Chapter 2: Mathematica

Mathematica - “A system for doing mathematics on a computer.”

An interpretive system with capabilities for

- Numerical Calculations
- Symbolic Manipulations
- Graphics
- List Processing
- Programming
- Special Packages
- Conversions to C, FORTRAN, TEX

etc.

Examples:

1. Numerical Calculation:

```
In [1]:=
    5+4  {user input  - bold Courier font
              (all characters have same width)
Out [1]:=
    9    {Mathematica output - plain Courier
              font
```

Mathematica adds the lines:

```
In [1]:=

Out [1]:=
```

We omit these lines in the following examples:

2. Symbolic Manipulation

```
Expand[(a+b)^2]
```

$a^2 + 2\ a\ b + b^2$

```
Integrate[Sin[x], x]
-Cos[x]
```
{Indefinite Integral

```
Integrate[Sin[x], (x, a, b)]
Cos[a] - Cos[b]
```
{definite Integral

```
D[Sin[x], x]
```
{differentiation

```
Cos[x]
```

3. Graphics:

```
Plot[Sin[x], (x, 0, 2Pi)]
```

```
- Graphics -
```

```
Plot3D[Sin[x] Sin[y], (x,0,2Pi), (y,0,2Pi)]
```

```
                    - SurfaceGraphics -
```

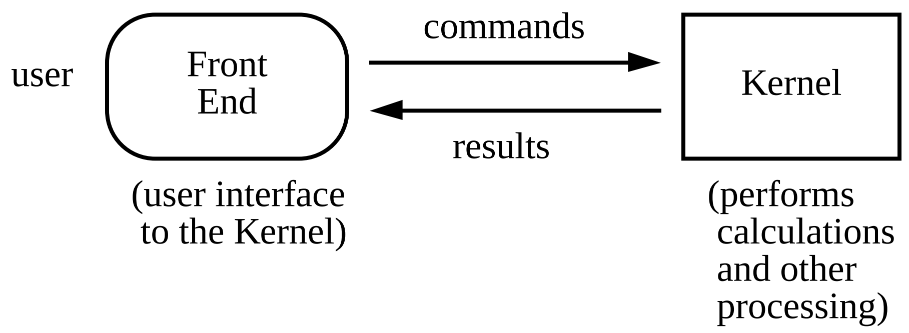
4. List Processing

```
(10, 15, 20, 25) + 5
(15, 20, 25, 30)

Sort[(18, -2, 7, 0, -1)]
(-2, -1, 0, 7. 18)
```

5. Programming

```
sum = 0.0;
Do[sum=sum+1.0/k, (k,10)]
Print[“sum = “,sum]

sum = 2.92897
```

<u>Components of Mathematics</u>


user
Front End
commands
results
Kernel
(user interface to the Kernel)
(performs calculations and other processing)

On PC machines, have Notebook Front End, which is similar to a word processor.

Notebook - any document created by front end.

Any expression you type is sent to the Kernel when you depress the

- <u>Enter</u> Key

or when you depress

- <u>Shift-Return</u>

Cell - basic organizational unit of a Notebook. Can contain text, Mathematical input or output, or other cells.

Extent and attributes of cells are marked by <u>cell brackets</u> in the right margin:

(markings on the brackets indicate cell type; input, output, etc.)

Basic Mathematica Syntax

- Variable Names - reference to memory locations storing data values.

  Can be any combination of letters, digits, $; but must start with a letter or $. Should start with lower case letter to avoid confusion with built-in functions.

  e.g., x, sum force2 (good programming practice: use descriptive names)

  Also can use as symbolic names in expressions, with no numeric values assigned to the name; e.g., (1+x)(1-x).

- Arithmetic Operators:

  +, -, *, /, ^

  Can also specify multiplication with a space:

  e.g., 5 12 is same as 5*12

- Assignment Statements - assign expressions or values to variable name using equal sign (=).

  e.g., sum = 0, force = m a

- Parentheses () - used for grouping.

  e.g., (a^2 + b)/(2c) (Note: 2c is same as 2*c)

- Square Brackets [] - used for specifying arguments of functions

  e.g., f[x] - note: <u>cannot</u> use parentheses.

- Curley Brackets (Brackets) {} - used to denote lists.
  e.g., point = {x, y, z}

- Double Square Brackets [[ ]] - used for indexing:

  i.e., specifying items in a list.

  List index positions are numbered 1, 2, 2, . . .

  For the list above, point[[3]] denotes the third item: z.

- Percent Sign % - references previous output.%- references last output.

  %% - references second to last output.

  %%%- references third to last output, etc.

  %n - references output number n.

Example:

```
3^4
81

5 %
405
```

- Suppressing Output - use semicolon at end of statement

```
x = 5^3;
```

(value of x not listed)

```
%
125
```

Built-In Mathematica Names - functions, special symbols, options, constants.

