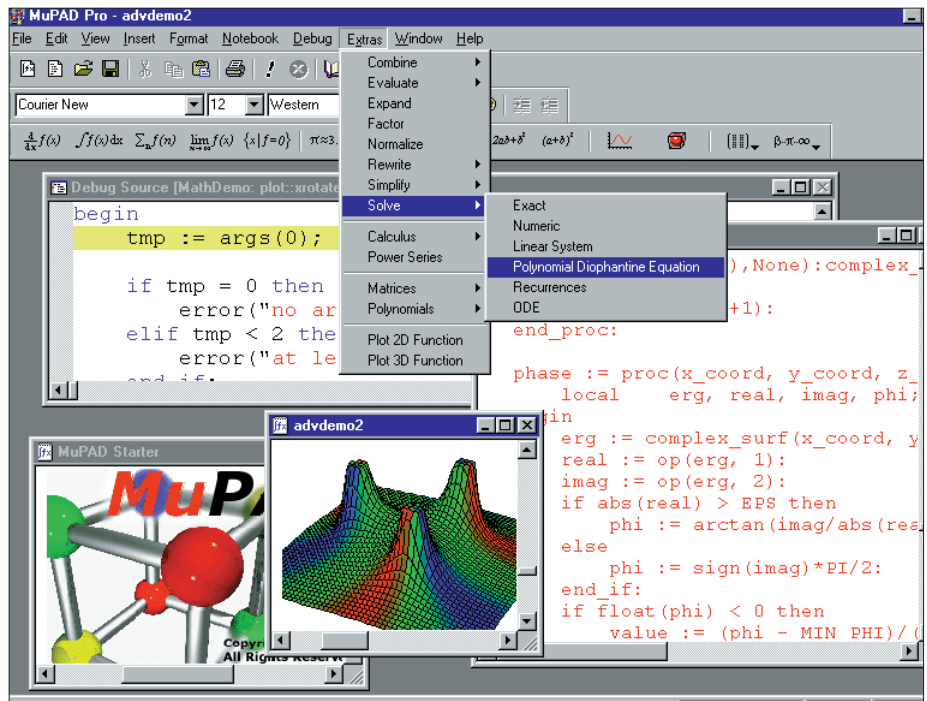
*Ray Girvan looks at MuPAD 2.0, the powerful computer algebra software which has moved on from its days as a freeware program*

When I last reviewed MuPAD, in September 1998, it had reached a crucial stage in its career. Developed at the University of Paderborn, Germany, it had seemed a classic example of a successful open source project: a powerful computer algebra system available as freeware for the scientific community. But the MuPAD Group had come under pressure from university and other backers for the project to begin paying its own way. Rather than throw MuPAD into the public domain, the Group decided that MuPAD 1.4 onward would be a commercial product. To market it, SciFace Software was set up, employing former MuPAD Group members and headed by Dr Oliver Kluge, one of MuPAD's leading developers. Would the strategy succeed?

Nearly three years on, I think the answer is yes. There has been a slight retreat on one front: plans to implement versions for parallel computers have been shelved except as a research project. But otherwise, MuPAD has gone on to a new version 2.0, and also recently attracted support from MacKichan Software in the form of a worldwide marketing agreement and MacKichan's decision to bundle MuPAD as one of the mathematics engines in the Scientific Workplace family of high-end LaTeX products. 'We were anxious to support MuPAD,' said Barry MacKichan, CEO, 'because SciFace was eager to work with us and develop the synergies between our products. We see our products evolving in parallel and supporting each other.' With this mainstream support, the future looks hopeful.

## Internal changes

However, the advance from MuPAD 1.4 to MuPAD 2.0 involved radical changes that will affect some users more than others. On



The advance from v1.4 to v.2.0 was 'much more than a normal upgrade', according to MuPAD's developers.

the surface, v2.0 is definitely easier to use: the very sparse menu of control functions in v1.4 has been extended with icons and pull-down menus of templates for the main symbolic operators: integrate, limit, solve, etc. The display is also improved: MuPAD is no longer outputting formulae as chunky monospaced ASCII but in legible typeset form. But once you begin to experiment, you find many altered library function names and properties which affect even simple command line mathematics; for instance, 'asin' (for inverse sine) is now 'arcsin'.

