Mathematica in Research.

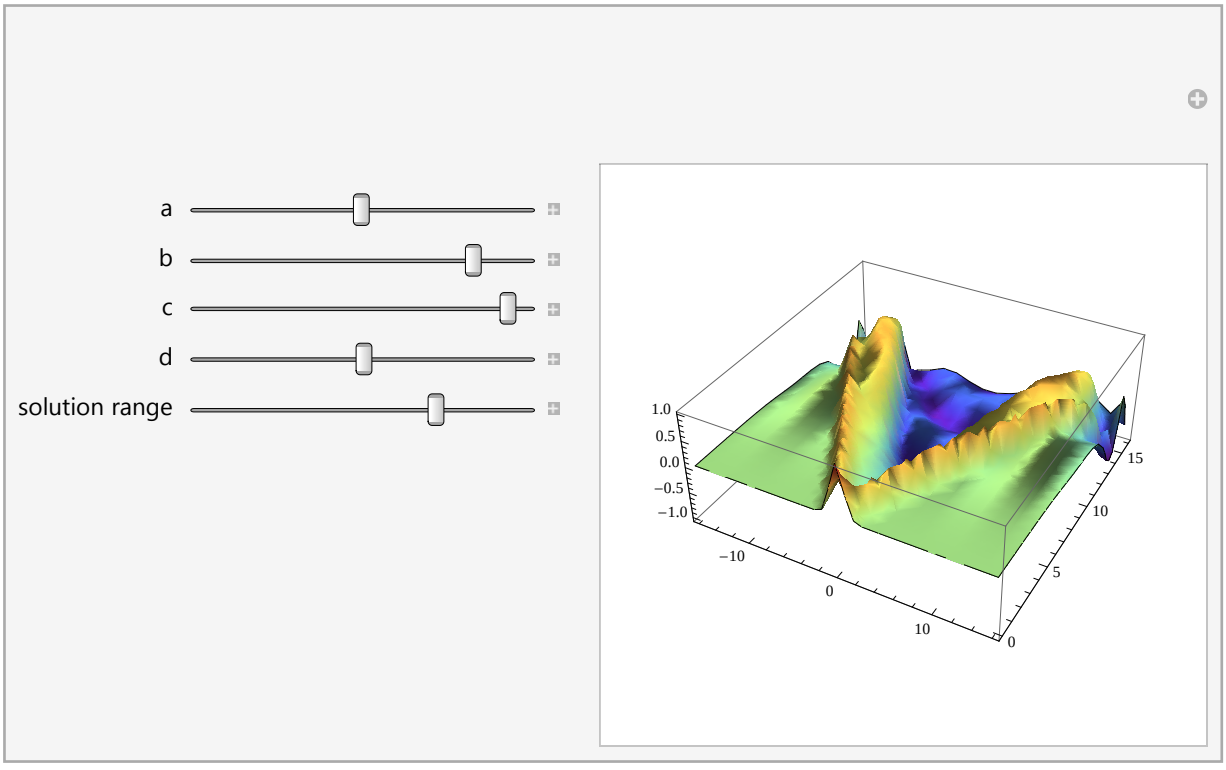
Wolfram *Seminar.*

Presenter: *Paritosh Mokhasi.*
paritoshm@wolfram.com,
Jeff Todd: *jtodd@wolfram.com,*
Justin Kehinde: *jkehinde@wolfram.com*

Mon 6 Jun 2011 12:11:01 (GMT –5.)

Mathematica presentation - Argonne National Laboratory.

What is *Mathematica?*



▲ Solve and dynamically visualize solution space of a nonlinear PDE

» **Compute & visualize**: wide range of functions for mathematics, science, engineering; automatic algorithm selection; visualize anything!

» **Develop:** consistent high-level language; choose the programming paradigm that is best for your computational needs – procedural, functional, rule-based, recursive, object-oriented; advanced development tools

» **Performance**: state-of-the-art performance, parallel computation with built-in multicore support, adaptive scalability, precision control

» **Data manipulation**: integrate computable data; statistical model analysis; charting and visualization tools for all your data

» **Dynamic interactivity**: automated interface construction; create dynamic interfaces quickly and easily; manipulate graphics, math, tables, text, etc.

» **Connectivity**: automated import/export; built-in connectivity with databases, spreadsheets, legacy C or Fortran code, Web Services, Java, .NET; web-enabled connectivity

» **Document & deploy**: interactive document system; write papers/reports/articles/books/homework; output to PDF, HTML, XML; create slide show presentations; dynamic report generation; deploy with *Mathematica* Player

