What Is Mathematica?

Mathematica is a general computer software system and language intended for mathematical and other applications.

You can use Mathematica as:

- A numerical and symbolic calculator where you type in questions, and Mathematica prints out answers.
- A visualization system for functions and data.
- A high-level programming language in which you can create programs, large and small.
- A modeling and data analysis environment.
- A system for representing knowledge in scientific and technical fields.
- A software platform on which you can run packages built for specific applications.
- A way to create interactive documents that mix text, animated graphics and sound with active formulas.
- A control language for external programs and processes.
- An embedded system called from within other programs.

Mathematical computations can be divided into three main classes: numerical, symbolic and graphical. Mathematica handles these three classes in a unified way.

Mathematica uses symbolic expressions to provide a very general representation of mathematical and other structures. The generality of symbolic expressions allows Mathematica to cover a wide variety of applications with a fairly small number of methods from mathematics and computer science.

The simplest way to use Mathematica is like a calculator. You type in a calculation, and Mathematica prints back the answer. The range of calculations that you can do with Mathematica is however far greater than with a traditional electronic calculator, or, for that matter, with a traditional programming language such as Fortran or BASIC. Thus, for example, while a traditional system might support perhaps 30 mathematical operations, Mathematica has over 750 built in. In addition, while traditional systems handle only numerical computations, Mathematica also handles symbolic and graphical computations.

Here are some simple examples. Each one consists of a short “dialog” with Mathematica. The text on the lines labeled In[n]:= is what you type in; the lines labeled Out[n]= are what Mathematica prints back. The “Tour of Mathematica” on page 1 gives more examples.
Numerical Computation

Example: Find the numerical value of \( \log(4\pi) \).

\[
\text{Mathematica}\quad \text{version of } \log(4\pi). \quad \text{The } \texttt{N} \text{ tells } \text{Mathematica} \text{ that you want a numerical result.}
\]

Here is \( \log(4\pi) \) to 40 decimal places.

\[
\begin{align*}
\text{In}[2]:= & \text{N[ Log[ 4 Pi ] , 40 ]} \\
\text{Out}[2]= & \quad 2.5310242469692977891594269411847798
\end{align*}
\]

Whereas a traditional calculator or numerical computation system handles numbers only to fixed degree of precision, \texttt{Mathematica} can handle numbers of any precision. In addition, \texttt{Mathematica} includes a full range of higher mathematical functions, from elliptic integrals and complex Bessel functions to hypergeometric functions and integer factorization.

\texttt{Mathematica} can do numerical computations not only with individual numbers, but also with objects such as matrices. It supports linear algebra operations such as matrix inversion and eigensystem computation. \texttt{Mathematica} can handle numerical data, allowing you to do statistical and other analysis, as well as performing operations such as Fourier transforms, interpolation and least-square fitting.

\texttt{Mathematica} can do numerical operations on functions, such as numerical integration, numerical minimization, and linear programming. It can also generate numerical solutions to both algebraic equations and ordinary differential equations.

Symbolic Computation

Example: Find a formula for the integral \( \int x^4/(x^2 - 1) \, dx \).

Here is the expression \( x^4/(x^2 - 1) \) in \texttt{Mathematica}.

\[
\begin{align*}
\text{In}[1]:= & \quad x^4 / (x^2 - 1) \\
\text{Out}[1]= & \quad \frac{x^4}{x^2 - 1} \\
\text{In}[2]:= & \quad \text{Integrate}[x^4 / (x^2 - 1), x] \\
\text{Out}[2]= & \quad \frac{x}{3} + \frac{\log(-1 + x)}{2} - \frac{\log[1 + x]}{2}
\end{align*}
\]
One major class of calculations made possible by Mathematica’s symbolic computation capabilities is those involving the manipulation of algebraic formulas. Mathematica can do many kinds of algebraic operations. It can expand, factor and simplify polynomials and rational expressions. It can find algebraic solutions to polynomial equations and systems of equations.

Mathematica can also do calculus. It can evaluate derivatives and integrals symbolically and find symbolic solutions to ordinary differential equations. It can derive and manipulate power series approximations, and find limits. Standard Mathematica packages cover areas such as vector analysis and Laplace transforms.

### Graphics

**Example: Plot the function \( \sin(xy) \) for \( x \) and \( y \) between 0 and \( \pi \).**

This generates a three-dimensional plot of \( \sin(xy) \) as a function of \( x \) and \( y \).

