Reprint from
The *Mathematica* Conference

June, 1992   Boston, MA

867.21.10.1992

Chapter 1 **Introduction**

Mathematica is a computer system which integrates symbolic and numerical mathematics with powerful computer graphics. These are supported by a concise and flexible programming language. This first chapter consists of a brief description of *Mathematica* as a system and how to use it.

1.1 Introduction to *Mathematica*

Before we start to look at *Mathematica* graphics we should first review some basic *Mathematica* concepts.

Mathematica can be split into two parts: the kernel, which actually performs computations, and the front end, which handles interaction with the user. The kernel contains the same functionality on all the computers that run *Mathematica*. The front end, on the other hand, is optimized for particular computers and is concerned with system-specific tasks such as on-line editing, graphical display and printing. On many computers, the front end for *Mathematica* supports sophisticated interactive documents called Notebooks. These consist of an interactive mixture of computations, text and graphics. However the same input commands work on all versions to produce equivalent results.

When you start up a *Mathematica* kernel you see an input prompt `In[`*num*`]:=`, where *num* is a positive integer. Your documentation will tell you how to do this, the actual operations may vary from one system to another. With an input prompt you may enter something and ask *Mathematica* to evaluate it. This is how *Mathematica* works, you give it input, this is evaluated and the result is returned. We shall now examine a sample session. Enter the text which follows the input prompt.

`In[` *num* `]`	*Mathematica* is waiting for an input
`Out[` *num* `]`	the result from *Mathematica*

This arithmetic operation evaluates to a result different from the input expression.

```
In[1]:= 2 + 5

Out[1]= 7
```

There are many built-in mathematical functions.

```
In[2]:= Sin[ Sqrt[ 10.5]]

Out[2]= -0.0986171
```

The ? is a simple and quick way to get help on a function.

```
In[3]:= ?Sin

Sin[z] gives the sine of z.
```

The result is a rational number which can not simplify further.

```
In[3]:= 100 50/ 89
```

$$Out[3]= \frac{5000}{89}$$

`N` converts its argument to a real number. The `%` means the previous result.

```
In[4]:= N[ %]

Out[4]= 56.1798
```

You can also refer to a result further back by using the number from `Out`.

```
In[5]:= N[ %1]^2

Out[5]= 49.
```

Mathematica Expressions

We have now looked at various forms of arithmetic. The main type of input we have seen is for numbers. *Mathematica* can of course do much more.

- Numbers
- Symbols
- Strings
- Compound expressions

Some of the different types of *Mathematica* expressions.

Assign the number 42 to the symbol a.

```
In[6]:= a = 42
Out[6]= 42
```

The symbol `a` evaluates as it was set in the previous input.

```
In[7]:= a
Out[7]= 42
```

A string is put inside double quotes " ". Notice how the result drops these.

```
In[8]:= "This is a string"
Out[8]= This is a string
```

`InputForm` shows what we would have to enter into *Mathematica* to get the same result.

```
In[9]:= InputForm[ %]
Out[9]//InputForm= "This is a string"
```

This is a `List`. It is a fundamental way to hold a group of objects.

```
In[10]:= expr = {a,b,c,d}
Out[10]= {42, b, c, d}
```

The command `Table` is another way to generate a `List`. Here `i` is set to go from 1 to 5 in steps of 1.

```
In[11]:= Table[ {i, i^2, i^3}, {i,5}]
Out[11]= {{1, 1, 1}, {2, 4, 8}, {3, 9, 27}, {4, 16, 64},
  {5, 25, 125}}
```

`FullForm` shows how *Mathematica* actually stores *expr*.

```
In[12]:= FullForm[ expr]
Out[12]//FullForm= List[42, b, c, d]
```

The `Head` of *expr* is `List`. The `Head` of an expression is a way to label it.

```
In[13]:= Head[ expr]
Out[13]= List
```

The number of arguments of an expression is given by the `Length` function.

```
In[14]:= Length[ expr]
Out[14]= 4
```

`Part` returns that component of an expression.

```
In[15]:= Part[ expr, 3]
Out[15]= c
```

This is identical to the previous input. This is an abbreviated way to refer to `Part`. A number of *Mathematica* commands have similar abbreviations.

```
In[16]:= expr[[3]]
Out[16]= c
```

Mathematica Options

Many *Mathematica* commands can be altered by the use of options. All of the graphics commands have a number of options which alter the way they behave. To see the options which are available for a *Mathematica* command one can use `Options`.

