Mathematica® 5 Reviewer's Guide

This *Mathematica* Reviewer's Guide provides an overview of *Mathematica* technology, history and facts about Wolfram Research, and links to other useful information. You are encouraged to read "What is *Mathematica*: A defining overview" even if you are already familiar with *Mathematica* and will focus on the new capabilities of Version 5 that are covered in "What's New in *Mathematica* 5: Key Points."

What is *Mathematica*?

A defining overview

Mathematica fulfills many different needs for many different audiences. Over the years, it has commonly been labeled in a variety of ways by reviewers—computer algebra system, symbolic calculator, or math package are typical. It is less common to see *Mathematica* referred to as a numerics package, technical documentation system, or programming language, yet these labels represent uses that are just as important to *Mathematica*'s approximately two million existing users.

To encompass all of these capabilities, we have taken to calling *Mathematica* a "technical computing system" —something to aid users through anything from daily tasks to multiyear projects. Often *Mathematica* is utilized to perform tasks that could be performed another way, but are more easily accomplished with *Mathematica*. It is also not unusual for *Mathematica*'s range of abilities to extend the scope of the work being undertaken.

Mathematica can be thought of as having a variety of pieces that contribute to its overall capability: the numeric and symbolic computational engine, graphics system, programming language, and document system. These elements are all tightly integrated and intertwined so that, for example, what seems to be a numerical computation may actually employ symbolic capabilities behind the scenes.

Far from being just for mathematicians, *Mathematica* is involved in every area of technical endeavor. Scientists, analysts, engineers, and educators account for the vast majority of users. Usually *Mathematica* is used directly as it comes out of the box, with its notebook interface. However, it is increasingly being used through alternative interfaces such as a web browser, or by other systems as a back-end computational engine.

As you review *Mathematica*, please be aware of all of these aspects—even if you choose to focus on only a few. Communicating the overall concept of *Mathematica* to potential users, or even to current ones, can be challenging, but their understanding of it greatly enhances their experience and utility of the system.

"As you utilize more aspects of *Mathematica*, the payoff increases exponentially," says Conrad Wolfram, Director of Strategic Development at Wolfram Research. "That's the result of *Mathematica's* consistency and tight integration."

Finding more information

For a more in-depth description of *Mathematica*, visit www.wolfram.com/mathematica.

For further press information about the development of *Mathematica*, including quotes, screenshots, and other information, go to media.wolfram.com/products/mathematica.

What's New in *Mathematica* 5: Key Points

Overview

By industry standards, Wolfram Research uses full digit release nomenclature sparingly. *Mathematica* 5 is the first full digit release of *Mathematica* since *Mathematica* 4.0 in May of 1999. The move to *Mathematica* 5 indicates that this release offers many more significant changes and new features than in the *Mathematica* 4.1 and 4.2 releases of the intervening years.

Mathematica 4's enhancements took *Mathematica* from what was primarily a prototyping environment to an environment capable of a comprehensive workflow, from initial idea through to final simulations. *Mathematica* 5 optimizes *Mathematica*'s capability for large-scale simulation, outperforming dedicated numerical systems (such as MATLAB, MATRIXx, and O-Matrix) on many tasks. At the same time, *Mathematica* 5 retains and improves upon other capabilities, such as the mixed numerics-symbolics approach, flexible document system, programming language, and system interoperability that have made *Mathematica* so renowned.

With *Mathematica* 5, Wolfram Research has initiated a marked transformation in the world of technical computing, providing all the capabilities of both a dedicated numerical package and a programming language in one complete numeric and symbolic environment. *Mathematica* 5 can now do production-scale work in terms of both speed and scalability, without sacrificing any of its broad scope of features. This speed, scope, and scalability make *Mathematica* 5 a uniquely compelling all-around technical system for institution-wide adoption.

Mathematica 5 is primarily an *advanced algorithm* release with a large number of newly developed and refined key technologies, including those that make *Mathematica*'s performance so outstanding. New and improved numerical algorithms and routines allow *Mathematica* 5 to run as fast as or faster than many dedicated numerical software packages and libraries. *Mathematica* 5 also includes a completely new numerical differential equation solver and many other algorithmic improvements.

"In some cases, *Mathematica* 5 is 1000 times faster than previous versions and surpasses the speed of dedicated numerical systems [like MATLAB] too," says Tom Wickham-Jones, Director of Strategic Kernel Development for Wolfram Research. "Yet to achieve this, we've compromised none of the accuracy or expert nature of *Mathematica*. To the contrary, we've enhanced them."

> "The really good news from these results [of our testing] is that *Mathematica* has broken through the 'solution time barrier'! Previous versions of *Mathematica* had solvers which were too slow for professional engineering work. Not anymore! The high speed of *Mathematica* 5 opens up a new world of professional engineering applications. Large problems can now be crunched in times which are competitive with any other code on the market."
>
> *Mathematica* 5 beta tester and engineering director at major defense contractor

Mathematica 5 introduces hundreds of new algorithms and functions, some of which are exclusive to *Mathematica* 5. Others, for example interior point method solvers, provide functionality that was up to now only available in specialized packages costing tens of thousands of dollars.