In [1]:= Plot3D[ Sin[x y], {x, 0, Pi}, {y, 0, Pi} ]

There are many options for controlling graphics in Mathematica.

Mathematica produces both two- and three-dimensional graphics, as well as contour and density plots. You can plot both functions and lists of data. Mathematica provides many options for controlling the details of graphics output. In three dimensions, for example, you can control shading, color, lighting, surface shininess and other parameters. Many versions of Mathematica also support animated graphics.

Mathematica incorporates a graphics language, in which you can give symbolic representations of geometrical objects using primitives such as polygons, then render the objects graphically. All graphics produced by Mathematica are in standard PostScript, and can be transferred to a wide variety of other programs.
The *Mathematica* Language

In addition to having a large number of built-in functions, *Mathematica* also includes a full programming language, which allows you to add your own extensions to the system.

*Mathematica* is a high-level programming language, in which you can write programs, both large and small. The fact that *Mathematica* is an interactive system means that you can run your programs as soon as you have typed them in.

**Example: Define a function to generate a list of primes.**

This defines a function \( f \) which makes a table of the first \( n \) prime numbers.

\[
\text{In}[1] := f[n_] := \text{Table}[\text{Prime}[i], \{i, n\}]
\]

You can use the definition of \( f \) immediately. Here is a table of the first 10 prime numbers.

\[
\text{In}[2] := f[10]
\]

\[
\text{Out}[2] = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29\}
\]

*Mathematica* programs can make use of the symbolic aspects of *Mathematica*. They can create and manipulate arbitrary symbolic data structures. *Mathematica* programs themselves are also symbolic expressions, and can be combined and manipulated using standard *Mathematica* operations.

*Mathematica* supports several programming styles, including:

- **Procedural programming**, with block structure, conditionals, iteration and recursion.
- **Functional programming**, with pure functions, functional operators and program-structure operations.
- **Rule-based programming**, with pattern matching and object orientation.

Fundamental to much of *Mathematica* is the notion of transformation rules, which specify how symbolic expressions of one form should be transformed into expressions of another form. Transformations are a very general and natural way to represent many kinds of information, particularly mathematical relations.

Using transformation rules you can, for example, transcribe almost directly into *Mathematica* the kind of material that appears in tables of mathematical formulas.
Example: Define your own logarithm function in Mathematica.

<table>
<thead>
<tr>
<th>Mathematical form</th>
<th>Mathematica form</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \log(1) = 0 )</td>
<td>( \text{log}[1] = 0 )</td>
</tr>
<tr>
<td>( \log(e) = 1 )</td>
<td>( \text{log}[e] = 1 )</td>
</tr>
<tr>
<td>( \log(xy) = \log(x) + \log(y) )</td>
<td>( \text{log}[x,y] := \text{log}[x] + \text{log}[y] )</td>
</tr>
<tr>
<td>( \log(x^n) = n \log(x) )</td>
<td>( \text{log}[x^n] := n \text{log}[x] )</td>
</tr>
</tbody>
</table>
What Is Mathematica?

Mathematica Interfaces

Many Mathematica systems are divided into two parts: the kernel, which actually performs computations, and the front end, which handles interaction with the user. The kernel works the same on all computers that run Mathematica. The front end, on the other hand, is optimized for particular computers and graphical user interfaces.

On many computers, the front end for Mathematica supports sophisticated interactive documents called notebooks. These consist of text arranged in a hierarchical way, together with graphics that can be animated, and Mathematica expressions that can be used for actual Mathematica computations. With notebooks, you can create pedagogical and other material that both explains and performs computations.

Mathematica follows many software standards that allow it to exchange material with other programs. Thus, for example, Mathematica graphics are represented in PostScript, so that they can be exchanged with desktop publishing and other programs. In addition, Mathematica can read data in various formats, and can generate output for systems such as C, Fortran and \TeX.

Mathematica can communicate at a high level with other programs using the MathLink communication standard. Many kinds of programs can be adapted to be MathLink compatible, so that they can exchange data and commands with Mathematica. The standard Mathematica front end, for example, can use MathLink to communicate with the kernel. The communication can be done both within a single computer, and across a network between different computers.

With MathLink, you can use Mathematica to control external programs. You can prepare input and commands using the Mathematica language, then send these to an external program via MathLink. You can then get results back into Mathematica for analysis or display.

You can also use MathLink to create programs that call Mathematica as if it were a subroutine. In this way, you can set up your own complete front end or control system for Mathematica.