This shows the options for `Plot`.

```
In[17]:= Options[ Plot]

Out[17]= {AspectRatio -> 1/GoldenRatio, Axes -> Automatic,
  AxesLabel -> None, AxesOrigin -> Automatic,
  AxesStyle -> Automatic, Background -> Automatic,
  ColorOutput -> Automatic, Compiled -> True,
  DefaultColor -> Automatic, Epilog -> {}, Frame -> False,
  FrameLabel -> None, FrameStyle -> Automatic,
  FrameTicks -> Automatic, GridLines -> None, MaxBend -> 10.,
  PlotDivision -> 20., PlotLabel -> None, PlotPoints -> 25,
  PlotRange -> Automatic, PlotRegion -> Automatic,
  PlotStyle -> Automatic, Prolog -> {}, RotateLabel -> True,
  Ticks -> Automatic, DefaultFont :> $DefaultFont,
  DisplayFunction :> $DisplayFunction}
```

If there are no options an empty list is returned.

```
In[18]:= Options[ Integrate]

Out[18]= {}
```

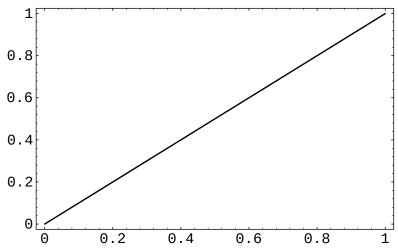
This shows the default setting for an option.

```
In[19]:= Options[ Plot, Frame]

Out[19]= {Frame -> False}
```

This changes the option `Frame` for `Plot`. The actual meaning of these options is discussed later.

```
In[20]:= Plot[ x, {x,0,1}, Frame -> True]
```



This changes the default value for an option. The semicolon at the end of the input line suppresses the printing of the result.

```
In[21]:= SetOptions[ Plot, Frame -> True];
```

The default setting has now changed.

```
In[22]:= Options[ Plot, Frame]

Out[22]= {Frame -> True}
```

We should change the default back.

```
In[23]:= SetOptions[ Plot, Frame -> False];
```

Miscellaneous Useful Features

Mathematica runs on many different types of computers. These computers have a number of differences in the way in which they refer to and manipulate files. *Mathematica* provides commands to work with files in a way that is the same for every computer on which it runs.

This shows the files which end with the letters `dat`.

```
In[24]:= FileNames[ "*dat"]

Out[24]= {file.dat, file1.dat}
```

This shows the contents of the file `file.dat`. If this does not work probably the file does not exist on your system. Use a text editor to make it and insert these numbers. Then try again.

```
In[25]:= !!file.dat

1   1
2   8
3   27
4   64
5   125
6   216
7   343
8   512
```

This will reenter input line 19.

```
In[25]:= In[19]

Out[25]= {Frame -> False}
```

This retrieves output 19.

```
In[26]:= %19

Out[26]= {Frame -> False}
```

This is an equivalent way to retrieve output.

```
In[27]:= Out[19]

Out[27]= {Frame -> False}
```

1.2 More Information on *Mathematica*

If you wish to learn more about using *Mathematica* you should read one of the many publications and tutorials to the system which are available. Wolfram Research maintains a list of such publications which is available on request. In addition Wolfram Research publishes a number of documents such as collections of notes from *Mathematica* conferences. These cover a wide variety of topics from the general introductory to the most complex aspects of the system.

The standard reference is the book *Mathematica: A System for Doing Mathematics by Computer*, Second Edition, by Stephen Wolfram, which I shall refer to as The *Mathematica* Book. It is distributed with every copy of the software and is also available in many technical bookstores. The *Mathematica* Book contains a reference guide which lists in alphabetical order every symbol defined by the system such as commands and options. In the reference guide each of these symbols is described in some detail. Additionally the body of the book gives further information and examples of use.