What's New

Wolfram|Alpha Integration

Core Algorithms

2D and 3D Graphics

Probability and Statistics

Financial Engineering

Image Processing

Control Systems

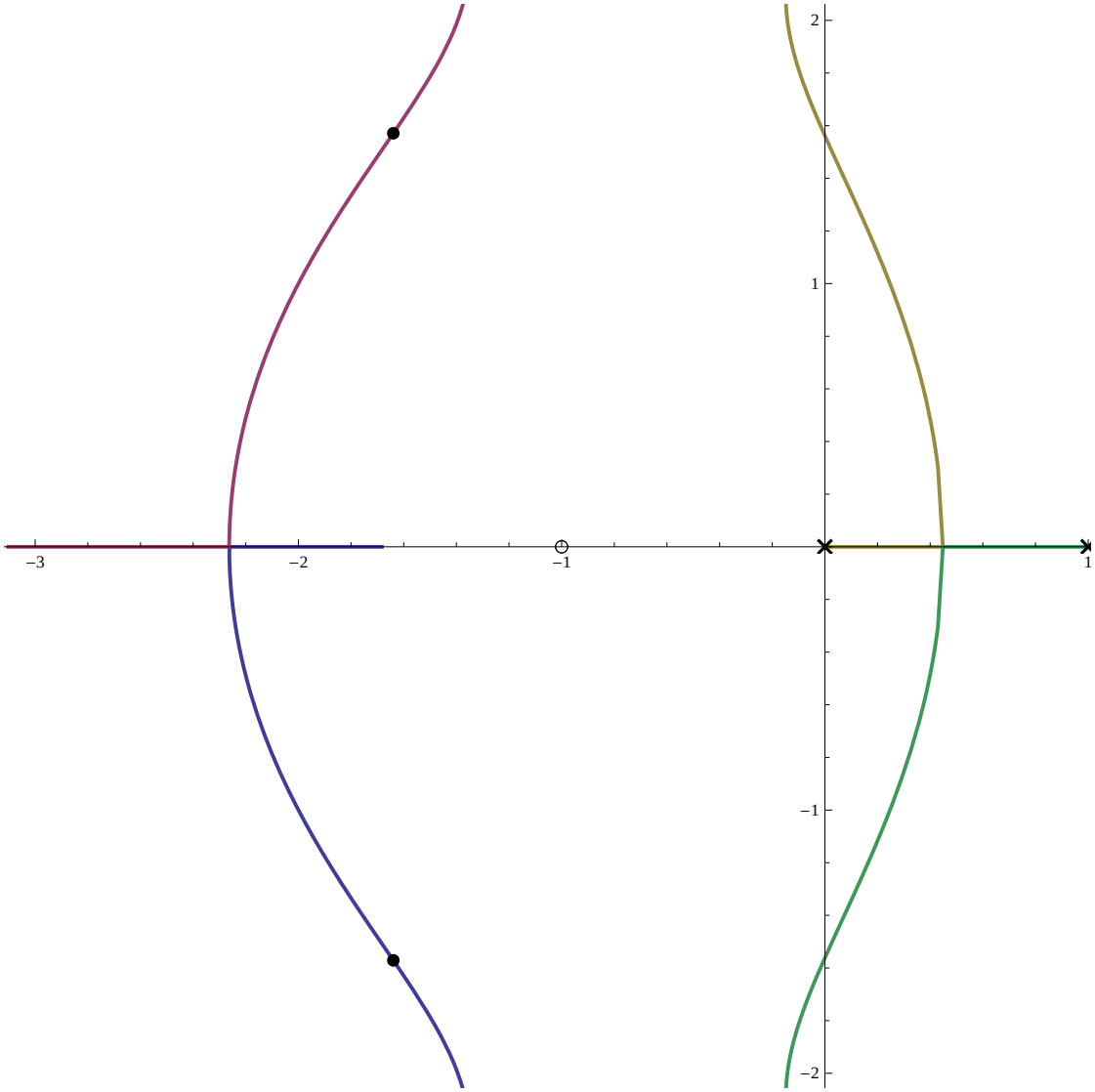
Wavelet Analysis

Graphs and Networks

Programming and Development

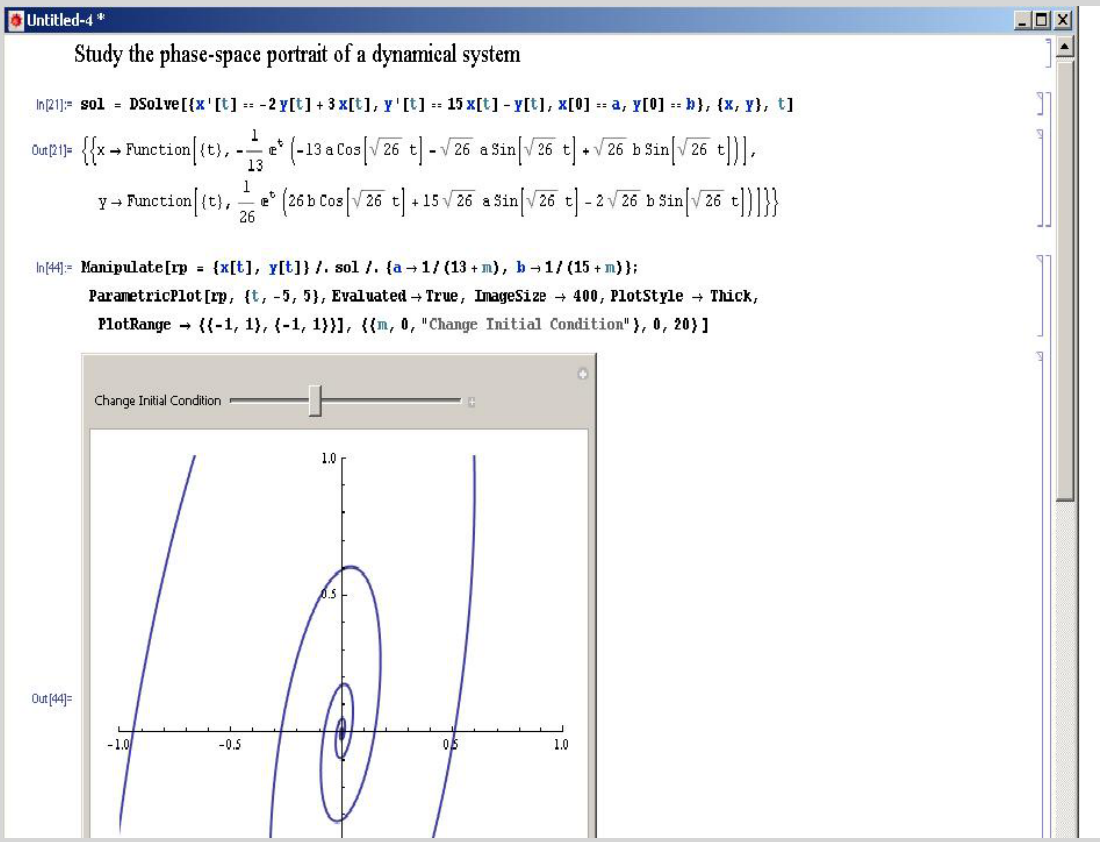
Mathematica: The Numerical Engine

ıs: Locus Plotting



Mathematica: The Symbolic Engine

nit | Series | Derivative | Integrate | Solve | Minimize | Differential Equations



Linear Algebra

Solve a small 3×3 linear system.

```
mat = {{2.0, 3.1, -1.1}, {0.2, -0.29, 3.3}, {-1.5, 4.2, 4.7}}
```

$$\text{mat} = \begin{pmatrix} 2.0 & 3.1 & -1.1 \\ 0.2 & -0.29 & 3.3 \\ -1.5 & 4.2 & 4.7 \end{pmatrix}$$

```
b = {-1.0, 2.2, 0.5};
{{2., 3.1, -1.1}, {0.2, -0.29, 3.3}, {-1.5, 4.2, 4.7}}
xvec = LinearSolve[mat, b]
```

Check the answer.

```
mat.xvec
```

Set up and solve a sparse system (`RandomSparseArray` is defined at the end of this notebook).

```
A = RandomSparseArray[{100 000, 100 000}];
b = RandomReal[{0, 1}, 100 000];
```

`LinearSolve` automatically chooses the appropriate algorithm to use on this sparse system.

```
xvec = LinearSolve[A, b]; // Timing
Norm[A.xvec - b]
Clear[A, xvec, b];
```

» Fast exact linear algebra

» Applications example - Solving elliptic PDE using finite differences.

Regression

Import the first sheet from a spreadsheet file and plot the data.

```
data = Import["noisydata.xls", {"Data", 1}]
dataplot = ListPlot[data]
```

Fit the data to a linear model.

```
fit = LinearModelFit[data, {1, x, x^2, x^3}, x]
```

Display the ANOVA table:

```
fit["ANOVATable"]
```

Show the fit function together with the data.

```
Show[dataplot,
 Plot[fit[x], {x, -5, 15}, PlotStyle → Red]]
```

Pulsar example

Solving Algebraic Equations

This solves a simple quadratic equation with symbolic parameters.

```
Solve[a x^2 + b x + c == 0, x]
```

Solve equations involving power series

```
y = 1 + Sum[a[i] x^i, {i, 3}] + O[x]^4;
eqn = D[y, x]^2 - y == x;
Solve[LogicalExpand[eqn]]
```

Whereas `Solve` is set up to give you generic solutions to equations, `Reduce` will give all possible solutions. This example restricts the solutions to the domain of real numbers.

```
Reduce[ Sqrt[27^(2 x-1), x] == Sqrt[9^(2 x-1)], x, Reals]
```

» Conditional results

Solve an equation over the reals and get a conditional result:

```
soln = Solve[x^2 + y^2 == z^2 && x + y + 3 z == 1, {x, y}, Reals]
ParametricPlot[{x, y} /. soln, {z, 0, 1}]
```

Solving Nonlinear Equations

Solving large nonlinear equations with constraints

```
Clear[f, cons, vars, x];
{f, cons, vars} =
  {20 x[2] x[6]/(x[1]^2 x[4] x[5]^2) + 15 x[3] x[4]/(x[1] x[2]^2 x[5] x[7]^0.5`) +
    10 x[1] x[4]^2 x[7]^0.125`/(x[2] x[6]^3) + 25 x[1]^2 x[2]^2 x[5]^0.5` x[7]/(x[3] x[6]^2),
   {0.1` ≤ x[1] ≤ 10, 0.1` ≤ x[2] ≤ 10, 0.1` ≤ x[3] ≤ 10,
    0.1` ≤ x[4] ≤ 10, 0.1` ≤ x[5] ≤ 10, 0.1` ≤ x[6] ≤ 10,
    0.01` ≤ x[7] ≤ 10, 1 - 0.2` x[3] x[6]^(2/3) x[7]^0.25`/(x[2] x[4]^0.5`) -
      0.7` x[1]^3 x[2] x[6] x[7]^0.5`/x[3]^2 - 0.5` x[1]^0.5` x[7]/(x[3] x[6]^2) ≥ 0,
    1 - 3.1` x[2]^0.5` x[6]^(1/3)/(x[1] x[4]^2 x[5]) - 1.3` x[2] x[6]/(x[1]^0.5` x[3] x[5]) -
      0.8` x[3] x[6]^2/(x[4] x[5]) ≥ 0,
    1 - x[2] x[3]^0.5` x[5]/x[1] - 0.1` x[2] x[5]/(x[3]^0.5` x[6] x[7]^0.5`) -
      2 x[1] x[5] x[7]^(1/3)/(x[3]^1.5` x[6]) - 0.65` x[3] x[5] x[7]/(x[2]^2 x[6]) ≥ 0,
    1 - 1/x[5]^(2/3) 0.3` x[1]^0.5` x[2]^2 x[3] x[4]^(1/3) x[7]^0.25` -
      0.2` x[2] x[5]^0.5` x[7]^(1/3)/(x[1]^2 x[4]) - 0.5` x[4] x[7]^0.5`/x[3]^2 -
      0.4` x[3] x[5] x[7]^0.75`/(x[1]^3 x[2]^2) ≥ 0,
```

$$\frac{20\,x[2]\,x[6]}{x[1]^2\,x[4]\,x[5]^2}+\frac{15\,x[3]\,x[4]}{x[1]\,x[2]^2\,x[5]\,x[7]^{0.5`}}+\frac{10\,x[1]\,x[4]^2\,x[7]^{0.125`}}{x[2]\,x[6]^3}+\frac{25\,x[1]^2\,x[2]^2\,x[5]^{0.5`}\,x[7]}{x[3]\,x[6]^2}\geq 100,\ \frac{20\,x[2]\,x[6]}{x[1]^2\,x[4]\,x[5]^2}+\frac{15\,x[3]\,x[4]}{x[1]\,x[2]^2\,x[5]\,x[7]^{0.5`}}+\frac{10\,x[1]\,x[4]^2\,x[7]^{0.125`}}{x[2]\,x[6]^3}+\frac{25\,x[1]^2\,x[2]^2\,x[5]^{0.5`}\,x[7]}{x[3]\,x[6]^2}\leq 3000\Big\},\ \{x[1],\ x[2],\ x[3],\ x[4],\ x[5],\ x[6],\ x[7]\}\Big\};$$

```
TraditionalForm[Column@{f, cons}]
FindMinimum[{f, cons}, vars] // Timing
```

Solving nonlinear equations with mulitple minima

Fitting ODE to Noisy Data

Solving Implicit Equations

Integration and Differentiation

Access a vast array of knowledge, for example, to compute an indefinite integral (easily entered with the Basic Math Assistant palette).

$$\int e^{-t^2}\, dt$$

Integrate a surface over the region defined by a cone.

```
Integrate[ (x^2 + y^2) Boole[0 ≤ z ≤ 1 && x^2 + y^2 ≤ z^2],
  {x, -1, 1} , {y, -1, 1}, {z, 0, 1} ]
```

