

Review of Maplets

Grant Keady
University of Western
Australia

keady@maths.uwa.edu.au



Maplets are a new front-end to Maple, an alternative to the Worksheet front-end. (They require Maple 7. They also require the Maplets package - a Maple 7 package - which was developed after Maple 7 was released. See www.maplesoft.com for distribution details.)

For some users, learning Maple syntax is a chore, and may distract from the task at hand. This can certainly apply to some first year undergraduates: for example, those for whom mathematics is a service course and who are unlikely to make extensive use of manipulative mathematics in their later courses or careers. (As software development does cost money, it is probably worth mentioning that there may well be other paying customers with similar desires to avoid learning computer language syntax. These people - some engineers and other professionals - may be proficient in other computer packages but may have an occasional need to do some standardised calculation where Maple is at its core. It may be more efficient for such engineers, etc. just to do the job in a pre-prepared Maplet, rather than suffer the distraction of yet another syntax to learn.)

Maplets enable these users access to the parts of Maple relevant to their needs while shielding them from any problems associated with Maple syntax.

Perhaps an example is in order here. The example given is one which might be appropriate in a first year Calculus course. Here is a picture of the screen when running the CurveAnalysis Maplet in the collection of Maplets already available (as examples on the demonstration CD or from www.mapleapps.com).

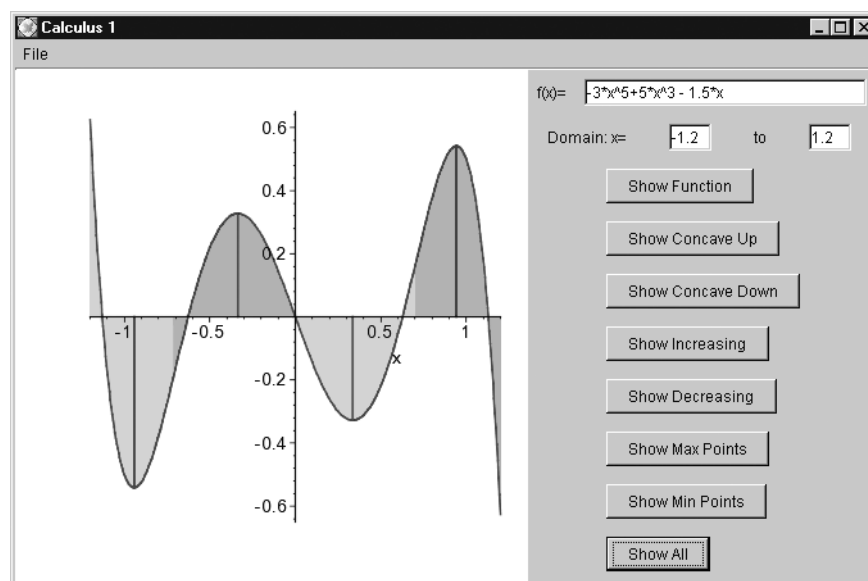


Fig 1 Screen from CurveAnalysis.

What the student sees in using Maplets: Maplets and Java Swing Classes

Maplets use calls to the Java Swing Classes to handle Graphic User Interface (GUI) details. The screen image of the CurveAnalysis example provided is typical enough. The elements from the Swing Classes used include dialog boxes (TextBox), and buttons. Items not illustrated there are pull-down menus, checkboxes, etc. (One can imagine pedagogic variations of the example in which these might be used. For example, in teaching, it is often appropriate to ask students to do tasks, and here it might be appropriate to ask them to select from a list of intervals provided via an appropriate element where the function

Supplier's contact details

Adept Scientific plc
Amor Way
Letchworth
Herts, SG6 1ZA

T: 01462 480055
F: 01462 480213
E: maple@adeptsience.co.uk
<http://maple.adeptsience.co.uk>

is increasing, etc. Maplets and the Swing Classes cope with this easily.) Many other GUI elements are illustrated in the Maplet examples publicly available, both those on the CD-ROM evaluation disk provided and those available at the mapleapps.com repository.

Maplets are launched from Maple. Once Maple knows the definition of myMaplet (say), the Maple command for running it is:

```
Maplets[Display]( myMaplet );
```

One could hide the Maple - both that defining myMaplet and also the callout to it - in a script or, on Windows, a .bat file. The script would be one which started Maple, read in Maple code which in turn launched the Maplet. (For better methods that will come with Maple 8, see the section, "How Maplets work".) A student using a Maplet this way need not even be aware that Maple is involved at all.

(For a succession of Maplets run under Maple, I guess that running the Maplets from within a Maple Worksheet format might be easiest and most effective. This would allow text written in text fields in the Maple Worksheet, and the usual Worksheet sectioning and other formatting features being used appropriately, in tandem with the Maplets. It would still be effective in shielding users from Maple syntax, though users would be aware of Maple.)

The instructor: authoring Maplets

The writing of code for complex code is a fairly detailed business. The layout, etc. has to be specified.