"The most impressive achievement is the quantity of original research that went into this version—over 100 new algorithms for symbolic and numeric computation have been implemented by in-house developers," says Roger Germundson, Director of R&D for Wolfram Research.

Mathematica 5 also extends Wolfram Research's position as the leader in providing integration with other software and standards. In addition to an updated version of *J/Link*™ and *MathLink*®, *Mathematica*'s standard connections to Java and C/C++, *Mathematica* 5 comes with a fully functional technology preview of *.NET/Link*™, a much requested feature that allows developers to seamlessly integrate *Mathematica* into applications using Microsoft's .NET Framework.

This section only gives a short overview of the changes in *Mathematica* 5. For more information, see the New In *Mathematica* 5 notebook, or the *Mathematica* 5 documentation.

List of Major Improvements

- Record-breaking speed for numerical linear algebra
- Wide-ranging support for fast sparse matrix operations
- New-generation optimized numerical solvers for ordinary and partial differential equations
- Major new algorithms for solving equations and inequalities symbolically over different domains
- Fully integrated solver for differential algebraic equations
- High performance optimization and linear programming including interior point method
- Extensive support for vector and general array variables in numeric solvers
- Industry-leading solver for recurrence equations
- Broader support for assumptions in symbolic computation
- Included *.NET/Link* for full integration with Microsoft .NET framework
- Flexible import and export of DICOM, PNG, SVG, and sparse matrix formats
- Optimized versions for 64-bit hardware and operating systems
- New quick-start interactive tutorial

Performance Enhancements

This section describes key performance improvements in *Mathematica* 5 and explains the importance of each one.

Fast Dense Numerical Linear Algebra

Dense numerical linear algebra is an important building block for most of *Mathematica*'s numerical analysis functionality, ranging from data analysis and matrix operations to numerical differential equation solvers and graphics. In addition to speed enhancements, *Mathematica* 5 also fills out the feature set by adding operations like matrix ranks, norms, generalized eigenvalues, generalized Schur decompositions, generalized singular value decomposition, and many others.

Why is it important?

The performance increases in *Mathematica* 5 affect virtually every aspect of *Mathematica*. On modern microprocessors, *Mathematica* 5 now offers class leading performance for certain kinds of operations—on par with FORTRAN or Matlab code.

High-Speed Sparse Linear Algebra

Fast sparse numerical linear algebra now allows *Mathematica* to solve linear systems with hundreds of thousands of variables within seconds.

Why is it important?

A large number of real world problems deal with sparse matrices, that is, matrices where most of the elements are zero. Examples of operations that involve sparse matrices include solving of ordinary and partial differential equations, optimization problems, and large scale simulations. *Mathematica* 5's implementation of sparse linear algebra is unique in that it allows for arrays of any dimension (or rank) and is fully integrated with the rest of the *Mathematica* system. The performance of basic sparse linear algebra operations in *Mathematica* is now on par with or better than in Matlab and FORTRAN libraries.

Large-Scale Linear Programming

Linear programming is used extensively in economics, engineering, logistics, and resource allocation. *Mathematica* 5 adds solvers using the interior point method, a much more efficient algorithm that can easily solve sparse linear programming problems with up to 1,000,000 variables.

Why is it important?

Up to now this capability was only available in expensive, special-purpose software. *Mathematica* 5 is the only general-purpose computing software to offer this functionality at an affordable price, making large-scale technical computing accessible to a wider audience.

Big Number Arithmetic

Mathematica's big number arithmetic deals with numbers that are too large to be represented by machine precision numbers. Some applications of big number arithmetic include extended precision computations, cryptography, and many number theoretical applications. *Mathematica* automatically selects between machine and big number arithmetic.

Why is it important?

Mathematica has been able to handle numbers with millions of digits for quite some time, but more efficient implementations and better algorithms have improved the performance by a factor of up to 3 for numbers with less than 1000 digits, and by significantly more than that for numbers with millions of digits, by switching to asymptotically more efficent algorithms. The big number performance in *Mathematica* 5 is on par with or faster than that of special purpose libraries and is unchallenged by general computation systems.

64-bit Platform Support

Mathematica 5 is optimized for a large number of 64-bit CPUs and operating systems, including Sun Solaris for Ultra-SPARC, HP-UX for PA-RISC, IBM AIX for the Power architecture, HP Tru64 on Alpha, and Linux on Alpha.

Why is it important?

Apart from speed increases for big number arithmetic due to the larger word size on 64-bit processors, *Mathematica* users who are running increasingly larger computations and applications can now access one million terabytes (1 terabyte = 1,024 GB) of address space instead of the 4-GB address ceiling in 32-bit systems like the current Intel IA-32 (Pentium) architecture. This change also will allow *Mathematica* users to take full advantage of planned performance increases in future versions of these 64-bit processors.