You can visualize three-dimensional regions using `RegionPlot3D`.

```
RegionPlot3D[0 ≤ z ≤ 1 && x^2 + y^2 ≤ z^2,
  {x, -1, 1}, {y, -1, 1}, {z, 0, 1}]
```

A seemingly simple function whose definite integral is anything but simple.

$$\int_0^\infty \frac{\text{Sin}[x^3]}{e^{x/2}}\, dx$$

» High oscillatory integration

SIAM challenge problem:

```
Plot[1/x Cos[Log[x]/x], {x, 0, 1},
  PlotRange → 15, Filling → Axis]
```

New methods handle highly oscillatory integration problems automatically:

```
NIntegrate[1/x Cos[Log[x]/x], {x, 0, 1}]
```

Solving Differential Equations

This system of ordinary differential equations give rise to the chaotic map known as the Lorenz attractor.

$$x'(t) = \sigma\,(y(t) - x(t))$$
$$y'(t) = \rho\,x(t) - y(t) - x(t)\,z(t)$$
$$z'(t) = x(t)\,y(t) - \beta\,z(t)$$

This gives the solution for the initial conditions $x(0) = 0$, $y(0) = 1$ and $z(0) = 0$.

```
soln = With[{σ = -3.0, ρ = 26.5, β = 1.0},
  NDSolve[{
    x'[t] == σ (x[t] - y[t]),
    y'[t] == -x[t] z[t] + ρ x[t] - y[t],
    z'[t] == x[t] y[t] - β z[t], x[0] == z[0] == 0, y[0] == 1},
   {x, y, z}, {t, 0, 200}, MaxSteps → ∞]]
```

```
ParametricPlot3D[Evaluate[{x[t], y[t], z[t]} /.soln],
 {t, 0, 200}, ColorFunction → (Hue[#4/210] &),
 ColorFunctionScaling → False, PlotPoints → 250,
 Axes → False, Background → Black,
 BoxRatios → 1, ImageSize → 7 × 72]
```

1D - Heat conduction problem with time dependent boundary conditions

Damped pendulum

Wolfram's 1D- Nonlinear wave equation.

2D modified Burger's problem with periodic boundary conditions on regular domain

Lotka-Volterra with delays

Stochastic Differential Equations example:

Visualizations

Plot the surface $\cos(x + \sin(y))$ together with the plane $\frac{x}{6} - \frac{y}{7} - \frac{1}{3}$.

```
Plot3D[{Cos[x + Sin[y]], x/6 - y/7 - 1/3},
 {x, 0, 2 π}, {y, -π, π}, PlotStyle → {Red, Blue}]
```

Using Textures - Begin with a stream plot:

```
arrows = Rasterize[StreamPlot[
   Evaluate[{Re[(x + i y)^4 - 1], -Im[(x + i y)^4 - 1]}],
   {x, 0, 3}, {y, 0, 3}, VectorScale →
    {Automatic, Automatic, Log[#5 + 1] &}, Frame → False]]
```

Layer it onto a 3D plot:

```
Plot3D[Sin[x y], {x, 0, 3}, {y, 0, 3}, Mesh → None,
 PlotStyle → Texture[arrows]]
```

More examples

Combine live video and textures

Image Processing

A simple demonstration of panning and zooming of a large image. The resizing method is adaptively selected.

```
With[{i =  },
 Manipulate[
  ImageResize[
   ImageCrop[i, {300, 300} / zoom, -pts], {300, 300},
   Resampling → ControlActive["Nearest", "Lanczos"]],
  {{pts, {0, 0}}, {-1, -1}, {1, 1}},
  {{zoom, 1}, 1, 10}
 ]]
```

Geometric Transformations

Feature Detection

Live Image Capture

Control Systems

Mathematica introduces a unifying symbolic representation of transfer function and state space models.

Get a state space model from differential equations for coupled damped oscillators:

```
model = StateSpaceModel[
  {2 x2[t] - x3[t] + 3 x2'[t]/2 - x3'[t]/2 + x2''[t] == F[t],
   -x2[t] + x3[t] - x2′[t]/2 + x3′[t]/2 + 2 x3′′[t] == 0},
  {{x2[t], 0}, {x2′[t], 0}, {x3[t], 0}, {x3′[t], 0}},
  {{F[t], 0}}, {x2[t], x3[t]}, t]
```

Convert between state space representation and classical transfer function representation:

```
TransferFunctionModel[model]
```

Immediately ask questions about the system.

Can the model be fully controlled with the specified input `F[t]`?

```
ControllableModelQ[model]
```

Simulate the model with varying driving frequency:

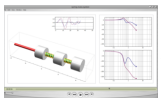
```
Plot[Evaluate[OutputResponse[model,
    Sin[2^(t/15)], {t, 0, 100}]], {t, 0, 100}]
```

Analyze the model:

Plot frequency response gain for amplitude and phase:

```
BodePlot[model, GridLines → Automatic,
 ImageSize → Medium, GridLinesStyle → GrayLevel[2/3]]
```

Combine with *Mathematica*'s 3D graphics, animate, and export:

```
Button[  ,
  SystemOpen[FileNameJoin[{NotebookDirectory[],
     "Movies", "SpringMassDamperSystem.mov"}]]]
```

Wavelets

Wavelets localize noise and features in frequency and time simultaneously, allowing more selective filtering and analysis:

Remove one of two overlapping signals.

Noisy signal.

```
f[x_] := Sin[50 π x] Exp[-100 π (x - 0.5)^2] +
   (Sin[50 π x] + 2 Cos[140 π x]) Exp[-50 π (x - 0.2)^2] +
   2 Sin[150 π x] Exp[-80 π (x - 0.8)^2];
data = Table[f[x], {x, 0., 1, 1/2047}];
ListLinePlot[data, PlotRange → All]
```

Wavelet transform:

```
cwt = ContinuousWaveletTransform[data,
   GaborWavelet[6], {10, 16}, Padding → 0.0,
   SampleRate → 2047, WaveletScale → Automatic]
WaveletScalogram[cwt, ColorFunction → "SolarColors",
 AxesLabel → {"time", "scale"}]
```

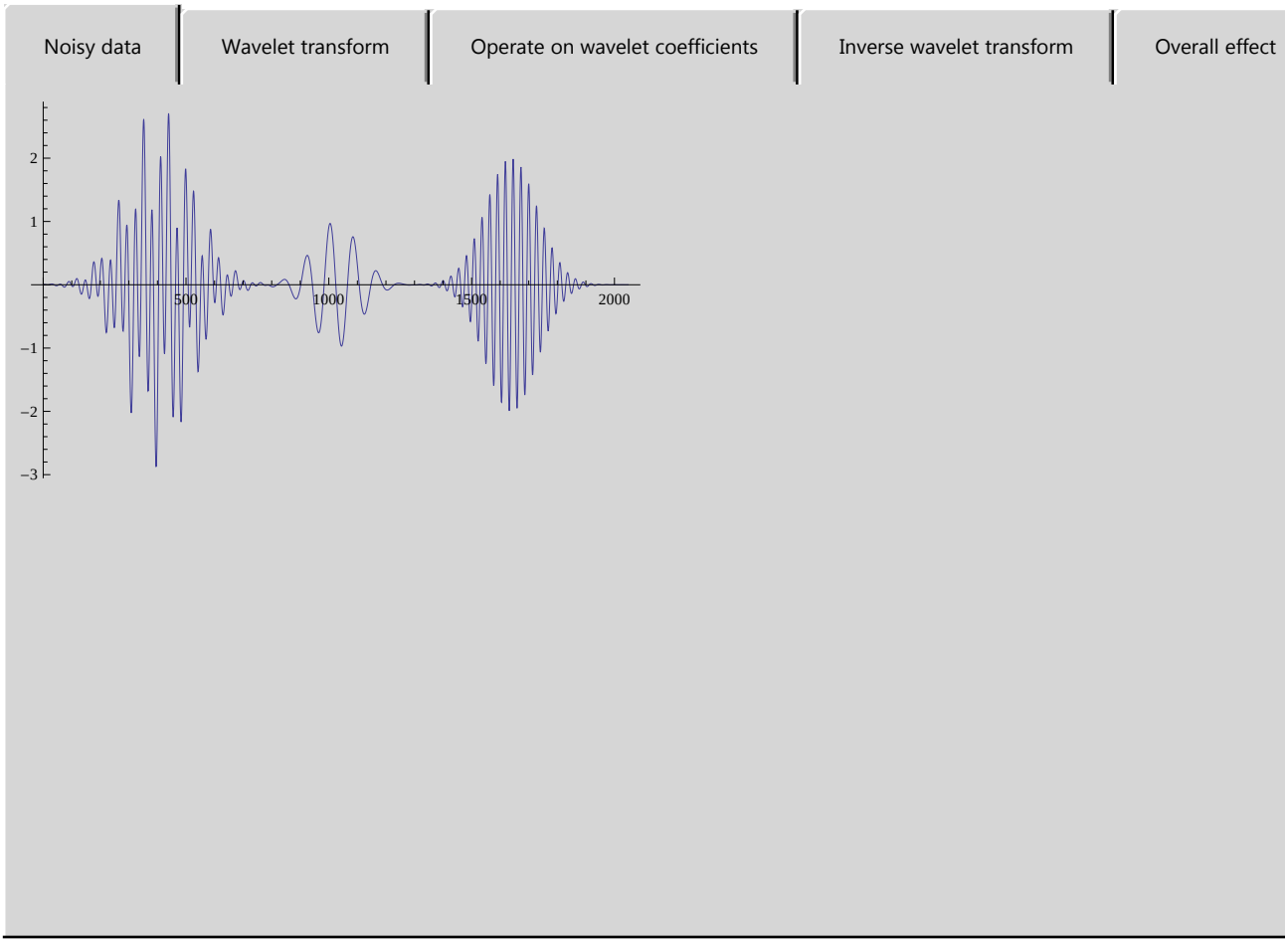
Filter out one feature:

```
vec = ConstantArray[1, Length[data]];
vec[[1 ;; 1000]] := 0.0;
cwtThresh = WaveletMapIndexed[
    #1 vec &, cwt, {{4 | 5, _}, {6, u_ /; u < 8}}];
WaveletScalogram[cwtThresh, ColorFunction → "SolarColors",
 AxesLabel → {"time", "scale"}]
```

Inverse wavelet transform:

```
rdata = InverseContinuousWaveletTransform[cwtThresh];
ListLinePlot[rdata, PlotRange → All]
```

Overall:



De-noising an image

Import and Export

Mathematica aims to support as many standard data formats as possible. With the release of *Mathematica* 8 the number of supported formats rises from 145 to 172.