Start with a capital letter.

Usually complete words, unless abbreviation is commonly known.

<u>Examples</u>:

Pi	(3.1415926...)
E	(base of natural logs: e = 2.718281828...)
I	($\sqrt{-1}$)
Sin	(arguments of trig functions in radians)
Cos	
Tan	
Abs	
Round	(closed integer to argument)
Floor	(largest integer ≤ argument)
Ceiling	(smallest integer ≥ argument)
Sqrt	
Exp	(raise e to a power)
Log	(log to specified base)
Plot, Plot3D	
Expand	
Factor	
FindRoot	(each word in name capitalized)
Convert	

- and many, many more.

Examples:

x	Round[x]	Floor[x]	Ceiling[x]
1.7	2	1	2
1.5	1	1	2
1.3	1	1	2
-1.3	-1	-2	-1
-1.5	-1	-2	-1
-1.7	-2	-2	-1

Log[x] $\rightarrow \log_e x$ (ln x)

Log[b, x] $\rightarrow \log_b x$

Help:

Info on any built-in name can be obtained by typing
? followed by the name. For example,

```
?Plot
Plot[f, {x, xmin, xmax}) generates a plot of f as a
function of x from xmin to xmax. Plot[{f1, f2,...}, {x,
xmin, xmax}] plots several functions fi.
```

Numerical Calculations in Mathematica

Type in expression, and press Enter Key.

```
132 45
5940
```

Result of a computation is represented as accurately as possible. E.g.,

```
Sqrt[12]
2 Sqrt[3]
```

```
(2.7 + Pi)/4.1
0.243902 (2.7 + Pi)
```

To obtain a numerical approximation to previous calculation, we use the function N.

```
N(%)
1.42478
```

Similarly,

```
N[Sqrt[12]]
3.4641
```

To obtain more than 6 digits, add precision field:

```
N[%,20]
3.4641016151377545871
```

Rational Number (ratio of two integers) Calculations - return a rational or an integer

```
1/2 - 3/5

  1
-(--)
  10
```

To obtain a decimal answer (which may be an approximation), change one or more digits to floating point:

```
1.0/2 - 3/5   {“mixed mode” - in general, this is bad
-0.1 .         programming practice
```

We can change a decimal number to a rational number with Rationalize function. E.g.,

```
Rationalize[-0.1]

  1
-(--)
  10
```

To find out how many decimal digits were computed, use Precision function. E.g.,

```
N[Sqrt]]

3.4641

Precision[%]

19            {default precision for Mac
```

For exact calculations, precision is “infinite”.

To find out how long it takes Mathematica to perform a calculation, use
Timing function. E.g.,

```
Timing[N[Sqrt[12]];]

{0.0333333 Second, Null}
```

Omit semicolor to get result of calculation. E.g.,

```
Timing[N[Sqrt[500000]]]

{0.05 Second, 707.107}
```

<u>Output Formats</u>

- can output numeric values in “scientific” or “engineering” form.

<u>Example</u>:

```
N[Pi^10,8]

93648.047
```

`ScientificForm[%,5]` {one nonzero digit
to left of decimal point

9.3648 10^4

`EngineeringForm[%,5]` {exponents divisible by 3

93.648 10^3

(Also have “accounting” form with numbers in standard decimal notation using parentheses for negative numbers.)

We can use variable names in assignment statements to save the results of numerical calculations values. E.g.,

```
x = Sqrt[12];
```

Future references to x (in any one session) invoke this value:

```
y = 5x + 1

1 + 10 Sqrt[3]
```

The most common mistake in Mathematica is forgetting about previously assigned values (numeric or expression) to a variable name.

We can avoid such mistakes by “clearing” variable names when we no longer need them -- using either of the following:

```
x =.
Clear[x]
```

Now variable x is treated as a symbolic name with no assigned values.

Similarly,

```
Clear[x, y]          {clears both x and y
```

We can also completely remove a variable from the system with the function:

Remove[x]

This function is particularly useful when using various packages that might involve conflicts between variables that have the same name.

Mathematica Packages

In addition to "built-in" functions, Mathematica provides "packages" that contain collections of related functions in specialized areas.

For example, the package **`PhysicalConstants contains the functions: SpeedOfLight, ElectronMass,`** and many others.

These standard packages are stored according to categories ("directories"), such as

`Algebra`	**`LinearAlgebra`**
`Calculus`	**`Miscellaneous`**
`DiscreteMath`	**`Statistics`**
`Geometry`	**`Utilities`**
`Graphics`	etc.

For example,

- directory **`Miscellaneous`** contains the packages**: `Calendar, ChemicalElements, PhyscialConstants, Units,`** and others.

- directory **Graphics** contains the packages: **`Animation, Graphics, Graphics3D, Shapes, Splines,`** etc.