Faster MathLink

Mathematica uses TCP/IP devices for communications between parts of *Mathematica* (for example between the front end and the kernel), as well as for the primary means of communications between multiple *Mathematica* kernels, as in grid*Mathematica* clusters. A new TCP/IP protocol allows *Mathematica* 5 to communicate at the speed of the underlying network. On a standard 100Base-T network, bandwith improves by a factor of ten, latency by a factor of 200. On faster networks, for example Gigabit networks and crossbars found in leading edge computing centers, the gains are even higher. Additional improvements, for example a new `SharedMemory` protocol for Windows platforms, give another tenfold improvement in communication between parts of *Mathematica* on the same machine.

Why is it important?

All *MathLink* bindings to *Mathematica*, like *J/Link* and *.NET/Link*, and almost all `Import` / `Export` formats, rely on *MathLink*. Therefore, these communications will also improve as a result of improvements to *MathLink*.

Numeric Computations

Mathematica 5 contains a number of new and enhanced functions for numeric computation, including:

NDSolve

`NDSolve` has been completely rewritten for *Mathematica* 5, resulting in better performance and much greater flexibility. While many of the enhancements are transparent and will benefit all users, the new `NDSolve` also gives advanced users many more options to select methods, set evaluation options, monitor the progress of the solver, and even implement their own custom solvers with `NDSolve`.

Major changes include:

-New additional solving methods

-More efficient implementation leading to large speed increases for many types of differential equations

-Support for vector and matrix variables

-Ability to solve some differential-algebraic equations

-New options to allow monitoring of the solution progress and more fine tuning of the solving procedure

-Ability to integrate users' custom solving methods into `NDSolve`

FindRoot

`FindRoot` now supports general array variables and includes new and improved algorithms that lead to speed increases and better handling of large-scale problems on computers with limited memory.

FindFit

The new `FindFit` function supercedes and extends the functionality of `Fit` and `NonlinearFit.` It now supports array variables, provides more methods, and can use `StepMonitor` and `EvaluationMonitor` for keeping track of the process.

FindMinimum and FindMaximum

The improved `FindMinimum` and `FindMaximum` functions now support general array variables, offer new and improved algorithms, and work better in limited memory situations.

Symbolic Computations

Mathematica 5 contains a number of new and enhanced functions for symbolic computation, including:

DSolve

`DSolve` can now find all rational function solutions to systems of linear equations with rational coefficients. It can also solve linear systems of differential algebraic equations with constant coefficients—particularly systems of linear equations of the form $A \,.\, x'(t) + B.x(t) == g[t]$, where the matrix A is singular.

RSolve

A recurrence equation (also called a difference equation) is the discrete analog of a differential equation. The general form is $f(n) - f(n-1) - ... - f(n-k) = g(n)$ with n and k being integers.

`RSolve` can solve systems of linear and nonlinear difference equations, difference algebraic equations, and partial difference equations. It can also solve q-difference or divide-and-conquer equations.

Reduce

`Reduce` has been extended to solve equations involving any combination of equalities, inequalities, existential quantifiers, universal quantifiers, and domain specifications.

Resolve

`Resolve` can eliminate quantifiers from arbitrary polynomial systems in complex or real variables, using the same methods as `Reduce`. If obtaining an implicit, quantifier-free form of the system is easier than computing explicit solutions, `Resolve` returns the implicit form. `Resolve` can also eliminate quantifiers involving Boolean variables.

FindInstance

`FindInstance` is a new function that takes the same input as `Reduce`, but gives at most the requested number of different solutions. `FindInstance` works for all problems `Reduce` can fully solve, but for some problems where the complete solution is not needed, finding instances of solutions can be much faster that trying to find the whole solution set. `FindInstance` may also be able to find instances in some problems `Reduce` cannot solve.

Maximize/Minimize

`Maximize` and `Minimize` find the exact global maximum or minimum for a function over a region. These exact optimization functions generally work with polynomial input; however, they can handle some transcendental inputs.

Assuming/Refine

The new function `Refine[`*expr*`,` *assum*`]` gives the form of *expr* that would be obtained if symbols in it were replaced by explicit numerical expressions satisfying the assumptions *assum*.

Import, Export, and Connectivity

Connection Technology

The included *.NET/Link* Technology Preview in *Mathematica* 5 provides full integration with Microsoft's .NET framework. *Mathematica* users can load any .NET object into *Mathematica* and extend it. It also provides an easy way to call any DLL or COM object from within *Mathematica*.

Import and Export

XHTML with CSS

Mathematica has been supporting HTML export since Version 3.0. XHTML with CSS allows customers to preserve the look and feel of their notebooks when exported to the web.

SVG

Mathematica 5 now supports Scalable Vector Graphics (SVG), the standard web language for describing two-dimensional vector and mixed vector/raster graphics in XML.

PNG

Mathematica 5 supports the Portable Network Graphics (PNG) format, a popular bitmap format for the web.

DICOM

Mathematica 5 supports the Digital Imaging and Communications in Medicine (DICOM) standard that is being phased in worldwide to deal with medical images such as xrays and MRI scans in a uniform way. This enables medical professionals and researchers to develop, prototype, and use image processing and analysis applications in *Mathematica*.

