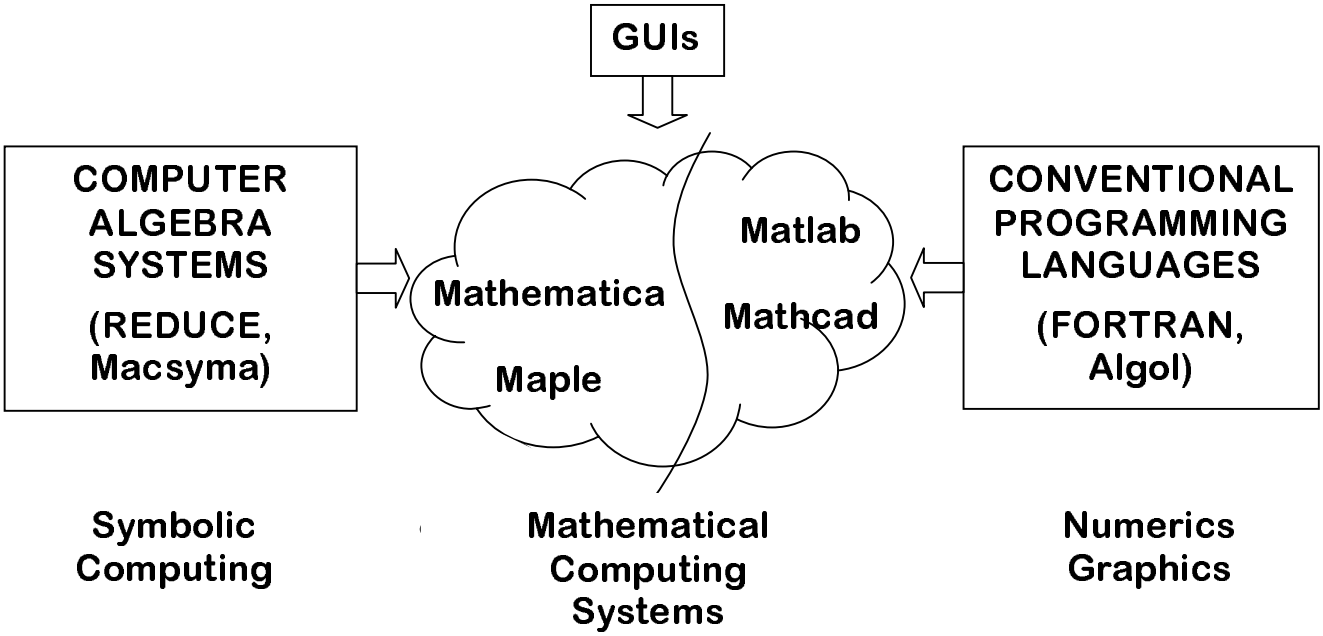
Using "Computer Algebra" Systems for Teaching

Francis Wright

Queen Mary, University of London

Open University
Tuesday 11th December 2001



Contents:

1 Comparison of Computer Algebra Systems
2 Limitations of Computer Algebra Systems
3 Teaching with Maple at Queen Mary
4 What students find difficult in Maple

1 Comparison of Computer Algebra Systems

My interests: REDUCE, Maple (not Mathematica!)

History

1960	FORTRAN ... FORMAC		
	Algol 60 ... Formula Algol		
	Lisp ... SAINT	Integration	
1970	REDUCE	Theoretical Physics	Lisp
	Macsyma	Artificial Intelligence	Lisp
	Scratchpad	IBM Research	Lisp
	muMath	Teaching *on PCs*	Lisp
1980	Maple	Teaching	C
	SMP	Applied Maths	C
	Cayley ... Magma	Pure Mathematics	C
	GAP (*free*)	Pure Mathematics	C
1988	Mathematica (*GUI*)	Science	C
	Derive	Teaching *on PCs*	Lisp
1990	Axiom / A# (*strong typing*)	Research	Lisp / C
	MuPAD	Pure Mathematics	C
	CA calculators		
1995	OpenMath / MathML		

Developments and current status

Implementation languages: Lisp versus C

Lisp

- First, but now old-fashioned?
- Lists and garbage collection built in
- Natural for symbolic computation

C

- Modern, popular, expertise available
- Must implement all data structures, memory management
- Good for GUIs, graphics, numerics

Modern Lisps implemented in C anyway!

Other Issues

User language

- Levels (REDUCE)
- Syntax (Lisp / Algol-like)
- Object-oriented programming support

Graphical User Interface

Graphics / Numerics

User support, updates, mailing lists, development activity

Speculations about the Future

Magma / GAP

Maple / Mathematica / muPAD (?) / Derive (calculators)

??? Open-source REDUCE / Emacs REDUCE ???

2 Limitations of Computer Algebra Systems

Not good for

- general, fast, low-precision numerics
  - [but access to NAG library]
- general-purpose graphics
  - [but OpenGL, real-time rotation]
- abstract mathematics, e.g.
  - *abstract* proofs
  - *abstract* vector spaces [need bases]
  - *infinite* sets
  - *indefinite* structures, such as matrices with indeterminate sizes
- general simplification, e.g.
  - trigonometric functions
  - most compact form
  - general powers

Other issues:

Linking with external routines

Object Oriented Programming

Customisability of I/O, parser

Use as a "compute server"

Data interfaces: OpenMath, MathML, TeX, ...

3 Teaching with Maple at Queen Mary

History

1980s, VAX VMS, *FORTRAN 77*

- Mathematical Computation I / II
- Numerical Analysis
- Principles of Numerical Computing

1985, BBC Workstations, *REDUCE*

- Algebraic Computing

1990, IBM PC, MS-DOS

1993, IBM PC, Windows 3.11 (95, ME), *Maple*

- Computational Mathematics I / II
- Experimental Mathematics
- Solving PDEs
- Projects
- (Algebraic Computing → Maple → dropped)

2001 Textbooks for CM I / II by FV / FJW (CRC Press)

CW etc. via web only (PDF or Maple)

Maple: GUI but not palettes

Typical CMI / II CW questions – *see separate Maple worksheet*

4 What students find difficult in Maple

Problems with programming languages reflect underlying general problems with precise use of formal language.

Technical details

- Editing, moving cursor, cut and paste
- Saving frequently: autosave option
- Execution groups: newline versus execute (*see worksheet*)
- Syntax – *see separate Maple worksheet*