New formats:

» Arrays: `Affymetrix`†, `NASACDF`

» Calendar: `ICS`†

» Fonts: `BDF`†

» Geodesy/geospatial: `ArcGRID`†, `KML`, `NDK`†, `SurferGrid`, `TLE`†

» Graphics: `EPS`, `VTK`

» Meterology: `GRIB`†

» Molecular: `CIF`†

» Spreadsheet: `XLSX`

» Other: `NEXUS`, `JSON`

Full listing of all import and export formats:

```
$ImportFormats
```

```
$ExportFormats
```

† Import only

Import the first sheet of an XLSX file:

```
Import["ExampleData/cities.xlsx", {"Data", 1}]
```

Real-time video processing from a camera:

```
Dynamic[EdgeDetect[CurrentImage[]]]
```

Programming and Development

Mathematica's multiparadigm language and broad range of constructs makes it a perfect choice for constructing programs at all levels.

	`f = Factorial[n]`
	`f[n_] := n!`
Rule–based	`f[n_] := n f[n - 1];` `f[1] = 1`

Procedural	`f[n_] :=`
	`  Module[{t = 1}, Do[`
	`  t = t i, {i, n}]; t]`

	`f[n_] :=`
	`  Module[{t = 1, i},`
	`  For[i = 1, i ≤ n,`
	`  i++, t *= i]; t]`

List–based	`f[n_] :=` `  Times @@ Range[n]`
	`f[n_] := Fold[` `  Times, 1, Range[n]]`
Recursive	`f[n_] := If[n == 1,` `  1, n f[n - 1]]`

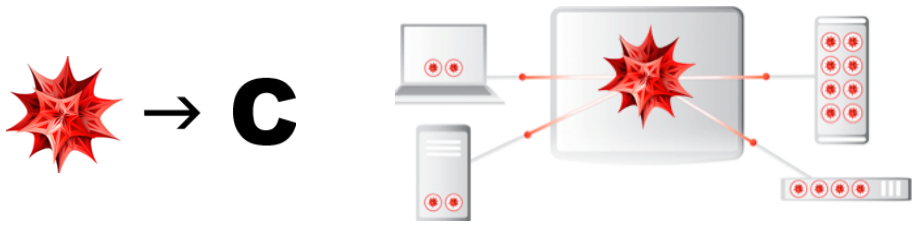
Functional	`f = If[#1 == 1, 1, #1 #0[#1 - 1]] &`
	`f[n_] := Fold[#2[#1] &, 1, Array[Function[t, #1 t] &, n]]`
Constructive	`f[n_] := Length[Permutations[Range[n]]]`
Pattern–based	`f[n_] := First[{1, n} //. {a_, b_ /; b > 0} :> {b a, b - 1}]`
String–based	`f[n_] := StringLength[Fold[StringJoin[Table[#1, {#2}]] &, "A", Range[n]]]`
Mathematical	`f[n_] := Gamma[n + 1]`
	`f[n_] := ∏_i^n i`

» Simulating the diffusion model.

» Theo Jensen walking linkage.

Programming and Development: Compile

In *Mathematica* 8 the Compile function can automatically generate C code, compile it, link dynamically to *Mathematica* at run-time, and run in parallel (multi-threading) when possible.



Create a compiled function that runs in native code and in parallel:

```
f = Compile[{{c, _Complex}},
  Module[{num = 1}, FixedPoint[(num++; #1^2 + c) &, 0,
    99, SameTest → (Re[#1]^2 + Im[#1]^2 ≥ 4 &)]; num],
  CompilationTarget → "C", RuntimeAttributes → {Listable},
  Parallelization → True]
```

Easily plot 200000 evaluations of the function:

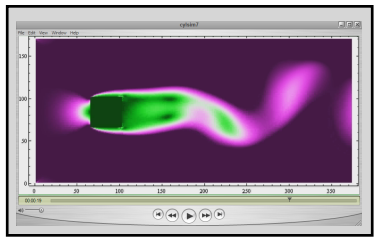
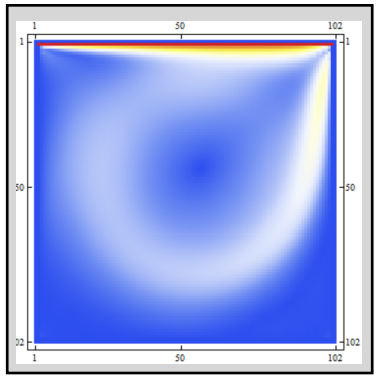
```
ArrayPlot[
 f[Table[x + i y, {x, -2, 1/2, 0.005}, {y, -1, 1, 0.005}]],
 ColorFunction → "Rainbow"]
```

Explore the Mandelbrot set interactively using the natively compiled function:

Code

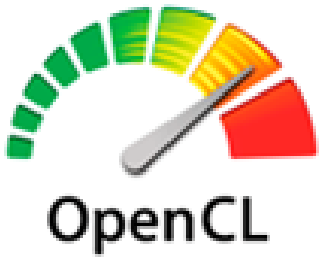
» Solving the 2D Navier-Stokes Equations.

Generate optimized code for solving fluid problems in 2D using Lattice Boltzmann method

`Flow around a 2D Cylinder`	
`Flow in a driven cavity`	

» C code generation

Programming and Development: CUDA and OpenCL Support



- CUDA is a parallel computing architecture from NVidia
- Uses GPUs to carry out massively parallel computations
- OpenCL is a unification of CUDA concepts for GPUs other than those from NVidia
- Integration with Symbolic C and the C Compiler interface to significantly help develop and execute applications.

Load the CUDA package and get information on the hardware.

```
Needs["CUDALink`"]
```

```
CUDAInfoTable[CUDAInformation[]]
```

Lots of CUDA functions:

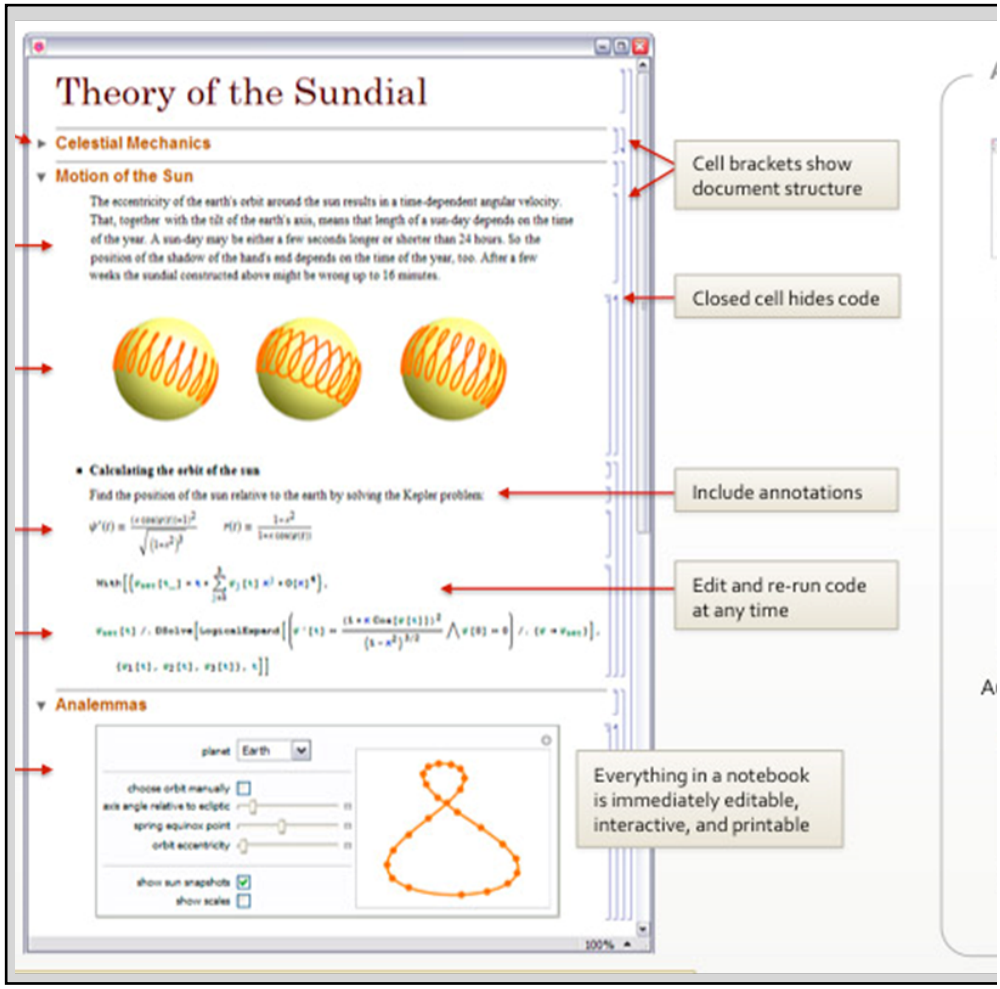
```
?CUDA*
```

```
CUDAFluidDynamics[ImageSize → 800]
```

