Differential Equations

Wolfram can solve ordinary, partial and delay differential equations. DSolveValue takes a differential equations and returns the general solution.

In[∘]:=
```
sol = DSolveValue[y'[x] + y[x] == x, y[x], x]
```

Out[∘]= $-1 + x + e^{-x} c_1$

In[∘]:=
```
DSolveValue[{y'[x] + y[x] == x, y[0] == -1}, y[x], x]
```

Out[∘]= $-1 + x$

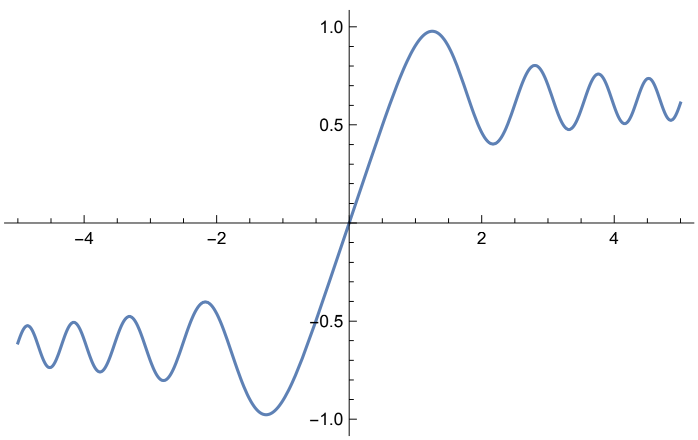
NDSolveValue finds the numerical solutions:

In[∘]:=
```
NDSolveValue[{y'[x] == Cos[x^2], y[0] == 0}, y[x], {x, -5, 5}]
```

Out[∘]= InterpolatingFunction[ Domain: {{-5., 5.}} Output: scalar][x]

You can plot this interpolating function directly

In[∘]:=
```
Plot[%, {x, -5, 5}]
```

Out[∘]=


Complex Analysis

In[∘]:=
```
I^2
```

Out[∘]= -1

In[∘]:=
```
(1 + I) (2 - 3 I)
```

Out[∘]= 5 - i

You can expand complex expressions

In[•]:=
```
ComplexExpand[Sin[x + I y]]
```
Out[•]=
```
Cosh[y] Sin[x] + ⅈ Cos[x] Sinh[y]
```

You can convert expresions between exponential and trigonometric form:

In[•]:=
```
ExpToTrig[E^(I x)]
```
Out[•]=
```
Cosh[I x] + Sinh[I x]
```

In[•]:=
```
TrigToExp[%]
```
Out[•]= e^{Ix}

To find out the conjugate value:

In[•]:=
```
(3 + 2 I)*
```
Out[•]=
```
3 - 2 ⅈ
```

To extract the real and imaginary parts of an expression:

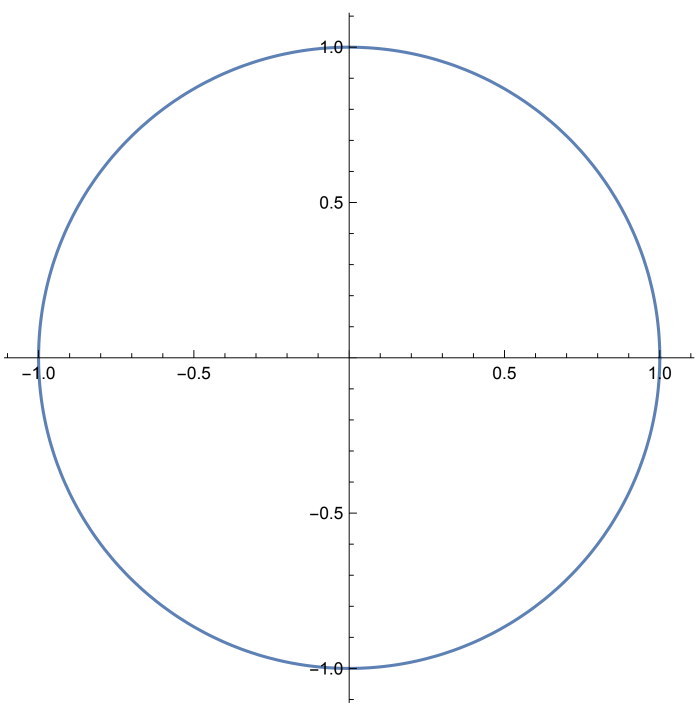
In[•]:=
```
ReIm[3 + 2 I]
```
Out[•]=
```
{3, 2}
```