SparseArrays

Mathematica 5 supports import and export of files in many standard sparse array formats such as MatrixMarket and Harwell-Boeing.

Tabular Import/Export

Since more and more of our customers process large data sets in *Mathematica*, both import and export of tabular data has been sped up by a factor between 10 and 100 depending on the file structure.

Other Functions

Statistical Plots

The new standard package `StatisticsPlot` offers a variety of plots and charts that are commonly used to gain an overview of data from a statistical perspective, including box-and-whisker plots, Pareto plots, quantile-quantile, and pairwise scatter plots.

Sow and Reap

The functions `Sow` and `Reap` can now be used together to accumulate lists of intermediate results in an evaluation.

Timing Functions

The function `Timing` calculates the amount of CPU time the kernel spends on a calculation. It does not take into account latency or other applications running on the CPU. The function `AbsoluteTiming` provides a type of "wall-clock timing" that measures the total time a command takes until the result is displayed.

New Linear Algebra Functions

New functionality in the area of linear algebra includes generalized eigenvalues, matrix norms, Cholesky decomposition, new singular value operations, characteristic polynomials, and matrix rank.

Algebraic Number Objects

Algebraic objects are now supported and can be used in other functions as well. *Mathematica* 5 represents algebraic numbers as `Root` objects. A `Root` object contains the minimal polynomial of the algebraic number and the root number—an integer indicating which of the roots of the minimal polynomial the `Root` object represents. High performance arithmetic can also be carried out with algebraic objects.

Authoring and Presentation

AuthorTools, first introduced in *Mathematica* 4.2, has been expanded and now includes tools for fixing corrupted notebook files and for comparing differences between notebooks. *Mathematica* 5 also adds a new authoring palette for slide shows and an improved slide show environment for all style sheets.

What Is Unique about *Mathematica*?

What makes *Mathematica* unique is the host of innovative technology that underlies it. Year after year, Wolfram Research has led the technical computing field by engineering a more accurate, easier-to-use, and wider range of *Mathematica* functionality—achieving many firsts along the way.

Some of the most sophisticated *Mathematica* technology isn't immediately apparent when you start out. It's under the surface—for example, choosing algorithms, checking precision, or formatting your output appropriately.Yet, *Mathematica* technology ensures that as the going gets tougher, so does the margin by which *Mathematica* outperforms other systems—at times being the only system able to deliver an accurate answer.

Discover *Mathematica*'s advantages at the outset by comparing technologies described in this section with those of other systems. Then decide which system is a better and more reliable investment.

Automatic Algorithm Selection

With automatic algorithm selection, you choose the task you want performed, and *Mathematica* picks the best algorithm(s) for performing it.

For example, you might want to solve a differential equation numerically. With *Mathematica* you would use the function `NDSolve`, which would "decide" which of its dozens of algorithms to deploy to get you an accurate answer quickly (you could also choose to override this and select manually). With a traditional system you would need to know which function name (e.g., `ode113`, `ode23e`) would best solve your problem, and you would select the algorithm yourself.

As well as picking an algorithm at the start of a calculation based on your input, *Mathematica*'s automatic algorithm selection can change its selection in midcalculation, based on the success of the current method, or preemptively as an optimization for the next stage. This capability means that automatic algorithm selection can usually outperform an individual manual selection of algorithms.

Nevertheless, the key benefit of automatic algorithm selection is that it enables users to quickly get accurate results to problems for which they do not have a specialist's algorithmic knowledge. In practice, this makes a dramatic difference in the range of successful computations that most users can perform and is becoming increasingly important as algorithmic knowledge becomes more specialized and as the breadth of available computations in software packages increases.

An additional important feature of *Mathematica* that is implemented with automatic algorithm selection is its ability to determine whether an input contains symbols, exact numbers, or approximate (possibly arbitrary-precision) numbers. Appropriate algorithms are selected automatically for each case, producing a result that matches the input type. For example, if symbolically specified equations are given to `Solve`, *Mathematica* will attempt to produce a symbolic result; if machine-precision input is given, `Solve` will utilize appropriate numerical algorithms and attempt to produce a machine-precision numerical result. The user does not need to use a different function call in these different cases.

Mathematica pioneered wide-scale implementation of automatic algorithm selection at its release in 1988. Since then, the range of algorithms, the sophistication of selection, and the number of functions for which automatic algorithm selection operates have all greatly increased. No other technical system today offers this approach.

Notebook Document-Centered Interface

Mathematica notebooks are today's most sophisticated manifestation of the document-centered approach to user interfaces and are a departure from the normal dialog box-based approach.

Traditionally, graphical user interface (GUI) software uses dialog boxes for actions and documents for user data on which those actions operate. Dialog boxes are distinguished as having nonscrolling, fixed layouts of buttons, menus, and so on, while documents are scrolling, increase in size as necessary, and have interactive structure and updatable content.

With a document-centered interface (DCI) approach, the actions, control elements for them, and structural information all reside together with the user data in the document itself.

