Mathematica® 5.2 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*?" even if you are already familiar with *Mathematica* and plan to focus on the new capabilities of Version 5.2 that are covered in "What's New in *Mathematica* 5.2: Key Points" and on our website.

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 the utility of the system.

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

Finding More Information

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

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

What's New in *Mathematica* 5.2: Key Points

This section gives a short overview of the new features and enhancements in *Mathematica* 5.2 from a reviewer's perspective. For more information, see the What's New in *Mathematica* 5.2 section on the *Mathematica* website (www.wolfram.com/mathematica/newin52) and the *Mathematica* 5.2 documentation.

Overview

As stated in the press release, *Mathematica* 5.2 continues to lead the industry in speed, scope, and scalability by bringing 64-bit technology to all supported platforms including desktop systems running Windows XP Professional x64 Edition or Mac OS X 10.4 Tiger—an industry first. More than 4.3GB of memory (the 32-bit limit) can now be addressed, and high-precision or large numbers are processed in 64-digit rather than 32-digit chunks for faster computation.

With the growing number of multicore and multiprocessor computers available, Wolfram Research has embarked on a major new initiative to parallelize *Mathematica*. *Mathematica* 5.2 is the first version with numerical linear algebra multithreading capabilities for all platforms. In addition, it also utilizes the vector processing units in modern microprocessors to speed up repetitive computations by a large margin.

"*Mathematica* is now the ideal environment for large computations or simulations," says Tom Wickham-Jones, Director of Kernel Technology. "From 5.0 onwards we've dramatically speed-up computation, reduced memory usage, and introduced grid computing; now with 5.2 we've enabled *Mathematica* to take advantage of more memory and more CPUs for computation too."

Mathematica 5.2 can do production-scale work in terms of both speed and scalability, without sacrificing any of its broad scope of features. This, combined with *Mathematica*'s long-standing strengths like deep algorithmic knowledge, ease-of-use, broad connectivity, and unique automatic algorithm selection makes 5.2 an unrivaled compelling all-around technical system for organization-wide adoption not only in technical fields, but also for applications requiring sophisticated data analysis such as bioinformatics, network analysis, or financial institutions.

List of Improvements

There are many new functions, toolkits, and performance improvements in *Mathematica* 5.2, including:

- All-platform support for 64-bit addressing
- Multicore support on major platforms
- Multithreaded numerical linear algebra
- 64-bit-enhanced arbitrary-precision numerics
- Vector-based performance enhancements
- Automatic installation selection
- Bundled notebook indexing for desktop search
- Support for ssh security for remote kernels
- vCard and RSS import
- New algorithms for symbolic differential equations
- Enhanced performance for linear Diophantine systems
- Enhanced quadratic quantifier elimination
- Singular-case support for high-level special functions
- Enhanced statistics charts

- Mathema*ticaMark™ 5.2* covering grids & clusters

And many more enhancements & innovations ...

Some New Features in Detail

All-platform Support for 64-bit Addressing

Across all major platforms, *Mathematica* now supports 64-bit memory addressing and 64-bit long number partitioning—both industry firsts that reflect Wolfram Research's commitment to delivering rapid support for the latest computing technology.

Traditionally, operating systems have been 32-bit, providing unique addresses for up to only 2^{32} bytes or about 4.3GB of memory. Now, UNIX, Linux, and new operating systems from Microsoft and Apple utilize 64-bit addressing making the memory limit 2^{64} bytes or about 18,000,000,000GB, although current hardware will only support a lower limit such as 2^{42}.

Why is This Important?

64-bit support lets *Mathematica* users solve much larger problems or solve problems with higher resolutions.

Starting with *Mathematica* 5, Wolfram Research has focused on making *Mathematica* the best system for solving large-scale problems, offering the scope of *Mathematica* with the speed and scalability of a dedicated numerical system.

Mathematica 5—released in June of 2003—offered class-leading performance for dense numerical linear algebra, a key building block for numerical functionality. These performance increases affect uses such as data analysis, numerical differential equation solving, optimization, and graphics. In most cases, computational speed now matched or exceeded Fortran/ MATLAB code, while maintaining *Mathematica*'s breadth of usability.

Mathematica 5.1—released in November of 2004—added a host of new capabilities, especially for working with large-scale, diverse types of data, including much faster import and export filters.

Now that *Mathematica* was being used as a replacement for specialist numerical systems, the final bottleneck was the so-called 4GB barrier. Although Wolfram Research has been pioneering full support for Unix and Linux-based 64-bit operating systems for years, Windows and Mac OS X only offered 32-bit addressing until recently. *Mathematica* 5.2, together with the releases of Windows XP Professional x64 Edition and Mac OS X 10.4 Tiger, removes this important limitation.