To find the absolute value and argument:

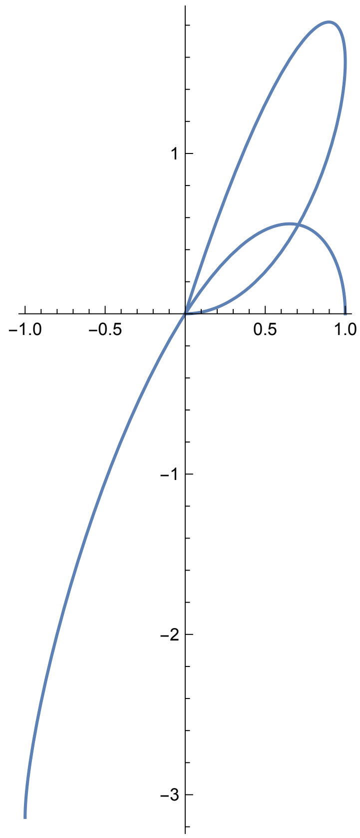
In[•]:=
```
AbsArg[(1 + I)]
```
Out[•]= $\left\{\sqrt{2}, \frac{\pi}{4}\right\}$

You can make a conformal mapping with Parametric Plot

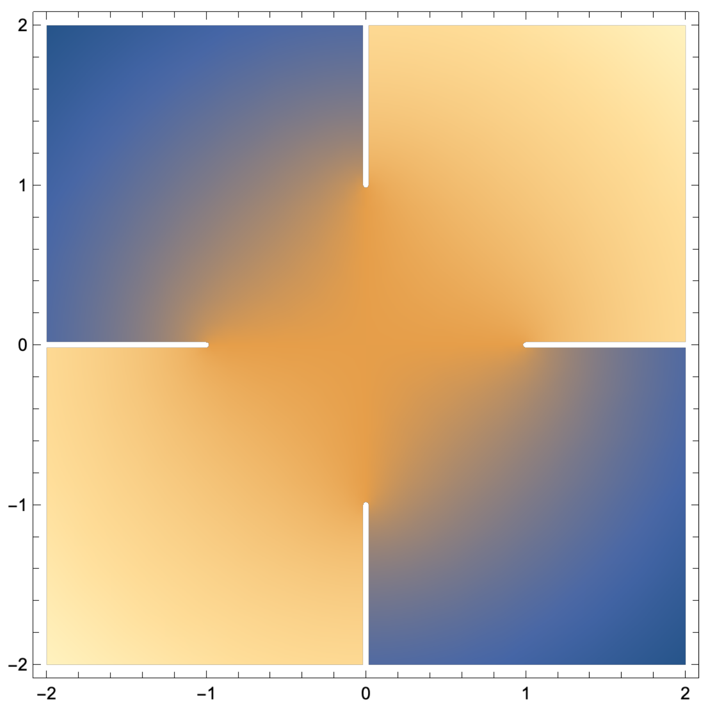
In[•]:=
```
ParametricPlot[ReIm[E^(I ω)], {ω, 0, 2 π}]
```
Out[•]=



In[•]:=
```
PolarPlot[AbsArg[E^(I ω)], {ω, 0, π}]
```

Out[•]=


In[•]:=
```
DensityPlot[Im[ArcSin[(x + I y)^2]], {x, -2, 2}, {y, -2, 2}]
```

Out[•]=


Matrix and Linear Algebra

Matrices as lists of lists.