(The next paragraph is not really about Maplets, but something which could be more or less functionally the same for the student user. It is certainly possible to have Java call Maple. See references under Kolokolnikov [2]. Contact me for simple examples calling out to his Maple.class if you wish. My examples are deliberately simple, but it is not hard to add GUI items, at least for tasks one at a time. What one gets with Maplets, though, is something different. In particular there is a unified method of authoring many similar Maplets.)

The Maplets package for Maple 7 enables the writing of the GUI code to be done from Maple. In writing a Maplet, the first command to Maple is

```
with(Maplets[Elements]);
```

In my view, the easiest way to begin writing Maplets is to start with the examples, and modify them. I had no difficulty writing simple Maplets. The Maplets package makes intelligent decisions on layout, and for a fairly simple GUI, writing a Maplet is easy; it is also capable of writing more complex and sophisticated GUIs.

How Maplets work

The following is a brief, high-level description of the Maplets architecture based on that provided by Jason Schattman of Waterloo Maple.

The user (student user, or staff author) interacts with the Maplets package within Maple. The Maplets[Elements] part of the package generates an XML description of a Maplet. Using Maplets[Display], this description is sent to the Maplets runtime engine, which is written in Java. This runtime engine interprets the XML description of the Maplet, dynamically creates the necessary user interface (Java Swing) components and displays the Maplet.

(Unlike the discussion above associated with Kolokolnikov's example code) there is no standalone Java code associated to a specific Maplet. Maplets are XML descriptions that are taken as input by a generic runtime engine.

We now quote verbatim from the Waterloo Maple source:

"That said, in Maple 8, users will be able to save a Maplet to a standalone file, which can be shared with other users. Using "Save as ... Maplet" stores all the information necessary to reconstruct a Maplet (defined in a worksheet) to a file. Executing (double-clicking) this file on any machine that has a Maple 8 installation launches the Maplet."

Pedagogic purposes?

This section is rather speculative. There are two aspects.

As a small incremental change? One view of software is that it can come in two main sorts - "appliance software" as one sort, "environments" as another. "Appliance software" does just one task. Appliance software is easy to use. (At least it is meant to be easy to use. Younger students usually work out how it goes experimentally faster than older academic staff!) However, if different appliances come in too many different styles, it can be confusing. Maplets, by using Java Swing classes, should feel familiar to our students. (This is not to say that one cannot succeed in writing a confusing Maplet if one set out to do so.)

"Computing environments" are, for us, big packages like Maple, Mathematica or Matlab or similar. Such software which can do lots of different tasks. The obvious problem with environments is that they can become bloated (and many people think this may be the case with Microsoft Office, for example), and, hence difficult to maintain as reliable software. Maple has correctly avoided too much in-package word processing (as LaTeX

does the job) and also too much in-package float numerics (as Matlab/Octave) does the job. Maple correctly aims at extending its functionality by intelligent connection or inter-relation with other software. Adding to its functionality the production of (Java-Swing-based) Maplets seems a reasonable increment.

In the near future, perhaps most 1st year students in maths departments where Maple is the dominant package in teaching will use Maplets (or similar) rather than Maple. There should then be a smooth transition at 2nd year to students using pre-prepared Worksheets more than Maplets, with some Maplets around both for those who really don't want to program and to achieve continuity with the first year experience. At 3rd year it seems reasonable to expect all maths students to be able to program in Maple (or similar) to solve appropriate tasks of their own.

Pedagogic purposes for Maplets associated with larger-scale teaching software? Many Calculus, etc. CDs have been written. Some even have Maple as an algebra engine underpinning them. (One of these that I have seen, but whose style I personally do not like, is the "Journey through Calculus" CD-ROM which accompanies the quite good first year "Calculus" book by James Stewart.) I imagine that Maplets could provide an attractive way of getting a consistent feel to pedagogic materials of this kind. Perhaps if the Mathwise project were to have started now rather than a decade earlier, Maplets would have been seriously considered to underpin it.

In reference [3] Douglas Quinney describes "Calculus Machina". This builds on Java, and is underpinned by its own Computer Algebra system. Once again, a reasonable fraction of the work described there could be implemented effectively in Maplets if one were to choose to do so. Of course, not every software developer is going to want to pay to buy Maple licences, and if the computer algebra needs are modest, it is valid to use smaller computer algebra packages.

The future? Java, Computer Algebra and the Web

I know of at least three Java-Maple combinations, all Web-server related. The first is AIM. AIM is reviewed in [4]. AIM uses Java servlets. The next two use JSP pages and involve GAP or Maple as alternatives in the algebra engine used. They are javamath, and (I've been told) the next edition of "Algebra Interactive".

LiveMath, then called MathView was reviewed in [1]. At some stage in the future one can imagine the functionality of LiveMath being delivered from something Maple based. Perhaps it will be something that to the students and lecturers will feel Maplets-like. It will almost certainly involve some sort of Maple-Java combination.

This is enough speculation about the future. Maplets, will have some role on the way through to it.

