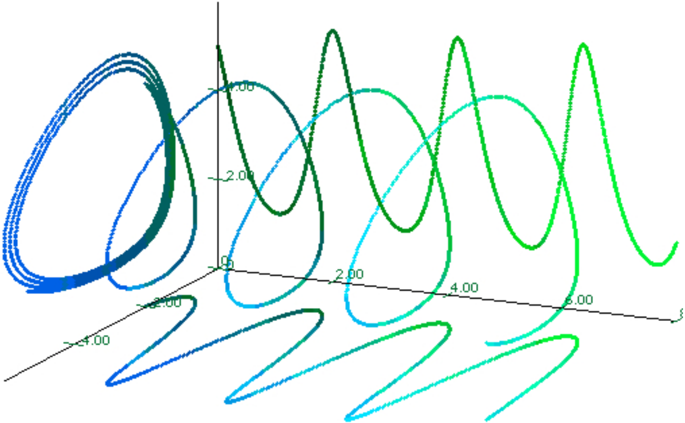
Introduction to Maple
MAT 3535

Wm C Bauldry

BauldryWC

Spring Semester, 2006



Maple plot of a Lotka–Volterra system in 3-space

Table of Contents

Day 0

Background

1. Course description. (*The Plan*)
2. What is a *computer algebra system*?

History

1. Why use computer algebra systems. (Maple poster)
2. A brief history of computer algebra.

Arithmetic Operators

Maple has the standard arithmetic operators using normal syntax:

Binary: $+, -, *, /, \hat{}, \text{ mod}$

Unary: $+, -, !$

Relation $<, \leq, =, >, \geq, <> (\neq)$

(Maple has the standard set and logic operators. See `?operators`.)

Arithmetic can be calculated in:

1. basic fields $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, and $\mathbb{C}$.
2. algebraic extensions, $\mathbb{Q}[\sqrt{2}]$, $\mathbb{Z}[i]$, etc.
3. finite fields, $\mathbb{Z}_p$, Galois fields GF(p^k), etc.

Statements are terminated with ";" or with ":" (no output). Maple 10 does not require terminators.

Data Structures

- Integers — represented as $n = \pm \sum_{k=0}^{n} i_k \left(10^4\right)^k$

intpos	i_0	i_1	...	i_n

intneg	i_0	i_1	...	i_n

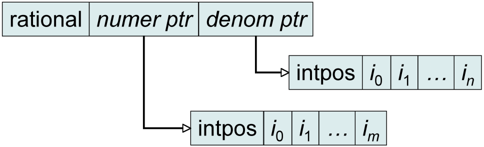
Data Structures

- Integers — represented as $n = \pm \sum_{k=0}^{n} i_k \left(10^4\right)^k$

intpos	i_0	i_1	$\ldots$	i_n

intneg	i_0	i_1	$\ldots$	i_n

- Rational Number — represented as $r = (\pm p)/q$

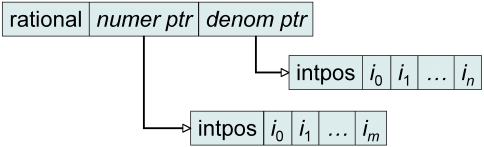


- Using pointers allows atoms to be stored once and referenced by many expressions.

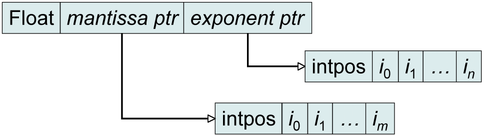
Data Structures

- Integers — represented as $n = \pm \sum_{k=0}^{n} i_k \left(10^4\right)^k$

- Rational Number — represented as $r = (\pm p)/q$



- Floating Point Number — represented as $x = \pm m \times 10^{\pm e}$



- Using pointers allows atoms to be stored once and referenced by many expressions.