In[∘]:= `{{1, 2}, {3, 4}}`

Out[∘]= `{{1, 2}, {3, 4}}`

In[∘]:= `MatrixForm[{{a, b}, {c, d}}]`

Out[∘]//MatrixForm=

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

You can construct a matrix with iterative functions:

In[14]:= `Table[x + y, {x, 1, 2}, {y, 0, 2}]`

Out[14]= `{{1, 2, 3}, {2, 3, 4}}`

In the above example, x values are 1,2,3 and y values are 0,1,2. For x = 1, y takes 0,1 and 2 values and x+y gives {1,2,3} and for x = 2, x+y gives {2,3,4} and so on.

In[17]:= `{1, 2, 3} {a, b, c}`

Out[17]= `{a, 2 b, 3 c}`

Compute the dot product of matrices:

In[18]:= `{{1, 2}, {3, 4}}.{{a, b}, {c, d}}`

Out[18]= `{{a + 2 c, b + 2 d}, {3 a + 4 c, 3 b + 4 d}}`

To find the determinant

In[19]:= `Det[{{a, b}, {c, d}}]`

Out[19]= `-b c + a d`

In[20]:= `Inverse[{{1, 1}, {0, 1}}]`

Out[20]= `{{1, -1}, {0, 1}}`

In[21]:= `LinearSolve[{{1, 1}, {0, 1}}, {x, y}]`

Out[21]= `{x - y, y}`

Discrete Mathematics

In[23]:= `FactorInteger[30]`

Out[23]= `{{2, 1}, {3, 1}, {5, 1}}`

In[24]:= `GCD[24, 60]`

Out[24]= `12`

Display the 4th prime number

In[25]:=
```
Prime[4]
```
Out[25]= 7

Use the Mod function for the remainder

In[26]:=
```
Mod[17, 5]
```
Out[26]= 2

Get all possible permutations of a list

In[27]:=
```
Permutations[{a, b, c}]
```
Out[27]= {{a, b, c}, {a, c, b}, {b, a, c}, {b, c, a}, {c, a, b}, {c, b, a}}

In[28]:=
```
5!
```
Out[28]= 120

In[29]:=
```
# For Combination Use Binomial
Binomial[4, 3]
```
Out[29]= Binomial Combination For Use #1

Out[30]= 4

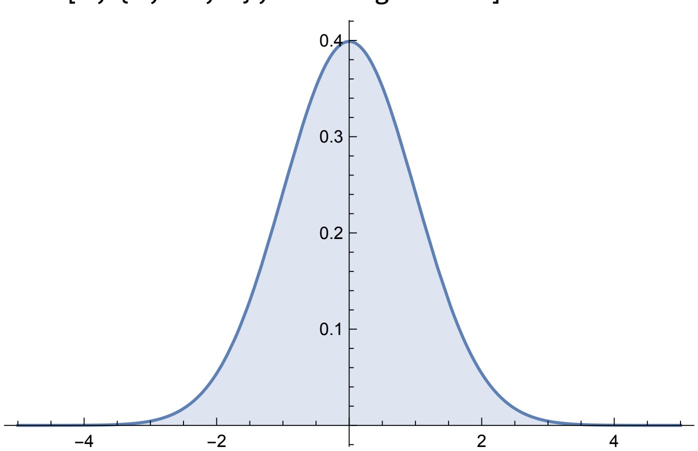
Calculate the expectation of a polynomial expression:

In[31]:=
```
Expectation[2 x + 3, x ≈ NormalDistribution[]]
```
Out[31]= 3

In[32]:=
```
PDF[NormalDistribution[0, 1], x]
```
Out[32]= $\frac{e^{-\frac{x^2}{2}}}{\sqrt{2\pi}}$

Above, PDF is used for a normal distribution in symbolic form. To make a plot of this distribution