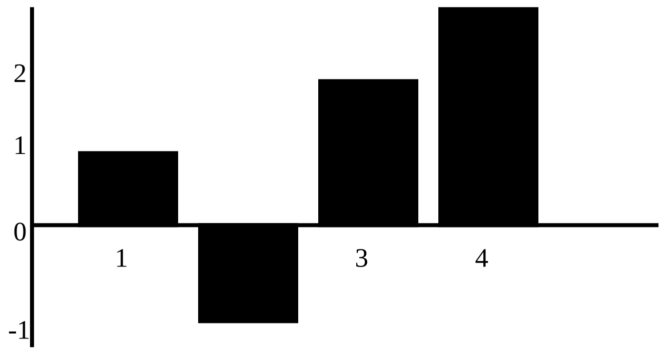
User-defined and "imported" directories and files (programs, etc.) can also be created.

To use the functions in a particular package, we must first load the package with the command:

```
<<dirName‘pkgName’
```

Example:

`<<Graphics‘Graphics`’ {2D graphics functions}
`BarChart[{0.8, -1.1, 1.5, 2.8}]`



If we need to use several functions from the different packages within a single directory, we can load all the packages at once using the package name “**`Master”`**. E.g.,

`<< Utilities‘Master’`

Packages can also be loaded with the function:

```
Needs[”dirName‘pkgName’”]
```

which does not reload the package (as << does) if it has already been loaded.

Random Number Generation

Most languages provide a random number generator (called Random, Rand, Rnd, Ranf, etc.).

Numbers returned by this routine are uniformly distributed over the interval (0,1). These numbers are referred to as **`pseudorandom numbers`**, since each number is calculated from the preceding number (and so sequence is predictable, given the first number). Otherwise, the numbers are statistically equivalent to true random numbers.

Random number generators are based on the calculations:

$$i, = ci_{k-1} + d \,(\mathrm{mod}\ m) \qquad k = 1, 2, 3, \ldots$$

$$r_i = \frac{i_k}{m} \qquad \text{normalized to interval (0,1)}$$

where c, d, m, i_k are integers, and i_0 is called the seed.

Parameter m is made as large as possible (e.g., 2^{31} -1 on machines with 32 bit word).

Constants c, d are chosen to minimize repeat values of *i*, i.e., we want large “period” for sequence of random numbers.

Argument for a random number function can be

- ignored
- a seed
- a range specification

Random Number Functions in Mathematica

Random[] - returns value in range (0,1)

Random[Real, rmax] - real value between 0 and rmax

Random[Real, {rmin, rmax}] - real in range (rmin, rmax)

Random[Integer, {intmin, intmax}] - random integer in range
(intmin, intmax)
Random[Integer] - 0 or 1, each with probability 0.5

Random[Complex] - returns random complex number in unit square.

Random[Complex, {zmin, zmax}] - complex number within rectangle
specified by zmin and zmax in complex plane (i.e., specified
in form `a + I b`)
Random[type, range, n] - random number of specified type and range
with n digits of precision

SeedRandom[] - reset seed as time of day
SeedRandom[seed] - r**eset** seed to specified integer value

Example:

```
SeedRandom[100]
r1 = Random[]
r2 = Random[]

0.216623
0.656427

SeedRandom[100]    {Useful for testing an algorithm
x = Random[]          with same random sequence

0.216623
```

Symbolic Manipulations in Mathematica

Expand[expr] - produces a sum of expanded products and powers

Factor[expr] - rearranges expr into a minimal product of factors

Simplify[expr] - attempts to rearrange expr into a form having the smallest number of parts

Collect[expr, x] - writes expr as a sum of powers of x

Collect[expr, {x1, x2, . . .}] - collects powers of x1, x2, etc.

(plus other functions)

Examples:

```
y = (1 - x)(1 + x);        {assuming no numerical
yexp = Expand[y]                assignment to x
```

$1 - x^2$

```
yfact = Factor[yexp]
```

```
(1 - x) (1 + x)
```

```
yfactsimp = Simplify[yfact]
```

$1 - x^2$

Can perform similar manipulations on functions of more than one variable.
E.g.,

```
x = (1 - x)^2 (1 + y)^2
%exp = Expand[z]
```

$$1 - 2x + x^2 + 2\,y - 4\,x\,y + 2\,x^2\,y + y^2 - 2\,x\,y^2 + x^2\,y^2$$

```
%expCollxpwr = Collect[%exp, x]
```

$$1 + 2\,y + y^2 + x\,(-2 - 4\,y - 2\,y^2) + x^2\,(1 + 2\,y + y^2)$$

and we obtain a polynomial in x with coefficients that are polynomials in y.

What would we obtain with the following functions?

```
Collect[%exp,y]
Collect[%exp, {x,y}]
Collect[%exp, {y,z}]
```