Algebraic and Complex Numbers

Maple can work with algebraic numbers; try:

- `simplify( (1+sqrt(2))*(2-sqrt(2)) )`
- `radsimp( (1+sqrt(2))/(2-sqrt(2)), 'ratdenom' )`
- `expand( (1-sqrt(3))^4 )`

Complex arithmetic is actually the standard for Maple; use i = `I`. Try:

- `simplify(I^I)`
- `expand((1+I)/(1-I))`
- `cosh(1.+I)`

Defined Constants and Functions

Many standard functions and constants are defined in Maple. (NB: Maple is case sensitive.)

- `Pi` $= \pi$, the constant, but `pi` is just the Greek letter
- Enter: `?initial names`
- Look at `Digits`

There is a long list of predefined functions containing both the usual trig and log/exponential as well as a large collection of "exotics."

- Enter `?inifcns` to see the whole list.
- See also `?LambertW` for an example of an exotic function.
- Investigate `evalf`, `ifactor`, `sqrt`, and `surd`.

Day 2: Variables & Names

- Names are strings that contain alphanumeric characters and underscores, begin with a letter or underscore, and have no more than $2^{28} - 8$ characters for 32 bit systems, $2^{35} - 17$ for 64 bit systems. (See `?keywords` for reserved names.) The data structure is:

Name	*assigned-expr ptr*	*attrib-expr ptr*	*chars*	*chars*	...

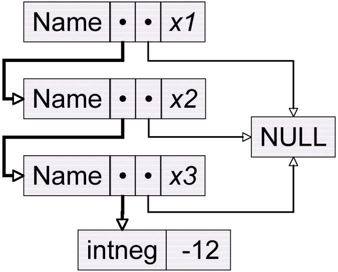
- Maple uses the standard mathematical assignment operator `:=` to define names. (No space is allowed between `:` and `=`.) E.g.,
  - `x := 1`
  - `y := x^2+2x+2`
  - `profit := revenue - cost`
- To "clear" a name
  - `unassign('x')`
  - `x := 'x'`

  Note the single quotes around `x` to prevent evaluation.

Name Evaluation

- Internally Maple uses *full evaluation*—follow the path to the end.

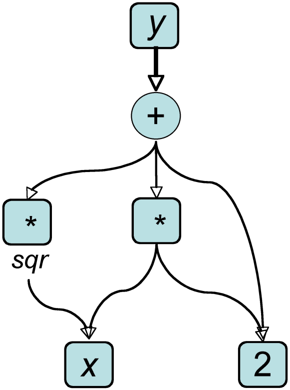
> *The structure at the right is created by:*
> ```
> > x1 := x2;
> > x2 := x3;
> > x3 := -12;
> ```
> *Try:*
> ```
> > x1;
> > eval(x1, 1);
> > eval(x1, 2);
> > eval(x1, 3);
> ```



1. Change `x3`, then check `x1` and `x2`.
2. Change `x2`, then check `x1` and `x3`.

Expression Data Structure

The statement `y:=x^2+2*x+2` generates a *directed acyclic graph*, abbreviated as DAG



Use `dismantle(y)` to see a text representation of this structure.

Substituting Values

- There are three methods for substituting values for variables (names). Define `y := 2*x+1`. To evaluate `y` when `x=1`, use:
  1. `> x:=1; y;` (Set `x=1`. Evaluate `y`.)
  2. `> subs(x=1, y);` (Replace each `x` with `1` in `y`.)
  3. `> eval(y, x=1);` (Evaluate `y` taking `x=1`)

  The first method sets `x` globally to `1`.
  The second uses pattern matching, replacing each string `x` in `y` with `1`.
  The last is an evaluation with `x` temporarily being `1`.

- `eval` has a number of variants: `Eval` (*inert*), `evala` (*algebraic*), `evalb` (*boolean*), `evalc` (*complex*), `evalf` (*floating point*), `evalhf` (*hardware float*), `evalm` (*matrix*), `evaln` (*name*), `evalr` (*range arithmetic*), and `evalrC` (*complex range arithmetic*).