For technical users this approach is especially beneficial. Technical-user data is highly complex in structure and content compared to the linear textual structure of a normal document. Ideally, a technical document must be "alive" with editable 2D typeset mathematical expressions, transformable graphics, and automatic formatting of results as they emerge. In essence, technical documents require actions to occur from within the document; the barrier between actions and documents in the traditional GUI approach is highly detrimental to efficient workflow. This is particularly the case for collaborative work; with a DCI approach, actions are embedded in any document and can be sent to others to reapply or adjust.

Standard HTML web pages are an example of a simple form of a document-centered interface, providing structure, links, and input boxes but lacking sophisticated interactivity and other elements. More recently, XML has provided a far more extensive structure for document-centered interface specification—in particular, supporting MathML and SVG, features relevant to the technical community.

Mathematica notebooks, first released in 1988, fully exploit the DCI approach. The notebook interface combines a word processor-like foundation with a clearly defined notion of "cells," which are arranged vertically in a scrolling window like paragraphs of text.

The cells are important because they visually and functionally segregate the text into inputs, outputs, text, graphics, headings, and so on. Yet, all components of *Mathematica* notebooks are still simply expressions in the *Mathematica* language. Therefore, unlike other more-restrictive interface models, the cells are flexible enough to support any type and size of expression, afford easy editing and insertion of contents, and are easily expandable for large calculations and documents.

As well as providing an optimized environment in which individuals can perform technical work, the notebook structure has proven to be an extremely effective tool for writing comprehensive reports and presentations of results. With most application software there is a huge gulf between users and developers. In *Mathematica*, as users work on a problem, they are automatically creating the outline of a (notebook) document that can become a useful tool for themselves or others to solve similar problems in the future.

Often with minimal revision and annotation, users can turn their raw work into notebooks that can be sent to colleagues who, in turn, can change the input, tweak the algorithm, and in very little time investigate problems of their own. In this way, a "user" has in effect become a developer of a tool that others can use.

Users can also learn to manipulate features of *Mathematica* that allow them to add buttons, palettes, and other user interface elements into their documents as the need and interest arise. But even a simple notebook is often a powerful, flexible piece of application software in its own right—an important benefit of the DCI approach.

Notebooks enable a wide range of collaborative and interactive workflows between, for example:

- Researchers testing each other's results
- Teachers setting up structured course work for their students
- Workgroup members working on a technical report in which they change parameter values, reevaluate calculations, and regenerate graphics

Depth of algorithmic knowledge

Mathematica contains thousands of functions covering many areas—numerical computation, symbolic computation, graphics, and general programming. Its collection of mathematical algorithms alone covers most published algorithms and also contains a significant number of proprietary algorithms.

These proprietary algorithms are the product of over 16 years of intensive research and development within Wolfram Research itself. The mathematicians and computer algorithm specialists on our staff are active participants in the latest advances and developments in their areas, and they work vigorously to integrate cutting-edge knowledge and research into each new version of *Mathematica*.

The sheer number of built-in algorithms alone would make *Mathematica* a leading technical computing package, but the number of algorithms is only a small part of what makes *Mathematica*'s knowledge base so powerful. A unique feature of *Mathematica* is that data and programs in *Mathematica* are all the same thing: symbolic expressions. This means that any *Mathematica* function can provide input for, or accept output from, any other relevant *Mathematica* function.

This feature allows *Mathematica* functions to combine different algorithms and methodologies to create optimal results. For example, `NDSolve`, *Mathematica*'s function for numerical differential equation solving, initially analyzes the systems of differential equations symbolically, transforms them into a form optimized for numerical computation, and chooses the algorithm that gives the best solution. *Mathematica* then compiles the equations for maximum efficiency before running the numerical solver. During the evaluation, *Mathematica* constantly analyzes the solution process and switches between stiff and nonstiff solvers as appropriate. This automatic algorithm selection process is another of *Mathematica*'s unique technologies.

Having a vast collection of algorithms that fit together through *Mathematica*'s symbolic programming paradigm means that new and sophisticated algorithms can often be implemented with a minimum of effort since they can draw on many existing algorithms. In fact, many *Mathematica* functions combine subalgorithms never previously attempted: numeric functions that use symbolic algorithms (e.g., to recognize nonlinear least squares problem for `FindMinimum` or to symbolically compute derivatives or gradients) and symbolic functions that use numeric algorithms (e.g., to safely prove numeric inequalities).

All algorithms are packaged into *Mathematica* functions according to what they do, not how they do it. This means that *Mathematica* users do not have to know the algorithms or their structure, areas of applicability, or limitations to make efficient use of them.

gigaNumerics™

gigaNumerics represents the unique set of *Mathematica* technologies that deliver high-speed numerical computations. Unlike dedicated numerical systems, *Mathematica* is known for its generality and accuracy checking. These characteristics would normally impose speed penalties on numerical calculations since a number of extra operations have to occur each time a calculation is executed. Initially, *Mathematica* checks the input to determine whether it should be handled symbolically with machine- or extended-precision arithmetic. Accuracy checks are made during the calculation, and the accuracy of the calculation is stepped up if necessary. Over- and underflows are sensed and handled correctly.