Image Processing

Linear Algebra

Working with Documents

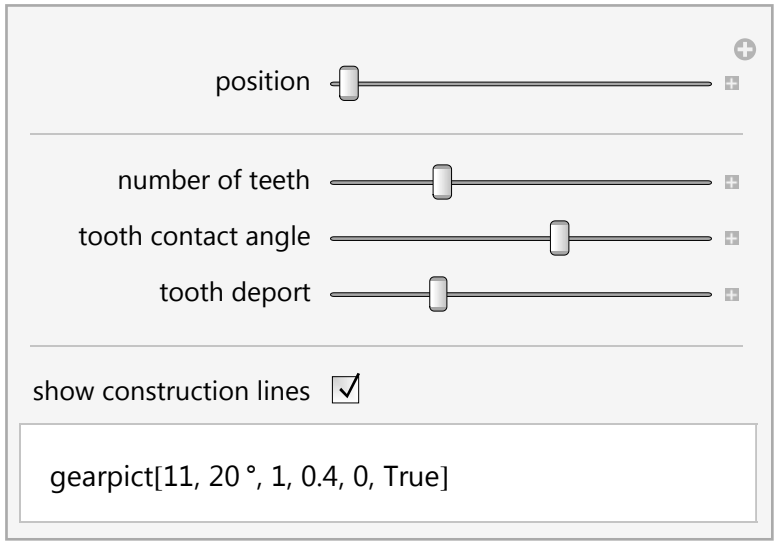


» "Good science is good observation" - James Cameron.

» homework assignment from computational physics course.

» Making teaching material for students.

» Making interactive material.


position
number of teeth
tooth contact angle
tooth deport
show construction lines
gearpict[11, 20 °, 1, 0.4, 0, True]

Connection Technologies

Mathematica can connect to external applications like databases, spreadsheets, or legacy C or Fortran. These external applications can then be manipulated within *Mathematica* as symbolic expressions (see the guide to Calling External Programs).

» ***Wolfram Workbench***: provides typical features of a state-of-the-art integrated development environment (IDE), customized and specialized for *Mathematica* and other Wolfram technologies (see www.wolfram.com/products/workbench).

» ***J/Link***™: *Mathematica* ships with a well-developed link to Java that allows you to interactively work with a Java runtime, accessing anything that the runtime can work with (see the Java Interface guide).

» **web*Mathematica***™: web*Mathematica* allows you to show results on the web, and more importantly to allow others to directly interact with your computations (see examples at www.wolfram.com).

» ***DatabaseLink***: *DatabaseLink* is a *Mathematica* application that provides a set of tools allowing convenient integration of *Mathematica* with database management systems. With *DatabaseLink* you can use all *Mathematica*'s numeric/symbolic/programming/visualization capabilities with data taken from SQL database applications (see the guide to `Database Connectivity`).

» ***Web Services Link***: The web services client for *Mathematica* allows users to call operations that are based on other platforms or languages that are not immediately accessible to *Mathematica* (see the reference InstallService).

» ***.Net/Link***™: *.Net/Link* integrates *Mathematica* and Microsoft's .NET platform; call .NET from *Mathematica* in a completely transparent way; use and control the *Mathematica* kernel from a .NET program (see the guide to .NET Interface).

Mathematica Community

There is a vast *Mathematica* community that has built up around the world and that can be tapped in a variety of ways.

» Wolfram Demonstrations Project: demonstrations.wolfram.com

» Applications packages such as *Rayica* or *Neural Networks*

» Hundreds of books

» Web resources such as Wolfram Library Archive and *MathWorld*™

» Training offered by Wolfram Education Group

» Forums and user groups such as MathGroup

» Conferences such as the annual International *Mathematica* User Conference or the International *Mathematica* Symposium

Initializations

» General

```
$DataDirectory = FileNameJoin[
    {NotebookDirectory[], "DataAndNotebooks"}];
AppendTo[$Path, $DataDirectory];
```

» RandomSparseArray

```
RandomSparseArray[{n_, m_}] := SparseArray[{
    Band[{1, 1}] :> RandomReal[],
    Band[{1, 2}] :> RandomReal[],
    Band[{2, 1}] :> RandomReal[],
    Band[{n - 1, 1}, {n, 2}] → RandomReal[1, {2, 2}],
    Band[{1, n - 1}, {2, n}] → RandomReal[1, {2, 2}]
   }, {n, m}]
```

DisplayPictureCell

```
DisplayPictureCell[expr_] := CellPrint[
  TextCell[expr, "Picture", ShowStringCharacters → False]]
```