Back to the present

This reviewer looked at the Maplets package from a Reviewer's CD supplied by Waterloo Maple. I tested it on a Windows98 machine with the Java Runtime Environment provided on the CD. On a Windows 2000 machine which had a slightly later version of the JRE, I chose to see how it worked if I didn't change it. It worked perfectly, with all the tests I made. So, in conclusion, it is a product which works without problems.

References

- [1] G Keady and D Hill, *Maths&Stats*, Nov 1997
- [2] T Kolokolnikov, Maple Forum, Feb 2002: http://www.scg.uwaterloo.ca/~maple_gr/Digests/Digest00.02
- [3] D Quinney, <http://ltsn.mathstore.ac.uk/articles/mathscaa-series/nov2001/index.htm#casestudy>
- [4] N Strickland, *MSOR Connections*, Feb 2002

Waterloo Maple comments supplied by Madelaine Currier (Public Relations Manager)

At time of publication, Maple 8 will have been formally announced by Waterloo Maple. Maple 8 is scheduled to ship sometime during the second quarter of 2002. One of the most significant features of this release, as alluded to in the article, are improvements to Maplets. In addition, Maple 8 will offer a brand new collection of functions for exploring introductory calculus, a comprehensive library of scientific constant definitions, numerical solvers for partial differential equations

and a series of improvements to the standard worksheet environment. Readers are invited to consult www.maplesoft.com for full details on the various new features and other improvements.

Finally, I would like to address your concerns about the distribution of Maplets through Maple Primes.... given your concerns, pointing readers to the [maplesoft.com](http://www.maplesoft.com) site as you have done in your article is very acceptable and I too appreciate your restraint. Indeed, restricting

THE INSTITUTE OF MATHEMATICS AND ITS APPLICATIONS

FOURTH CONFERENCE ON **Mathematical Education of Engineers** 1st – 3rd April 2003, Loughborough University

FIRST ANNOUNCEMENT AND CALL FOR PAPERS

Conference Objectives

The first three conferences “Mathematical Education of Engineers” held at Loughborough provided an overview of the then current general education background relevant to the teaching of mathematics to engineers. Within this forum, mathematics teachers from a wide variety of higher educational institutions together with engineering colleagues and participants from industry were able to discuss issues such as decreased teaching time, increased numbers of students, improved quality assurance, relevance of topics in the syllabus and the maintenance of both university and professional standards. Since the 1994 conference some important reports have been published. These include the Engineering Council’s revisions to its SARTOR requirements, which makes strong reference to the level of competence in mathematics expected within the training of engineers, and “Engineering Mathematics Matters”, published by the IMA and taking into account the views of a number of Engineering Institutions, which sets out a proposed core curriculum in mathematics for engineering undergraduates. In addition, the report “Measuring the Mathematics Problem” presents some of the general concerns about the mathematical ability of students entering tertiary education. Since the last conference the HE sector has been involved with Subject Review (QAA), Benchmarking, Programme Specifications and student intake changes arising from the introduction of A/S and revised A-level qualifications.

This conference aims to reflect the progress and experiences of initiatives within the teaching of mathematics to engineers in recent years, to debate areas of known concern and to learn together from current best practice. A further key aim is to examine collectively as mathematicians, academic engineers and engineers in industry the engineering mathematics provision for the future in the context of ever-increasing demands of competence, effectiveness, quality and relevance. The programme will include invited speakers, contributed papers (or posters), workshop sessions, demonstration sessions and a forum.

The proceedings of this conference will be published by the IMA.

This event will meet the requirements of the IMA Continuing Professional Development Scheme.

Registering an interest

The programme will include invited speakers, contributed papers (or posters), workshop sessions and a forum. Informal enquiries can be made by e-mail to L.R.Mustoe@lboro.ac.uk or stephen.hibberd@nottingham.ac.uk

If you are interested in this conference please register your interest by contacting Belinda Morris, Conference Office, IMA, (email: conferences@ima.org.uk) indicating whether you would like to present a paper or display a poster, or contribute to a workshop session.

Local Organising Committee

Leslie Mustoe (Loughborough University), Stephen Hibberd (Nottingham University), Duncan Lawson (Coventry University), Sarah Williamson (LTSN Engineering).

Waterloo Maple comments (continued)

the availability of Maplets to EMP subscribers through Maple Primes is a corporate strategy that Waterloo Maple has decided to take for a variety of reasons that I will not cover in great detail here. However, suffice it to say that such exclusivity is not permanent. Maple Primes is meant as a vehicle to provide early access to new features and features-in-progress, instead of (as we have done in the past) saving up these new features and releasing them at once at the time of a new release. Those who have access to the early and

preliminary releases are a well-defined, easily managed (eg from a tech-support perspective), Maple-savvy and Maple-loyal community. It simply makes sense to reward this loyalty and we have decided that this is an appropriate way to do it.

I do appreciate your comments and have forwarded them to the appropriate people at Waterloo Maple.