While external libraries still limit *Mathematica* 5.2 to roughly 4.3 billion elements per expression (or variable), or about 16 gigabytes for an individual complex matrix, *Mathematica* can perform operations on variables that require many gigabytes or terabytes of memory. The 4.3 billion elements per expression will be removed in a future version of *Mathematica*.

Multicore Support on Major Platforms

Mathematica 5.2 introduces new all-platform support for threading of numerical linear algebra over multiple CPU or multicore computers.

In addition, *Mathematica*'s notebook front-end is a separate process from its computational kernel, allowing them to run on separate cores or CPUs—giving a responsive interface even when the kernel core is under full load.

Multicore chips have more than one CPU core; multiple processor computers have more than one CPU chip. Both are being introduced by manufacturers to speed up tasks by splitting threads or processes up between different processors so that they can be performed in parallel.

Optional grid*Mathematica*™ is available for tying together multiple computers, each containing one or several CPUs and/or cores.

Why is This Important?

While Moore's Law (which states the number of transistors on a piece of silicon will double every 18 months) is still valid, CPU clock speed hit the so-called "heat wall" in late 2003. Since then, clock speed has only gone up marginally. The major CPU manufacturers chose multicore processors as a way out of this performance speed bump and by mid-2006 all mainstream systems will have multicore CPUs. *Mathematica* is the first technical software to take advantage of multicore processors by using highly optimized numerical linear algebra libraries that utilize all available processors and processor cores to speed up computations, and by running the front end and the kernel on separate cores. This enables users to fully utilize these exciting new systems.

Vector-based Performance Enhancements

Major speedups have been attained on key platforms when applying elementary functions to vectors, matrices, and arrays of floating point numbers.

Packed array technology—introduced in *Mathematica* 4—achieves this by utilizing vectorized math libraries optimized by CPU. On certain platforms these libraries use multicore technology.

Why is This Important?

Many computations in *Mathematica* require evaluating the same expressions thousands of times; examples include plotting, numerical integration, and solving of differential equations. Utilizing the vector units of modern processors like the Pentium 4 can speed up these computations significantly, improving performance throughout *Mathematica*.

Mathematica and Large Problems—a Summary

With 5.2, *Mathematica* is now the ideal platform for solving large problems:

- Universal 64-bit support means that there's effectively no memory barrier.
- Sparse and packed array technology introduced in *Mathematica* 4, 5.0 and 5.1 made computations highly memory efficient.
- Computational speedups since *Mathematica* 5 have improved some calculation times as much as 1000 fold. *Mathematica* 5.2 continues to outperform specialized numerical systems for many computations, while providing unmatched scope and scalability.
- Large speedups in import and export filters in *Mathematica* 5.1 removed final bottlenecks in the reading and writing of large data sets.
- Optional grid versions of *Mathematica* are available to distribute computations in parallel over multiple processors or computers.

In a sentence: *Mathematica* 5 introduced the computational power to analyze large amounts of data; 5.1 added the ability to read and write large binary data sets in near real time; and *Mathematica* 5.2 breaks the 4GB memory barrier on the desktop.

Bundled Notebook Indexing for Desktop Search

The Wolfram Notebook Indexer has been included with *Mathematica* 5.2. After autoinstalling the correct plug-in for Google Desktop Search, Apple Spotlight, or the Windows Desktop, the Indexer parses notebook expressions so that *Mathematica* expressions, control language and arbitrary defined Unicode characters including Japanese, Chinese, and other 16-bit characters can be searched for.

The Indexer supports extra features in certain indexers such as special category identifiers in Spotlight.

Why is This Important?

As Google explains on its site, “Desktop search is how our brains would work if we had photographic memories.” By making every notebook on your computer searchable, the Wolfram Notebook Indexer, together with the main desktop search utilities, puts your information easily within your reach and gives you additional options to find your *Mathematica*-based work quickly, even if you have thousands of notebooks and want to see, for example, only the ones that refer to “water table estimations.”

Support for SSH Security for Remote Kernels

Mathematica’s computational kernel can be run on a remote computer from the user and front-end notebook interface. This is advantageous in cases where a remote higher-powered computer is available.

New in 5.2, *Mathematica* can now communicate through a secure shell environment that is typical in many organizations, rather than just insecure connections such as remote shell.

Why is This Important?

