PRELIMINARY TECHNICAL EVALUATION of Wolfram's *Mathematica*

William Bricken
April 1988

CAVEAT

Evaluation is from reading the manual thoroughly, and a few minutes of playing with the program.

OVERVIEW

The work is impeccable.

The functionality is encyclopedic: *all* mathematics, slanted toward theoretical physics.

"All common special functions of mathematical physics."

Documentation is exceedingly clear, the best text on mathematics I have seen.

Fully integrated with C, UNIX, SunView, TEX, PostScript.
Supercedes PROLOG, LISP, Smalltalk.

Overwhelming argument to adopt Wolfram's notation. Natural mathematical expression. Function composition and equality as main techniques.

Many sophisticated optimizations, including several mathematical breakthroughs.

Real time response, interactive system.

Notebook capability for computable textbooks.

MAIN TECHNIQUE

"Mathematical Programming"
Uniform representation as EXPRESSIONS: Head[arguments]
Uniform transformation: rewrite expressions to preferable expressions.
Substitution semantics (equals for equals)
Exact/symbolic computation, reverting to numerical methods as last resort.
N operator exact->approximate.
Undefined or irreducible labels are passed as literal.

COMMERCIAL ADVANTAGES

Scope and efficiency suggest this tool will be a dominant programming language in the 90's. Architectures can be optimized to the technique.

MARKET

All scientific computation (the FORTRAN market),
all symbolic computation (the AI market),
all graphic computation (the CAD market).

INTEGRATION WITH CAD

Graphics for arbitrary curves, 2D and 3D, with lighting and viewpoint.

Hidden line removal, boxing, ...

We do the display better. It provides symbolic/algebraic description and computation. Animation potential.

WEAKNESS

Formal logic (eg Boolean minimization), set theory. Tools for this are built in, but not developed.

Power for computational logic, animation not documented.

Scope of applicability is too much, needs packaging.

PLOTTING

Exact functions rather than splines.

Specifiable hidden surface removal, shading, mesh, clipping, contours and projections.

PROGRAMMING

Procedural, functional, logical, rulebased, constraint, object-oriented and mathematical techniques fully integrated.

Iteration, recursion, streams/generators, pipes, lambda abstraction, set functions.

Error handling, debug, trace, contexts, dynamic programming.

Operator theory, functionals, abstraction.

MATRIX TECHNIQUES

Very sophisticated and advanced.
Pure functionals as maps into matrix elements, accessor functions.
Associative vector multiplication.
Exact inverses, symbolic matrices. Generalized rectangular inverses.
Optimized linear programming, rectangular matrices.
Singularity detection and removal within specified tolerances.
Tensors, generalized rank-k outer, inner and dot products.
Arbitrary rank maps, switch for functional or tabular computation.
Relational abstraction, data-type implements relational properties.

STATISTICS

least squares with arbitrary fit function, nth polynomial, exponential, exact fits.

BUILT-IN NUMBERS

Integer:	arbitrary length exact
Rational:	Integer / Integer
Real:	Approximate to any precision
Complex:	Number + Number I

All attributes (base, precision, type, ...) specifiable.

NUMERICAL APPROXIMATION

Under smoothness assumptions, has adaptive sampling, automatic extrapolation, singularity/discontinuity detection.

Point selection specifiable accuracy, recursive depth, precision.

Optimized numerical roots, point convergence (Newton), side convergence (Secant method), numerical minimization.