Programmers who write procedures in the MuPAD language will find even greater differences. The central change is the switch from dynamic to lexical scoping. Lexical scoping, used by languages such as Pascal and C, fixes the role of variables – e.g. local 'x' – according to the block where they are first defined. Dynamic scoping, used by MuPAD 1.4 and languages such as BASIC and early Lisp, decides a variable's role by the context

where it is called. The practical outcome is that dynamic scoping in MuPAD gave rise to funarg, closure and level problems – instances of unwanted 'cross-talk' between local and global variables – that are cured by the change to lexical.

Since much of MuPAD is written in the MuPAD language, this needed major work, explaining why v2.0 has been slow in arriving. Kluge commented: 'This was much more than a normal upgrade. The data structures were redesigned and all library functions needed modifying. We needed a very long beta test to ensure this version was really stable.' The advantages, though, are that procedures operate independently of context, making them easier to write, add and test. MacKichan said: 'During the development of our MuPAD support, I was impressed by the promptness and ease with which the MuPAD team was able to fix bugs and provide enhancements.'

This tells me that MuPAD is, as the devel-

Web: [www.mackichan.com](http://www.mackichan.com) Email: [info@mackichan.com](mailto:info@mackichan.com) Toll Free: 877-724-9673 Telephone: 206-780-2799

opers claim, modular and easily maintainable.'

Fortunately, there are additions to v2.0 which will help existing users adjust. While there is no translation utility as such, MuPAD 2.0 has a 'prog::changes( )' command that runs MuPAD in 'compatibility mode'. Whether in command line or program use, this flags obsolete v1.4 functions and, if possible, passes the command to the equivalent in v2.0. Programmers can get help in the Windows MuPAD from the graphical source code debugger, previously only implemented in Unix and Mac versions. There's also considerably more documentation than previously; the 550-page online manual of v1.4 has been expanded to 2,300 pages including a tutorial (also in print form), quick reference, and detailed references for all commands in MuPAD's library. As with v1.4, the manuals are in LaTeX hypertext files.

### Advanced maths and programming

My general impression of MuPAD is of a package written by pure mathematicians and advanced programmers. Its mathematical repertoire is larger than most algebra packages; in 1994, MuPAD was joint winner (with the now-defunct PC-Macsyma) of Michael Wester's test suite for range of problems handled. Beyond the university-level maths functions usual with algebra packages, MuPAD has tools for working with more advanced pure maths concepts such as rings and fields. Its special packages include Combinatorics, Graph Theory, Groebner Basis calculations, and Number Theory, along with an expanded Plot library with new types such as Turtle graphics, L-systems and a direct plotter for numerically solved ODEs.

MuPAD 2.0 is also remarkably rigorous; for instance, solving  $a*x^2+b*x+c=0$  for  $x$  returns not only the classic quadratic solution, but also alternative results for special cases when coefficients are zero. But there are surprising omissions: tools for PDE (partial differential equation) solution are very limited, almost as if the developers had ignored this mainstay of scientific computing because most PDEs don't have analytical solutions.

MuPAD programming also goes into unusual areas. Apart from the usual integers, reals, etc, MuPAD data types extend to 'axioms', 'categories' and 'domains' based on mathematical classes such as polynomials or finite fields. The language supports functional programming techniques, such as folding, nesting and currying, and object-oriented concepts such as 'overloading' functions with user definitions to extend their application outside their normal data type. Every function has user-definable 'function environments' that can customise what happens

## MuPAD: The development of an algebra tool

MuPAD ('Multi Processing Algebra Data tool') dates from 1990, the name also alluding to its origin and development at the University of Paderborn. It was intended originally as a tool for efficiently handling gigabytes of data from investigations of group theoretical structure in non-linear systems. However, over the decade Paderborn's MuPAD Group developed it, 'as a service to the scientific community', into a general-purpose algebra system supported on many platforms including Unix systems (Linux, Sun OS, Solaris and SGI), Macintosh OS, and Microsoft Windows.