More and more of our customers are relying on the increasingly available clusters and grids to speed up their biggest computations using grid*Mathematica* or *Parallel Computing Toolkit*. Increasing security concerns throughout IT departments have led many of them to rely on secure shell and other authentication and encryption technologies to secure their networks. Built-in ssh security in *Mathematica* 5.2 makes this much easier.

Additional Computational Enhancements

Like any previous release, *Mathematica* 5.2 also includes numerous additions and enhancements to its mathematical capabilities, algorithm base, and other functionality. These enhancements include new algorithms for symbolic differential equations, enhanced performance for linear Diophantine systems, enhanced quadratic quantifier elimination, singular-case support for high-level special functions and enhanced statistics charts, as well as many under-the-hood improvements in stability and performance.

Why is This Important?

As more methods have been added to *Mathematica*’s built-in functions, it allows our users to solve more and more problems—such as differential algebraic equations, piecewise equations, and first integrals of nonlinear PDEs—that cannot be solved in any other system. Thanks to *Mathematica*’s unique automatic algorithm selection, they can do so automatically, without having to learn new commands or even being aware of the new functionality at all.

These new methods are all fully integrated into *Mathematica*’s automatic algorithm selection, so users can take advantage of the functionality without having to learn new commands, or even being aware of it at all.

MathematicaMark 5.2 Covering Grids & Clusters

MathematicaMark 5.2 provides scalability for *Mathematica* benchmark scores, allowing benchmarking of single-CPU systems as well as SMP machines, clusters, and grids.

Why is This Important?

The performance that many users demand when using *Mathematica* to solve their largest and most complex computations can press hardware to its limit. *MathematicaMark* creates quick benchmark reports that show how a particular computer compares to other machines for overall *Mathematica* performance as well as typical computations and processes, giving existing and potential customers an accurate estimation of how well their computer performs and whether or not investing in different hardware would be advantageous.

MathematicaMark is also of interest to computer manufacturers and reviewers because it tests and compares a wide range of capabilities, so it provides a better measure of real-world performance as opposed to a benchmark that tests some single component like floating point performance, or memory throughput with synthetic problems.

Because of its platform independence, *MathematicaMark* and its wide range of tests also provide a level playing field upon which to compare different computers and operating systems. And because Wolfram Research is equally dedicated to building *Mathematica* for all of its available systems, *MathematicaMark*'s results are based strictly on performance, without favoring any one platform or system over another.

Many Additional Innovations

In addition to the major highlights outlined earlier, *Mathematica* 5.2 comes with many additional improvements and new functions, ranging from new combinatorial operations like computations of tuples and subsets, tools for detecting branch cuts in complex functions, support for vector derivatives, and additional matrix decompositions. A full list of improvements is available from Wolfram Research. If you are interested in reviewing any of these additional features, please contact Ben Wilson, Media Relations Manager, at press@wolfram.com, or by calling +1-217-398-0700, ext. 3189.

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 mid-calculation, 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 DCI, providing structure, links, and input boxes but lacking sophisticated interactivity and other elements. More recently, XML has provided a far more extensive structure for DCI 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 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 coursework for their students
- Workgroup members working on a technical report in which they change parameter values, re-evaluate 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 problems 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 pre-applying 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.

Mathematica 5.2 outperforms traditional dedicated numerical systems in terms of raw computational speed alone. Advances in gigaNumerics technologies 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 runtime. 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 16 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.

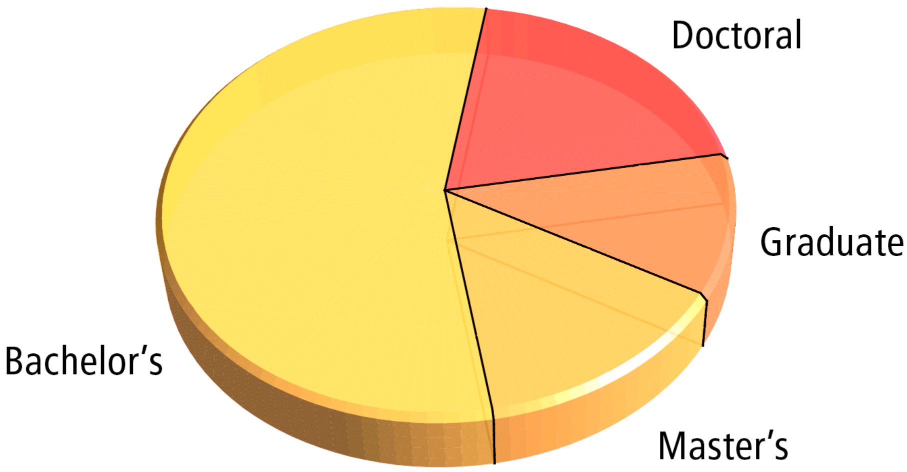
In the last couple of years, symbolic programming has come to the forefront of computing as the next large-scale change in programming paradigms, with the last having been object-oriented programming. An example of a new technology that draws a number of design concepts from symbolic programming is Extensible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