DisplayOption and TTDisplayOption

```
DisplayOption[a_List] :=
  StringJoin[Riffle[DisplayOption /@ a, " "]];
DisplayOption[a_String] :=
  StringTrim[StringJoin[StringSplit[a,
     z : RegularExpression["[A-Z]"] :> " " <> z]]];
TTDisplayOption[expr_] := Tooltip[
    Style[DisplayOption[expr], FontFamily → "Times"],
    Labeled[Grid[List /@ FinancialDerivative[expr],
      Alignment → Left], Style["Parameters", Bold], Top]];
SetOptions[EvaluationNotebook[],
  StyleDefinitions → Notebook[{Cell[CellGroupData[
```

```
{Cell[TextData[{"Style Definitions for \n",
    StyleBox["Wolfram Seminars",
     FontSlant → "Italic"]}], "Subtitle"],
 Cell[StyleData[StyleDefinitions → "Default.nb"]],
 Cell[StyleData[StyleDefinitions →
    "StyleMenuClear.nb"]], Cell[CellGroupData[
   {Cell["Style Environment Definitions",
     "Section"], Cell[StyleData[All,
      "SlideShow", StyleDefinitions →
       StyleData[All, "Presentation"]],
     PageWidth → WindowWidth, DockedCells → Cell[
       BoxData[GridBox[{{TooltipBox[ButtonBox[
             StyleBox[RowBox[{"  ", "□", " "}],
              FontSize → 20, GrayLevel[0.7`]],
             Appearance → None, ButtonFunction :→
              (FEPrivate`NotebookToggleFullScre:
                 en[]; {}),
             Evaluator → None], DynamicBox[
             FEPrivate`FrontEndResource[
              "SlideshowToolbarText",
              "TooltipToggleFullScreen"]]],
           StyleBox[RowBox[{TooltipBox[
              ButtonBox[StyleBox[
                RowBox[{" ", "⏮", " "}],
                FontSize → 25, GrayLevel[0.7`]],
               ButtonData → "ScrollPageFirst",
               Appearance → None], DynamicBox[
               FEPrivate`FrontEndResource[
                "SlideshowToolbarText",
                "TooltipFirstSlide"]]],
             "", TooltipBox[ButtonBox[
               StyleBox[RowBox[{" ", "◂", " "}],
                FontSize → 25, GrayLevel[
                 0.7`]], ButtonData →
```

```
        "ScrollPagePrevious",
       Appearance → None], DynamicBox[
       FEPrivate`FrontEndResource[
        "SlideshowToolbarText",
        "TooltipPreviousSlide"]]],
     "", TooltipBox[ButtonBox[
       StyleBox[RowBox[{" ", "‣", " "}],
        FontSize → 25, GrayLevel[0.7`]],
       ButtonData → "ScrollPageNext",
       Appearance → None], DynamicBox[
       FEPrivate`FrontEndResource[
        "SlideshowToolbarText",
        "TooltipNextSlide"]]],
     "", TooltipBox[ButtonBox[
       StyleBox[RowBox[{" ", "⏭", " "}],
        FontSize → 25, GrayLevel[0.7`]],
       ButtonData → "ScrollPageLast",
       Appearance → None], DynamicBox[
       FEPrivate`FrontEndResource[
        "SlideshowToolbarText",
        "TooltipLastSlide"]]]}],
  ButtonBoxOptions → {Alignment →
     {Automatic, FEPrivate`If[
       FEPrivate`$OperatingSystem ===
        "Unix", -0.4`, -0.5`]},
    Appearance → "Palette",
    ButtonFunction :→
     (FEPrivate`FrontEndExecute[
        FrontEnd`FrontEndToken[
         FrontEnd`ButtonNotebook[],
         #1]] &), Evaluator → None,
    FrameMargins → 0, ImageMargins →
     0, ImageSize → Medium}],
 DynamicModuleBox[{cells$$ = {},
```

```
tagFind$$ = False}, TagBox[
DynamicBox[PopupMenuBox[Dynamic[0,
    With[{nb$ = ButtonNotebook[]},
      SelectionMove[nb$, Before,
       Notebook, AutoScroll → False];
      If[tagFind$$, Do[NotebookFind[
       nb$, "SlideShowHeader", Next,
       CellTags, AutoScroll → False],
       {#1}], Do[NotebookFind[nb$,
       "SlideShowNavigationBar",
       Next, CellStyle, AutoScroll →
       False], {#1}]]; SelectionMove[
       nb$, After, Cell, AutoScroll →
       False]; NotebookWrite[nb$,
       Cell["", Deletable → True,
       ShowCellBracket → False],
       All]; NotebookDelete[nb$];
      SelectionMove[nb$, If[
       tagFind$$, Previous, Next],
       Cell]] &], cells$$,
   StyleBox[RowBox[{DynamicBox[
       CurrentValue[{"CounterValue",
       "SlideShowNavigationBar"}]],
      DynamicBox[
       FEPrivate`FrontEndResource[
       "SlideshowToolbarText",
       "SlideshowToolbarCounterText"
       ]],
      DynamicBox[CurrentValue[
       {"MaxCounterValue",
       "SlideShowNavigationBar"}]]}]
    ,
   "ControlStyle"], Appearance →
   None, ImageSize → Automatic],
```

```
ImageSizeCache → {50.`,
  {6.`, 18.`}}], EventHandlerTag[
{"MouseDown" :→ Module[{cnt$, nb$ =
     InputNotebook[]}, cells$$ =
    {}; cnt$ = Apply[(NotebookFind[
     nb$, #1, All, #2, AutoScroll →
     False]; CurrentValue[nb$,
     {"CellCount", True}]) &,
     {{"SlideShowNavigationBar",
     CellStyle},
     {"SlideShowHeader",
     CellTags}}, {1}]; cnt$ =
    (If[tagFind$$ = #1 == 0,
     #2, #1] &) @@ cnt$;
   SelectionMove[nb$, Before,
    Notebook, AutoScroll → False];
   If[tagFind$$, cells$$ =
     (#1 → "Slide " <> ToString[
     #1] &) /@ Range[cnt$],
    Do[NotebookFind[nb$,
     "SlideShowNavigationBar",
     Next, CellStyle, AutoScroll →
     False]; SelectionMove[nb$,
     Next, Cell, AutoScroll →
     False]; AppendTo[cells$$,
     NotebookRead[nb$] /.
     s_StyleBox :→ With[{ss = Join[
     Take[s, 1], DeleteCases[Rest[
     s], _String | _[FontColor,
     _]]]}, ss /; True]], {cnt$}];
    cells$$ = Replace[cells$$[[
     All, 1]], {BoxData[c_] :→ c,
     TextData[c_] :→ RowBox[
     If[ListQ[c], c, {c}]]}, 1];
```

```
                cells$$ = Thread[Range[
                  cnt$] → cells$$]]],
              PassEventsDown → True,
              EvaluationOrder → Before,
              PassEventsDown → Automatic,
              PassEventsUp → True}]],
          DynamicModuleValues :→ {}]}},
      AutoDelete → False, GridBoxItemSize →
       {"Columns" → {Automatic,
          Fit, Automatic}}]],
   "DockedCell", CellDynamicExpression :→
    If[CurrentValue[EvaluationNotebook[],
       {TaggingRules,
        "DisableController"}] ===
      True, Null, (FrontEndExecute[
         FrontEnd`FrontEndToken[
          FrontEnd`InputNotebook[], #1]] &)[
      Switch[ControllerState["Apple IR",
        {"B1", "B2"}],
       {True, _}, "ScrollPagePrevious",
       {_, True}, "ScrollPageNext",
       _, None]]], Magnification → 0.75`],
ScrollingOptions → {"PagewiseDisplay" → True,
  "VerticalScrollRange" → Fit},
ShowGroupOpener → False, MenuSortingValue →
 1300], Cell[StyleData["DockedCell"],
Editable → False, Evaluator → "Local",
PageWidth → WindowWidth, CellFrame →
 {{0, 0}, {0, 0}}, ShowSelection → True,
CellMargins → {{0, 0}, {0, 0}},
CellElementSpacings → {"ClosedCellHeight" →
   0, "ClosedGroupTopMargin" → 0},
DynamicUpdating → True, CellFrameMargins →
 {{0, 0}, {0, 4}}, ContextMenu → None,
```

```
    ComponentwiseContextMenu → {},
    ShowStringCharacters → False,
    LineBreakWithin → False, Magnification → 1,
    Background → GrayLevel[0.97`],
    TooltipBoxOptions →
     {TooltipDelay → Automatic}]},
  Open]], Cell[CellGroupData[
  {Cell["Styles for Headings", "Section"],
   Cell[CellGroupData[{Cell[StyleData["Title"],
       MenuSortingValue → 1, MenuCommandKey →
        "1", FontFamily → "Georgia", FontSize →
        36, FontWeight → Plain, FontColor →
        RGBColor[0.796078`, 0.0705882`, 0]],
      Cell[StyleData["Title", "Presentation"],
       CellMargins → {{90, 100}, {10, 100}},
       FontSize → 54], Cell[StyleData["Title",
        "SlideShow"]]}, Closed]], Cell[
    CellGroupData[{Cell[StyleData["Subtitle"],
       MenuSortingValue → 100, MenuCommandKey →
        "2", FontFamily → "Georgia", FontSize →
        24, FontWeight → "Plain", FontSlant →
        "Italic"], Cell[StyleData["Subtitle",
        "Presentation"], CellMargins →
        {{90, 100}, {100, 20}}, FontSize → 32],
      Cell[StyleData["Subtitle", "SlideShow"]]},
     Closed]], Cell[CellGroupData[
     {Cell[StyleData["Subsubtitle"],
       MenuSortingValue → 200, MenuCommandKey →
        "3", FontFamily → "Georgia", FontSize →
        20, FontWeight → "Plain", FontSlant →
        "Italic", FontColor → GrayLevel[0.2`]],
      Cell[StyleData["Subsubtitle",
        "Presentation"], CellMargins →
        {{90, 20}, {2, 120}}, LineSpacing →
```

```
      {1.25`, 1}, FontSize → 26], Cell[
      StyleData["Subsubtitle", "SlideShow"]]},
    Closed]], Cell[CellGroupData[
    {Cell[StyleData["DateSection"],
      CellMargins → {{30, 0}, {12, 24}},
      ShowStringCharacters → False], Cell[
      StyleData["DateSection", "Presentation"],
      CellMargins → {{90, 20}, {2, 36}},
      ShowStringCharacters → False, FontSize →
       21], Cell[StyleData["DateSection",
       "SlideShow"]]}, Closed]]},
Closed]], Cell[CellGroupData[
{Cell["Styles for Sections", "Section"], Cell[
  CellGroupData[{Cell[StyleData["Chapter"],
     CellMargins → {{27, Inherited}, {8, 34}},