In[33]:=
```
Plot[%, {x, -5, 5}, Filling → Axis]
```
Out[33]=



Statistics

In[34]:=
```
Mean[{1, 2, 4, 5}]
```
Out[34]= 3

In[38]:=
```
Correlation[{1, 3, 5}, {6, 4, 2}]
```
Out[38]= -1

In[42]:=
```
StandardDeviation[PoissonDistribution[2]]
```
Out[42]= $\sqrt{2}$

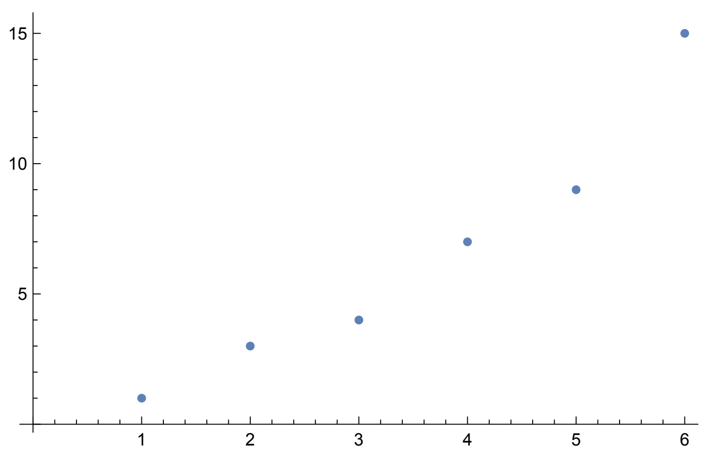
In[45]:=
```
Moment[{x, y, z}, 2]
```
Out[45]= $\frac{1}{3}\left(x^2+y^2+z^2\right)$

In[46]:=
```
MomentGeneratingFunction[NormalDistribution[μ, σ], t]
```
Out[46]= $e^{t\,\mu+\frac{t^2\,\sigma^2}{2}}$

Use // Short to get the short summary of the output.

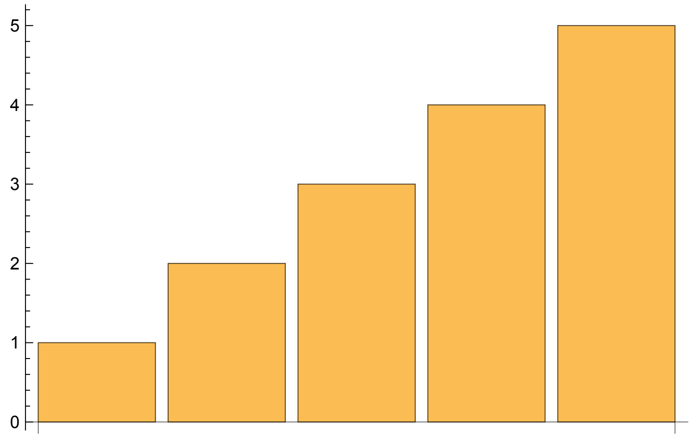
Data Plots and Best-Fit Curves

In[48]:=
```
ListPlot[{1, 3, 4, 7, 9, 15}]
```
Out[48]=



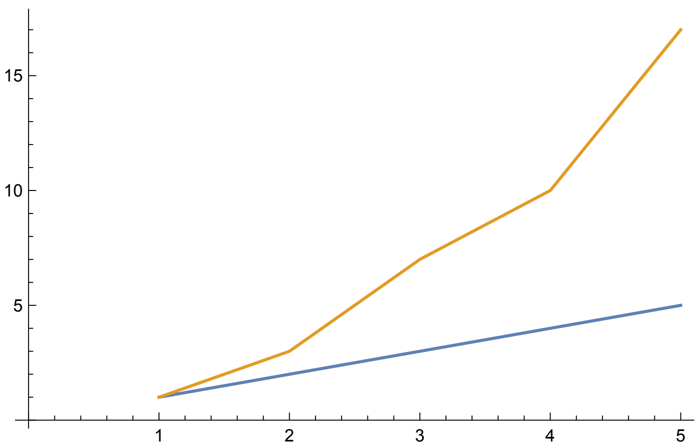
In[49]:=
```
BarChart[{1, 2, 3, 4, 5}]
```