MuPAD was freeware until 1998, when lack of funding led the MuPAD Group into setting up SciFace Software ('Scientific Interfaces') to develop MuPAD 1.4 onward as commercial versions. Despite some users' disappointment, this appears to have been a successful strategy, leading to the new version 2.0, as well as a partnership with MacKichan Software.

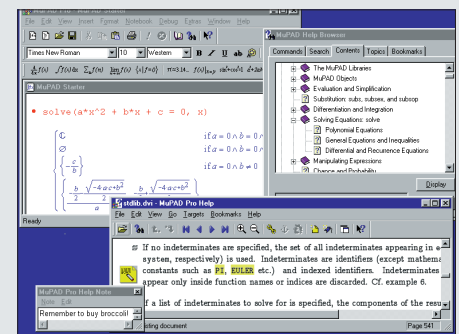
Free academic licences are, however, still available for older releases, and 'Light' and 'Academic' versions with limited functions. MuPAD's use is still primarily in the academic market. According to Oliver Kluge, CEO of SciFace, the main users are mathematicians, engineers and economists, but there are also commercial customers using it in research, such as Brother in Japan, and Siemens-Nixdorf using it to develop banking and retail systems.

Version 2.0 Pro has been launched for Windows 95, 98, ME/2000 and NT, and offers a large symbolic and numerical command set in a modern Windows 'notebook' format with typeset formula layout and Vcam ('virtual camera') graphics viewer for well-rendered 2D and 3D plots. MuPAD notebooks use a command line interface, with designated active 'regions'. A red prompt indicates an input region for formulae and commands; on execution, the result

returns in blue in an output region. Like a spreadsheet, MuPAD allows global recalculation, where changes will cascade down the notebook after an input is altered. For commentary, text regions can be set up, which in Windows versions can contain live OLE (Object Linking and Embedding) links to other programs.

MuPAD is an open system: apart from the C/C++ kernel containing its core functions, most of its mathematical expertise resides in user-readable and portable libraries written in the MuPAD programming language. User programs are easily developed by writing Maple-style 'proc ... end\_proc' procedures, but despite the Pascal-like syntax, the MuPAD language is built centrally around functional programming, with procedural and object-oriented elements. For programmers, MuPAD Pro contains a source-code debugger for troubleshooting user-written procedures, and advanced users can also create 'dynamic modules', C/C++ application code blocks that MuPAD loads at run-time.

Commercial versions come with a 360-page MuPAD tutorial which is expanded with over 2300 pages of hypertext documentation. MuPAD will also launch a Web browser to access FAQs, technical support, news, and web demos at the SciFace site.



MuPAD's detailed help system allows user annotation.

when the function is called. Many of these concepts are, frankly, beyond the knowledge of the average scientist used to routine procedural programming, but for serious mathematicians and 'power users' developing large applications, the capacity is there.

When I last reviewed MuPAD, I thought that the documentation needed a friendlier touch. Although the printed tutorial is a great improvement, my reservations still apply to the hypertext material, which is translated from German. Much of the content is equivalent to, say, *Maple's Programming Guide* or *The Mathematica Book*, but the style is pitched at a difficult, sometimes obscure, level (and not helped

by a rather dated font). Nor do MuPAD and its manuals communicate easily. Despite useful features in its help browser – such as the ability add notes, bookmarks and custom hyperlinks – it's inconvenient that MuPAD can't print the LaTeX pages and only special highlighted examples can be copied to a notebook. HTML manuals will be a future option, but I think that annotated MuPAD notebooks (many more than the four introductory ones) would be an even friendlier medium for learning how to use it. MuPAD is a very impressive package that has made a successful transition to commercial form; my only reservation is that its documentation still has an academic flavour.