     CounterIncrements → {"Chapter"},
     MenuSortingValue → 2410, FontFamily →
      "Georgia", FontSize → 28, FontColor →
      RGBColor[0.811765`, 0.117647`,
       0.145098`], CellTags → "Chapter"],
    Cell[StyleData["Chapter", "Presentation"],
     CellFrame → None, CellMargins →
      {{80, 60}, {20, 30}}, FontSize → 36],
    Cell[StyleData["Chapter",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "LargeChapter", StyleDefinitions →
       StyleData["Chapter"]], CellMargins →
      {{27, Inherited}, {8, 34}},
     MenuSortingValue → 2430, FontFamily →
      "Georgia", FontSize → 30,
     FontColor → RGBColor[0.811765`,
       0.117647`, 0.145098`]],
    Cell[StyleData["LargeChapter",
```

```
      "Presentation"], CellMargins →
      {{100, 100}, {10, 150}}, FontSize → 54],
    Cell[StyleData["LargeChapter",
      "SlideShow"]]}, Closed]], Cell[
  CellGroupData[{Cell[StyleData["Section"],
    CellFrame → 0, MenuSortingValue → 1200,
    MenuCommandKey → "4", FontFamily →
     "Georgia", FontSize → 24, FontWeight →
     Plain, FontColor → RGBColor[
      0.811765`, 0.117647`, 0.145098`]],
   Cell[StyleData["Section", "Presentation"],
    CellFrame → None, CellMargins →
     {{80, 14}, {30, 30}}, FontFamily →
     "Georgia", FontSize → 34, FontWeight →
     Plain, FontColor → RGBColor[0.811765`,
      0.117647`, 0.145098`]], Cell[StyleData[
     "Section", "SlideShow"]]}, Open]],
 Cell[CellGroupData[{Cell[StyleData[
     "Subsection"], CellDingbat → "»",
    CellGroupingRules → {"SectionGrouping",
      40}, MenuSortingValue → 1300,
    MenuCommandKey → "5", FontFamily →
     "Georgia", FontSize → 22, FontWeight →
     Plain, FontColor → GrayLevel[0.25`]],
   Cell[StyleData["Subsection",
     "Presentation"], CellMargins →
     {{80, 60}, {12, 20}}, FontSize → 25],
   Cell[StyleData["Subsection",
     "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
     "Subsubsection"], CellDingbat →
     "»", MenuSortingValue → 1400,
    MenuCommandKey → "6", FontFamily →
     "Georgia", FontSize → 20, FontWeight →
```

```
      Plain, FontColor → GrayLevel[0.3`]],
    Cell[StyleData["Subsubsection",
      "Presentation"], CellMargins →
      {{88, 60}, {0, 12}}, FontSize → 22],
    Cell[StyleData["Subsubsection",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "MoreExamplesSection",
      StyleDefinitions →
       StyleData["Subsubsection"]],
     CellDingbat → None, ShowGroupOpener →
      True, CellGroupingRules →
      {"SectionGrouping", 45},
     MenuSortingValue → 2100, FontColor →
      RGBColor[0.636862`, 0.0564706`, 0.`]],
    Cell[StyleData["MoreExamplesSection",
      "Presentation"], CellMargins →
      {{88, 60}, {20, 12}}], Cell[StyleData[
      "MoreExamplesSection", "SlideShow"],
     CellMargins → {{88, 60}, {20, 12}},
     WholeCellGroupOpener → True]}, Closed]]},
Closed]], Cell[CellGroupData[
{Cell["Styles for Text", "Section"],
 Cell[CellGroupData[
   {Cell[StyleData["Text"], MenuSortingValue →
      2000, MenuCommandKey → "7", FontFamily →
      "Georgia", FontSize → 18],
    Cell[StyleData["Text", "Presentation"],
     CellMargins → {{80, 20}, {12, 12}},
     LineSpacing → {1.25`, 1}, FontFamily →
      "Georgia", FontSize → 20],
    Cell[StyleData["Text", "SlideShow"]]},
   Closed]], Cell[CellGroupData[
   {Cell[StyleData["MathCaption"],
```

```
      MenuSortingValue → 2010, MenuCommandKey →
       "8", FontFamily → "Georgia", FontSize →
       18, FontColor → GrayLevel[0]], Cell[
      StyleData["MathCaption", "Presentation"],
      CellMargins → {{120, 12}, {6, 34}},
      LineSpacing → {1.25`, 1}, FontSize → 20],
     Cell[StyleData["MathCaption",
       "SlideShow"]]}, Closed]],
  Cell[CellGroupData[{Cell[StyleData[
       "IntegrationMathCaption",
       StyleDefinitions →
        StyleData["MathCaption"]],
      CellMargins → {{0, 12}, {2, 24}},
      CellFrameMargins → {{0, 0}, {0, 3}},
      CellFrameLabels → {{PaneBox[
          FrameBox[StyleBox["Integration",
            FontFamily → "Helvetica", FontSize →
             9, FontColor → GrayLevel[1],
            FontWeight → Plain], Background →
            RGBColor[0.811765`, 0.117647`,
             0.145098`], BoxFrame → 0,
           FrameStyle → None, FrameMargins →
            {{3, 3}, {1, 1}}], ImageSize → 56,
          Alignment → Left, FrameMargins → 0],
         None}, {None, None}}], Cell[
      StyleData["IntegrationMathCaption",
       "Presentation"], CellMargins →
       {{30, 12}, {12, 12}}, CellFrameMargins →
       {{10, 0}, {0, 5}}, FontSize → 20],
     Cell[StyleData["IntegrationMathCaption",
       "SlideShow"], CellMargins →
       {{30, 12}, {12, 12}}, CellFrameMargins →
       {{10, 0}, {0, 5}}]}, Closed]],
  Cell[CellGroupData[{Cell[StyleData[
```

```
      "AutomationMathCaption",
      StyleDefinitions → StyleData[
        "IntegrationMathCaption"]],
     CellMargins → {{0, 12}, {2, 24}},
     CellFrameMargins → {{0, 0}, {0, 3}},
     CellFrameLabels → {{PaneBox[
         FrameBox[StyleBox["Automation",
           FontFamily → "Helvetica", FontSize →
            9, FontColor → GrayLevel[1],
           FontWeight → Plain], Background →
           RGBColor[0.811765`, 0.117647`,
            0.145098`], BoxFrame → 0,
          FrameStyle → None, FrameMargins →
           {{3, 3}, {1, 1}}], ImageSize → 56,
         Alignment → Left, FrameMargins → 0],
        None}, {None, None}}],
    Cell[StyleData["AutomationMathCaption",
      "Presentation"], CellMargins →
      {{30, 12}, {12, 12}}, FontSize → 20],
    Cell[StyleData["AutomationMathCaption",
      "SlideShow"], CellMargins →
      {{30, 12}, {12, 12}}]}, Closed]],
  Cell[CellGroupData[{Cell[StyleData[
      "Comment", StyleDefinitions →
       StyleData["Text"]], CellMargins →
      {{0, 12}, {2, 24}}, CellFrameMargins →
      {{0, 0}, {0, 3}}, CellFrameLabels →
      {{PaneBox[FrameBox[StyleBox["Comment",
           FontFamily → "Helvetica", FontSize →
            9, FontColor → GrayLevel[1],
           FontWeight → Plain], Background →
           RGBColor[1.`, 0.647059`, 0],
          BoxFrame → 1, FrameStyle → RGBColor[
           1.`, 0.647059`, 0], FrameMargins →
```

```
                {{3, 3}, {1, 1}}], ImageSize → 56,
              Alignment → Left, FrameMargins → 0],
            None}, {None, None}},
          MenuSortingValue → 2200, FontSize →
           14], Cell[StyleData["Comment",
           "Presentation"], CellMargins →
           {{30, 12}, {12, 12}}, FontSize → 16],
         Cell[StyleData["Comment", "SlideShow"],
          CellMargins → {{30, 12}, {12, 12}},
          FontSize → 16]}, Closed]],
       Cell[CellGroupData[{Cell[StyleData[
           "DisplayFormula"], CellMargins →
           {{60, Inherited}, {Inherited,
             Inherited}}, DefaultFormatType →
           DefaultInputFormatType,
          HyphenationOptions →
           {"HyphenationCharacter" → "⁖"},
          LanguageCategory → "Formula",
          ScriptLevel → 0, SingleLetterItalics →
           True, MenuSortingValue →
           1700, UnderoverscriptBoxOptions →
           {LimitsPositioning → True},
          StripStyleOnPaste → True], Cell[StyleData[
           "DisplayFormula", "Presentation"],
          CellMargins → {{120, Inherited},
            {1.75` Inherited, 1.75` Inherited}},
          LineSpacing → {1, 5}, FontFamily →
           "Georgia", FontSize → 20],
         Cell[StyleData["DisplayFormula",
           "SlideShow"]]}, Closed]]}, Open]],
   Cell[CellGroupData[{Cell["Styles for Items",
      "Section"],
     Cell[CellGroupData[{Cell[StyleData["Item"],
         CellDingbat → "» ", LineSpacing →
```

```
      {1.5`, 0}, MenuSortingValue → 2700,
     FontFamily → "Georgia", FontSize →
      18, FontColor → GrayLevel[0.2`]],
    Cell[StyleData["Item", "Presentation"],
     CellMargins → {{116, 20}, {12, 12}},
     FontFamily → "Georgia", FontSize → 20],
    Cell[StyleData["Item", "SlideShow"]]},
   Closed]], Cell[CellGroupData[
   {Cell[StyleData["ItemParagraph"],
     MenuSortingValue → 2710, FontFamily →
      "Georgia", FontSize → 18, FontColor →
      GrayLevel[0.2`]], Cell[StyleData[
      "ItemParagraph", "Presentation"],
     CellMargins → {{120, 90}, {12, 12}},
     LineSpacing → {1.25`, 0}, FontFamily →
      "Georgia", FontSize → 20],
    Cell[StyleData["ItemParagraph",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "ItemNumbered"], LineSpacing →
      {1.5`, 0}, MenuSortingValue → 2720,
     FontFamily → "Georgia", FontSize →
      18, FontColor → GrayLevel[0.25`]],
    Cell[StyleData["ItemNumbered",
      "Presentation"], CellMargins →
      {{116, 20}, {12, 12}}, FontFamily →
      "Georgia", FontSize → 20],