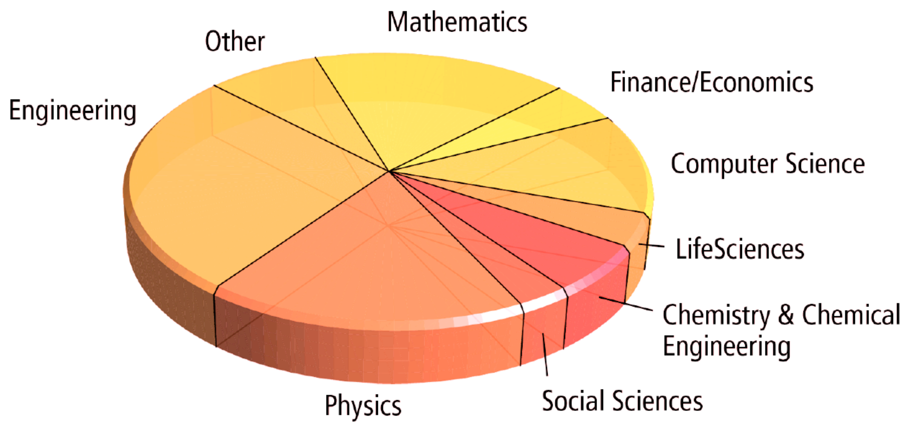


As this chart demonstrates, the majority of professional *Mathematica* users possess a bachelor’s degree—one does not need an advanced degree to become an expert *Mathematica* user. Although this chart highlights professional users of *Mathematica, Mathematica* has a very large population of student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

Professional Users by Field

WIDE RANGE OF USERS

Across the world there is a growing range of uses in all fields of science and industry to which *Mathematica* is put:



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 less than 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 the top 50 universities as ranked in *U.S. News and World Report*
- Almost 700 universities total in the U.S. and Canada, and thousands more 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, the State Baden-Württemberg, the state Thüringen and the Israeli MACHBA consortium.

About Wolfram Research

Wolfram Research is the world's leading developer of computational software for science and technology, offering organization-wide computing solutions. Led by *Mathematica*, its flagship product, the company's software is relied on today by several million enthusiastic users around the world and has been the recipient of many industry awards. Wolfram Research was founded in 1987 by Stephen Wolfram, who continues to lead the company today. The company is headquartered in the United States, with offices in Europe and Japan.

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 the Ultimate Data Analysis Tool

May involve discussion of:

- new data import and export options
- web services support and built-in database connectivity
- the range of analysis tools available
- the ease of developing new tests

Mathematica as a Software Component/Back-End

May involve discussion of:

- applications for which *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 runtime
- how typical *Mathematica* programs are only 5–10% of the size of those created in traditional languages or numerical systems

Mathematica as a Tool for Advanced Business Analytics

May involve discussion of:

- how *Mathematica* helps business analysts to process the ever-growing amounts of data
- exploring new algorithms and visualizations to help mine corporate data more efficiently
- cost-effectiveness with respect to purchase price as well as work hours and/or specialized skills
- how *Mathematica*'s extended precision and advanced algorithms help minimize errors in business analytics

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 5.2

www.wolfram.com/mathematica/newin52

Information about Wolfram Research Products

www.wolfram.com/products

Facts and Figures

Prices and Versions

unchanged since 3 years

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.

Press Contact Information and Review Copies

Andreas Heilemann

Produktmanager *Mathematica*

andreas.heilemann@additive-net.de

Tel: 06172-5905-29 (36)

© 2005 Wolfram Research, Inc. *Mathematica*, *CalculationCenter*, *MathLink*, and *Publicon* are registered trademarks of Wolfram Research, Inc. *Calculus WIZ*, gigaNumerics, *J/Link*, *MathematicaMark*, and *.NET/Link* are trademarks of Wolfram Research, Inc. *A New Kind of Science* is a trademark of Stephen Wolfram, LLC. All other trademarks are the property of their respective owners. *Mathematica* is not associated with Mathematica Policy Research, Inc. or MathTech, Inc.