Traditional numerical systems fail to carry out these procedures. Yet they are critical to your getting the right answers without being a numerics expert, and they contribute to making *Mathematica* the most accurate and generally applicable system available.

The challenge Wolfram Research tackled with *gigaNumerics* was to achieve exceptional raw computing speed while maintaining *Mathematica*'s generality and accuracy. This challenge was met successfully by the following combination of *gigaNumerics* technologies developed at Wolfram Research.

Precompilation

Compilation can speed up numerical calculations for certain types of input. *Mathematica* optimizes its performance and efficiency by preapplying compilation automatically as a transparent part of many numerical calculations in cases in which *Mathematica* assesses that it is feasible.

Packed Arrays

Computations to be performed on machine-precision matrices and arrays are analyzed to decide whether packing them into a specialized format will improve the performance of the computation. This process of analysis and application occurs transparently, with outputs presented the same way regardless of which methodology *Mathematica* chooses.

Automatic Algorithm Adaptation and Selection

Many *Mathematica* functions automatically choose between a variety of algorithms and, in addition, adaptively adjust their sampling rate throughout the calculation to optimize speed and accuracy.

Processor Optimization

Libraries are optimized for each processor, including the latest 64-bit varieties.

Symbolic Preprocessing

In some cases the total calculation time is smallest if you simplify a problem algebraically before evaluating the result numerically. *Mathematica* employs this technique automatically where appropriate.

Vectorization

Certain *Mathematica* operations can work on an entire vector, matrix, or array rather than on just a single element. Operating on all the data at once reduces the number of top-level calls to *Mathematica*, replacing them with optimized internal routines.

For the first time with Version 5, *Mathematica* outperforms traditional dedicated numerical systems in terms of raw computational speed alone. Advances in *gigaNumeric*s technologies have achieved this–they more than cancel out the speed deficit that might be expected from the generality and accuracy that *Mathematica* delivers. In the future, *Mathematica*'s lead is expected to increase because traditional numerical systems do not have integrated symbolic capabilities with which to perform symbolic preprocessing.

Symbolic Programming

Mathematica is widely known as the world's most powerful system for technical computing. What is less widely known is that *Mathematica* is also a uniquely powerful programming language based on symbolic programming—the unifying idea that every element can be represented as a symbolic expression.

When Stephen Wolfram first began to design *Mathematica* in the mid-1980s, he saw that no existing programming paradigm could support everything he wanted to do. Convinced from his discoveries in science that a much more powerful paradigm should be possible, he built on disparate ideas from computer science, logic, and mathematics to create the new paradigm of symbolic programming.

In this paradigm all different kinds of objects—formulas, lists, data, and graphics, to name a few—are represented in a uniform way as expressions. A prototypical example of a *Mathematica* expression is `f[x]`. This expression can represent a mathematical function, a graphic, a sound, a program, or even a complete *Mathematica* notebook. Functions can be both input and output of another function, enabling very concise and simple coding. Also, since algorithms can be parameterized not only by numbers or some fixed number of parameters but also by functions, algorithms are infinitely more flexible.

Another key feature of *Mathematica*'s programming language is the ability to write programs that generate or manipulate other programs, commonly known as metaprogramming. In *Mathematica*, any expression can be generated programmatically at run-time. For example, it is entirely possible to create and manipulate *Mathematica* notebook documents algorithmically, a feature many of our customers now use to generate customized reports, web pages, and even printed marketing materials automatically.

The symbolic programming paradigm has served as the foundation for *Mathematica* since its first release in 1988. Over the past 14 years, the programming language embodied in *Mathematica* has been used in an immense number of technical computing applications and has become well integrated into many areas of technical education. Now what is emerging is the use of the symbolic programming capabilities of *Mathematica* as the basis for a new generation of implementation strategies for general computing applications.

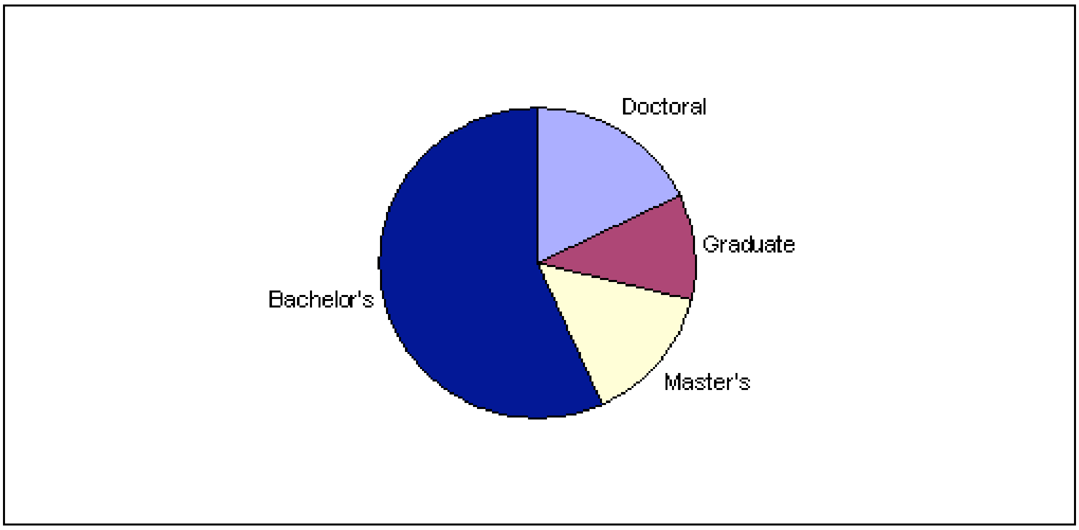
In the last couple of years, symbolic programming has come to the forefront of computing as the next large-scale change in programming paradigms, with the last having been object-oriented programming. An example of a new technology that draws a number of design concepts from symbolic programming is Extensible Markup Language (XML), the new universal standard for machine-to-machine communication.