Out[49]=



In[50]:=
```
ListLinePlot[{{1, 2, 3, 4, 5}, {1, 3, 7, 10, 17}}]
```

Out[50]=



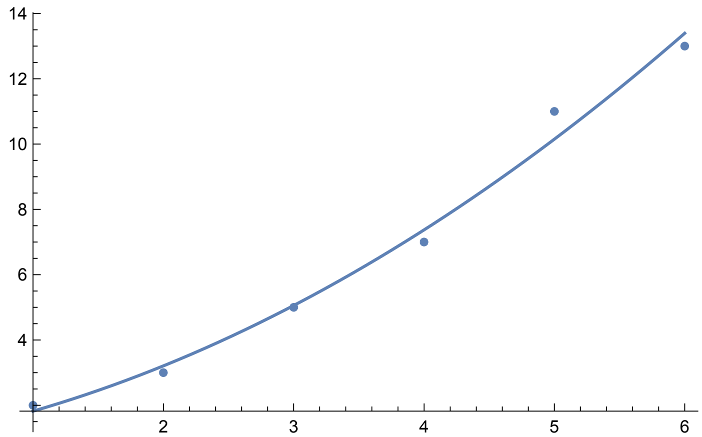
Find a curve of best fit with the Fit command:

In[51]:=
```
Fit[{2, 3, 5, 7, 11, 13}, {1, x, x^2}, x]
```

Out[51]= $0.9 + 0.689286\,x + 0.232143\,x^2$

Use Show to compare the curve with its data points:

In[52]:=
```
Show[{Plot[%, {x, 1, 6}], ListPlot[{2, 3, 5, 7, 11, 13}]}]
```

Out[52]=


Some functions:

In[53]:= $\sum_{i=1}^{n} i^2$

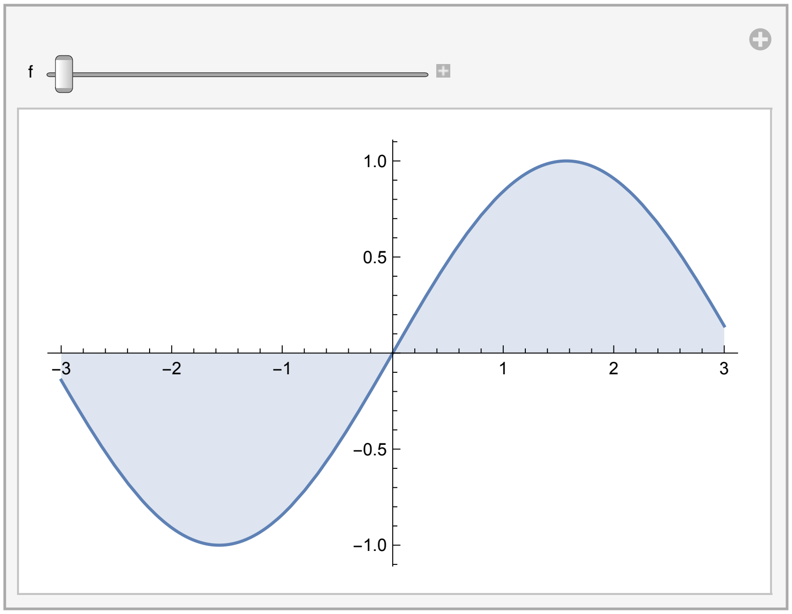
Out[53]= $\frac{1}{6} n (1 + n) (1 + 2 n)$

Interactive Models

The Manipulate command lets you interactively explore what happens when you vary parameters in real time:

In[57]:=
```
Manipulate[Plot[Sin[f t], {t, -3, 3}, Filling → Axis], {f, 1, 5}]
```

Out[57]=



A single Manipulate command can have multiple controllers, and the Wolfram Language automatically chooses the best layout of those controllers for you:

In[59]:=
```
Manipulate[Plot[Sin[f * t + p], {t, -3, 3}, Filling → fill, PlotStyle → color],
 {f, 1, 5}, {p, 3, 9}, {fill, {Bottom, Top, Axis}}, {color, Red}]
```

Out[59]=