    Cell[StyleData["ItemNumbered",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "Subitem"], CellDingbat →
      "» ", LineSpacing → {1.5`, 0},
     MenuSortingValue → 2730, FontFamily →
      "Georgia", FontSize → 16, FontColor →
```

```
      GrayLevel[0.3`]], Cell[StyleData[
      "Subitem", "Presentation"],
     CellMargins → {{136, 20}, {10, 10}},
     FontFamily → "Georgia", FontSize → 21],
    Cell[StyleData["Subitem",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "SubitemParagraph"], LineSpacing →
      {1.5`, 0}, MenuSortingValue → 2740,
     FontFamily → "Georgia", FontSize →
      16, FontColor → GrayLevel[0.3`]],
    Cell[StyleData["SubitemParagraph",
      "Presentation"], CellMargins →
      {{136, 90}, {10, 8}}, FontFamily →
      "Georgia", FontSize → 18],
    Cell[StyleData["SubitemParagraph",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "SubitemNumbered"], LineSpacing →
      {1.5`, 0}, MenuSortingValue → 2750,
     FontFamily → "Georgia", FontSize →
      16, FontColor → GrayLevel[0.3`]],
    Cell[StyleData["SubitemNumbered",
      "Presentation"], CellMargins →
      {{136, 20}, {10, 10}}, FontFamily →
      "Georgia", FontSize → 18],
    Cell[StyleData["SubitemNumbered",
      "SlideShow"]]}, Closed]]},
Closed]], Cell[CellGroupData[
{Cell["Styles for Input and Output",
  "Section"],
 Cell[CellGroupData[{Cell[StyleData[
      "Input"], MenuSortingValue → 2020,
     MenuCommandKey → "9", FontSize → 18],
```

```
    Cell[StyleData["Input", "Presentation"],
     CellMargins → {{120, 10}, {7, 12}},
     FontSize → 18], Cell[StyleData[
      "Input", "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "CellLabel"], CellLabelMargins →
      {{12, Inherited}, {Inherited,
        Inherited}}, LanguageCategory → None,
     StyleMenuListing → None, FontFamily →
      "Helvetica", FontSize → 9, FontColor →
      RGBColor[0.269993`, 0.308507`, 0.6`]],
    Cell[StyleData["CellLabel",
      "Presentation"], CellMargins →
      {{12, Inherited}, {1.5` Inherited,
        1.5` Inherited}}, FontSize → 12],
    Cell[StyleData["CellLabel", "SlideShow"],
     CellMargins → {{12, Inherited},
       {1.5` Inherited, 1.5` Inherited}},
     FontSize → 12]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "WolframAlphaShort"], FontSize → 18],
    Cell[StyleData["WolframAlphaShort",
      "Presentation"], CellMargins →
      {{120, 10}, {7, 12}}, FontSize → 18],
    Cell[StyleData["WolframAlphaShort",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "WolframAlphaShortInput"],
     FontSize → 18], Cell[StyleData[
      "WolframAlphaShortInput",
      "Presentation"], CellMargins →
      {{120, 10}, {7, 12}}, FontSize → 18],
    Cell[StyleData["WolframAlphaShortInput",
      "SlideShow"]]}, Closed]],
```

```
 Cell[CellGroupData[{Cell[StyleData[
      "WolframAlphaLong"], FontFamily →
      "Helvetica", FontSize → 18],
    Cell[StyleData["WolframAlphaLong",
      "Presentation"], CellMargins →
      {{120, 10}, {7, 12}}, FontSize → 18],
    Cell[StyleData["WolframAlphaLong",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "WolframAlphaLongInput"], FontSize → 18],
    Cell[StyleData["WolframAlphaLongInput",
      "Presentation"], CellMargins →
      {{120, 10}, {7, 12}}, FontSize → 18],
    Cell[StyleData["WolframAlphaLongInput",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "Output"], MenuSortingValue →
      2030, FontSize → 18], Cell[StyleData[
      "Output", "Presentation"], CellMargins →
      {{120, 10}, {7, 12}}, FontSize → 18],
    Cell[StyleData["Output", "SlideShow"]]},
   Closed]], Cell[CellGroupData[
   {Cell[StyleData["Print"], FontSize → 18],
    Cell[StyleData["Print", "Presentation"],
     CellMargins → {{120, 10}, {7, 12}},
     FontSize → 18], Cell[StyleData[
      "Print", "SlideShow"]]}, Closed]],
 Cell[StyleData["MSG"], FontSize → 16]},
Closed]], Cell[CellGroupData[
{Cell["Styles of Links", "Section"],
 Cell[StyleData["Link", "SlideShow"],
  FontSize → 21],
 Cell[StyleData["SearchLink",
   StyleDefinitions → StyleData["Link"]]],
```

```
 Cell[StyleData["FunctionLink"]]},
Closed]], Cell[CellGroupData[
{Cell["Special Styles", "Section"],
 Cell[CellGroupData[
   {Cell[StyleData["Picture"],
     CellMargins → {{100, 100}, {20, 20}},
     ShowStringCharacters → False,
     MenuSortingValue → 5000, Magnification →
      1], Cell[StyleData["Picture",
      "Presentation"], CellMargins →
      {{100, 100}, {20, 20}}, Magnification →
      1], Cell[StyleData["Picture",
      "SlideShow"]]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "PictureCaption", StyleDefinitions →
       StyleData["Text"]], CellDingbat → "▲",
     CellMargins → {{114, 100}, {15, 4}},
     ShowStringCharacters → False,
     MenuSortingValue → 5100, FontSize →
      14, FontColor → GrayLevel[0.3`]],
    Cell[StyleData["PictureCaption",
      "Presentation"], CellMargins →
      {{114, 100}, {20, 4}}, FontSize → 16],
    Cell[StyleData["PictureCaption",
      "SlideShow"], CellMargins → {{114, 100},
       {20, 4}}, FontSize → 16]}, Closed]],
 Cell[CellGroupData[{Cell[StyleData[
      "OverviewTitle"], CellFrame → None,
     CellMargins → {{100, 100}, {20, 60}},
     FontFamily → "Calibri", FontSize →
      24, FontWeight → "Bold", FontColor →
      GrayLevel[0.4`]], Cell[StyleData[
      "OverviewTitle", "Presentation"],
     CellFrame → None, CellMargins →
```

```
      {{100, 100}, {20, 60}}, FontFamily →
       "Calibri", FontSize → 24, FontWeight →
       "Bold", FontColor → GrayLevel[0.4`]],
     Cell[StyleData["OverviewTitle",
       "SlideShow"], CellFrame → None,
      CellMargins → {{100, 100}, {20, 60}},
      FontFamily → "Calibri", FontSize →
       24, FontWeight → "Bold", FontColor →
       GrayLevel[0.4`]]}, Closed]],
   Cell[CellGroupData[{Cell[StyleData[
       "OverviewSection"], CellFrame → None,
      CellMargins → {{120, 100}, {10, 10}},
      FontFamily → "Georgia", FontSize → 22,
      FontWeight → Plain, FontColor → RGBColor[
        0.811765`, 0.117647`, 0.145098`],
      ReturnCreatesNewCell → True],
     Cell[StyleData["OverviewSection",
       "Presentation"], CellFrame → None,
      CellMargins → {{120, 100}, {10, 10}},
      FontFamily → "Georgia", FontSize → 22,
      FontWeight → Plain, FontColor → RGBColor[
        0.811765`, 0.117647`, 0.145098`],
      ReturnCreatesNewCell → True],
     Cell[StyleData["OverviewSection",
       "SlideShow"], CellFrame → None,
      CellMargins → {{120, 100}, {10, 10}},
      FontFamily → "Georgia", FontSize → 22,
      FontWeight → Plain, FontColor → RGBColor[
        0.811765`, 0.117647`, 0.145098`],
      ReturnCreatesNewCell → True]}, Open]],
   Cell[CellGroupData[{Cell[StyleData[
       "ChapterTagline"], CellMargins →
       {{60, 90}, {15, 0}}, MenuSortingValue →
       2420, FontFamily → "Georgia", FontSize →
```

```
      20, FontWeight → "Plain", FontSlant →
      "Italic", FontColor → GrayLevel[0.3`]],
    Cell[StyleData["ChapterTagline",
      "Presentation"], CellMargins →
      {{90, 90}, {15, 0}}, FontSize → 24],
    Cell[StyleData["ChapterTagline",
      "SlideShow"]]}, Closed]],
Cell[StyleData["PlotLabel"],
 FontFamily → "Helvetica",
 FontSize → 14],
Cell[StyleData["NonPrintButton"],
 CellMargins → {{128, Inherited}, {16, 0}},
 FontFamily → "Helvetica",
 FontSize → 12,
 FontColor → GrayLevel[1],
 ButtonBoxOptions →
  {Background → RGBColor[0.811765`, 0.117647`,
     0.145098`], ButtonMargins → 4}],
Cell[CellGroupData[{Cell[StyleData[
     "Program"]], Cell[StyleData["Program",
     "SlideShow"], CellMargins → {{116, 48},
      {16, 16}}, FontSize → 18]}, Closed]],
Cell[CellGroupData[{Cell[StyleData[
     "Editorial"], ShowCellBracket →
     Automatic, CellMargins → {{24, 10},
      {6, 6}}, CellBracketOptions → {"Color" →
       RGBColor[0.786999`, 0.0386053`,
        0.0571756`]}, CellFrameLabels →
     {{Cell[TextData[{"Edit:"}], FontFamily →
         "Helvetica", FontSize → 16,
        FontWeight → "Bold", CellBaseline →
         Baseline], None}, {None, None}},
    DefaultNewInlineCellStyle → "None",
    Hyphenation → True, LineSpacing → {1, 3},
```

```
            LanguageCategory → "NaturalLanguage",
            FontFamily → "Helvetica", FontSize →
             19, FontColor → RGBColor[1, 0, 0]],
           Cell[StyleData["Editorial",
            "SlideShow"], CellMargins →
            {{Inherited, Inherited}, {0, 0}},
           CellElementSpacings → {"CellMinHeight" →
              1, "ClosedCellHeight" → 0}, CellOpen →
            False]}, Closed]]}, Open]]}, Open]]},
Visible → False, FrontEndVersion →
 "8.0 for Microsoft Windows
   (64-bit) (February 23, 2011)",
StyleDefinitions → "StylesheetFormatting.nb"]];
```