Like *Mathematica*, XML provides a uniform way to represent arbitrary objects, whether they are data structures, documents, or even program code. Both ways of representation are basically trees of expressions called *Mathematica* expressions and XML documents, respectively. In *Mathematica* these expressions are operated on by transformation rules, and in XML they are operated on by programming methodologies such as Extensible Stylesheet Language Transformations (XSLT) and the Document Object Model (DOM).

Mathematica's rich symbolic programming language was designed from the ground up for manipulation of structured expressions, and operations that can be expressed naturally in a single line of *Mathematica* input are generally much more difficult to write in Java or XSLT. This fact alone makes *Mathematica* an ideal tool for dealing with XML data and documents.

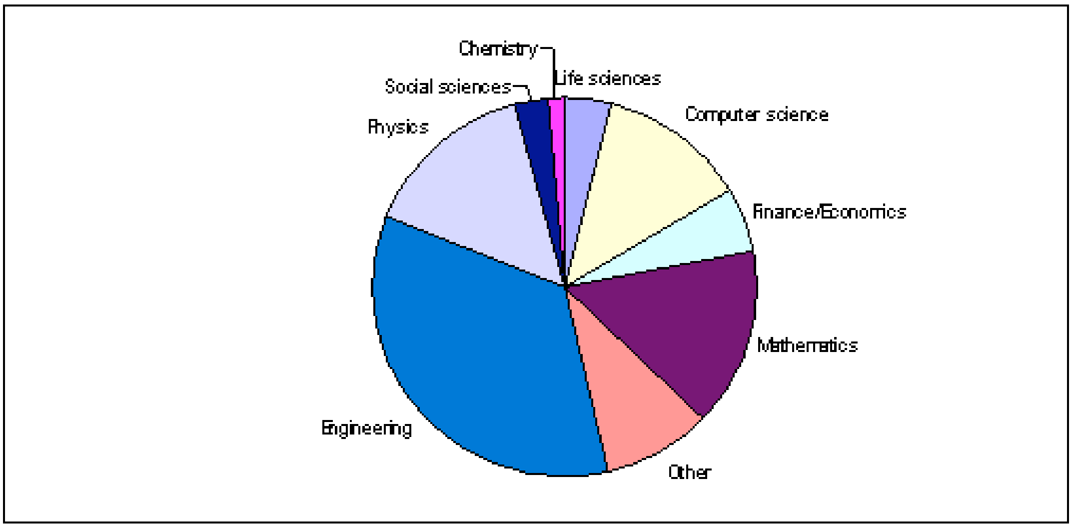
Who Is Using *Mathematica* Technology?

Professional users by last-completed degree



As this chart demonstrates, the majority of professional *Mathematica* users possess a bachelor's degree—one does not need an advanced degree to become an expert *Mathematica* user. Although this chart highlights professional users of *Mathematica, Mathematica* has a very large population of student users as well.

Professional users by field



Mathematica users span a broad diversity of fields, with the largest user population coming from the engineering disciplines. Although it is sometimes mistakenly assumed that *Mathematica* is only for mathematicians, in actuality they are only 15% of *Mathematica* users.

A sampling of users

Mathematica continues to have a strong user base in traditional professional markets for technical computing software—such as national labs, military and civilian research organizations, and industrial R&D departments, as well as in traditional academic markets. Recent trends indicate increasing demand for nontraditional applications of *Mathematica*, as well as rapid expansion in postsecondary and precollege education.

Mathematica users represent a broad cross-section of the following:

- The vast majority of Fortune 500 corporations, including The Boeing Company, General Electric, and Intel Corporation

- Over a dozen key government agencies, including NASA, the CDC, and the U.S. Patent Office

- All major national and international research labs, such as SLAC, INRIA, and the German Fraunhoffer Institute

- Virtually all of the top 25 universities as ranked in U.S. News and World Report

- Almost 400 universities total in the U.S. and Canada, and approximately 800 universities worldwide

- Numerous system-, state-, and even countrywide university systems in the U.S. and abroad, including the California State University system, the Pennsylvania State University system, the City University of New York system, and the Israeli MACHBA consortium

About Wolfram Research

Wolfram Research, Inc., was founded in 1987 by Stephen Wolfram and released the first version of *Mathematica*, its flagship product, on June 23, 1988. The release of *Mathematica* 5 on June 23, 2003, coincides with the product's 15th anniversary. Over the last 15 years, the award-winning *Mathematica* has been adopted in an unprecedented range of fields in industry, government, and academia. In fact, it has been responsible for bringing advanced mathematics and computing to fields that were traditionally less technical, and in so doing has substantially increased the market for technical software in general. A growing industry of applications, consulting services, books, and courseware serves the international community of nearly two million *Mathematica* users worldwide.

Although *Mathematica* remains the company's flagship product, it is now part of an expanding product family built on a common technology base. At the time of the *Mathematica* 4.0 release in 1999, *Mathematica* was the only stand-alone product from Wolfram Research. During the last three years, the company has broadened its horizons to become a multiproduct corporation offering organization-wide technical computing solutions. By the end of 2003, the Wolfram Research product family will include all of these independent titles: *Mathematica,* web*Mathematica,* grid*Mathematica, CalculationCenter®, Mathematica Teacher's Edition, The Mathematical Explorer, Publicon®, A New Kind of Science™ Explorer*, *A New Kind of Science Explorer Mathematica Kit*, and *Calculus WIZ®*.

A full company background and history of Wolfram Research, including a history of *Mathematica* and information about corporate structure, grants and sponsorships, web resources, and other company contributions to research and education, is available online at www.wolfram.com/company.

Suggestions for Reviewing *Mathematica*

There are many possible angles for writing about *Mathematica*. We have listed below a few with which we are familiar. If you would like us to assist you with these or any other subjects, please contact our Public Relations department. They can also put you in contact with experts for discussion or interview on these topics, either at Wolfram Research or from among our users.

Frequent review topics

Evaluation of the newest version of Mathematica

Comparison of Mathematica and other technical software

Using Mathematica as a calculator or computation system

Mathematica use in the classroom

Mathematica and the Stephen Wolfram story

Less frequent topics

Mathematica as a programming language

May involve discussion of:

- the diversity of *Mathematica*'s language structures

- the best way to learn *Mathematica*'s language

- how *Mathematica* compares with other languages

- what it means that *Mathematica* is a "symbolic language"

- example programs

Mathematica as a software component/back-end

May involve discussion of:

- for which applications *Mathematica* is particularly suitable

- web versus local delivery

- custom interfaces to *Mathematica*

How Mathematica affects the way math is taught

May involve discussion of:

- how *Mathematica* makes teaching math in an experimental/observational way possible

- how *Mathematica* is used in the classroom

- to what extent students should be exposed to *Mathematica* as a tool versus an instructional aid

- whether *Mathematica* helps or hinders math education

Mathematica as an automated document creation system

May involve discussion of:

- *Mathematica* as a fully fledged technical documentation system

- how live computation and traditional publishing capabilities are combined

- XML

- how documents relate to the rest of the *Mathematica* system

Mathematica as a cost-effective solution for technical computing

May involve discussion of:

-how *Mathematica* includes algorithms that are otherwise only available in expensive packages

-cost-effectiveness with respect to purchase price as well as work hours and/or specialized skills

-how writing original routines is not always less expensive

-rapid development cycles thanks to decreased programming and run time

-how typical *Mathematica* programs are only 5-10 percent of the size of those created in traditional languages or numerical systems

Places to go for information

Media resources, images, and logos

media.wolfram.com

Press releases and other Wolfram Research news

www.wolfram.com/news

General information about Mathematica

www.wolfram.com/mathematica

www.wolfram.com/mathematica/qa.html

New features in Mathematica 5

www.wolfram.com/mathematica/newin5

Revision history since Mathematica 3.0

www.wolfram.com/mathematica/history.html

Information about Wolfram Research products

www.wolfram.com/products

Facts and figures

Purchase programs and licensing options

All Wolfram Research products are offered with multiple licensing and discount options, including volume purchase agreements, concurrent-use lease agreements, and licenses for home, web, and intranet use. Unlimited use agreements are also available for large organizations. Under all purchase programs, multiple licensing options are available, including licenses for single machine use, concurrent use through license managers, and web licenses.

Weird and wonderful

Total weight of all Mathematica books shipped

65 metric tons

Estimated person hours of development time

About 1.4 million since 1988

Mathematica builds per day

Wolfram Research’s automated build and test system processes about 130 distinct builds per night, including about 90 builds of *Mathematica* components for about 15 distinct operating systems. The total size of the code generated in every round is in the range of 100GB.

Most distant Mathematica delivery ever

A replacement copy of *Mathematica* was delivered to the space station MIR.

U.S. state with the most Mathematica licenses

California

European country with the most Mathematica licenses

Germany

Country outside of the U.S. with the most Mathematica licenses

Japan

Requests served by all Wolfram Research websites combined

About 300,000,000 per year

Press contact information and review copies

Andreas Heilemann

Marketingleiter ADDITIVE GmbH Engineering+Science

andreas.heilemann@additive-net.de

Tel: ++49-(0)6172-5905-29