Day 3: Polynomials, Rational Expressions, & Functions

- Polynomial expressions are automatically simplified over addition & subtraction, but not over multiplication & division.
- The order of the terms appears random; actually, it's based on the order of atoms in memory. Put a polynomial in standard order with `sort`.
- Use `expand`, `simplify`, and `collect` to manipulate polynomials. Coefficients can be extracted individually with `coeff` or all in a list with `coeffs`.
- For rational expressions, use `expand`, `simplify`, and `normal`.
- The `factor` function is applied to polynomials and the parts of rational expressions, functions, and lists.

Experiment with `randpoly` to make random polynomials and rational expressions to `expand`, `simplify`, `factor`, etc.

Factoring Polynomials

Maple uses sophisticated techniques to factor polynomials and expressions.

Experiment

Set `p := x^5+x^3-x^2-1`. *Enter:*

1. `factor(p)`
2. `factor(p, I)`
3. `factor(p, {I, sqrt(3)})`
4. `factor(p, real)`
5. `factor(p, complex)`

Describe your results.

Try factoring several rational expressions.

Rational Expression Cost

Rational expressions can be written in *normal*, *Horner*, aand *continued fraction* form.

Experiment (Calculating Complexity)

1. *Setup:*
   ```
   with(codegen, cost)
   n:=randpoly(x, degree=5, coeffs=rand(-2..2))
   d:=randpoly(x, degree=5, coeffs=rand(-2..2))
   ```
2. *Define the three rational functions forms:*
   ```
   r1:=n/d
   r2:=convert(r1, horner)
   r3:=convert(r1, confrac, x)
   ```
3. *Calculate the "cost" of each form with* `cost(r1)`*, etc.*

Repeat the experiment several times. Which form is the easiest to understand? Which is the easiest to compute?

Functions

There are three basic methods used to define a function in Maple.

Definition (Defining a Function in Maple)

Arrow	*name* := *args* -> *result*
Procedure	*name* :=`proc(`*args*`)` *statements* `end proc`
Expression	*name* :=`unapply(`*expression*, *args*`)`

Example (Define $f(x) = \sin(x^2)$.)

1. `f:=x -> sin(x^2)`
2. `f:=proc(x) sin(x^2) end proc`
3. `f:=unapply(sin(x^2), x)`

Query?

- What does `f(1):=2` do?
- Why doesn't `f(x):=sin(x^2)` work to define $f(x)$?

Recursive Functions

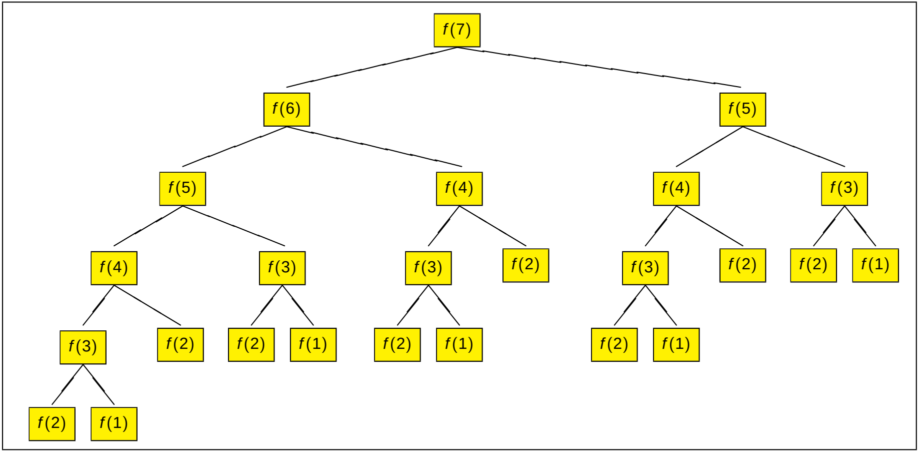
Example (Fibonacci Sequence)

Set $f(n) = f(n-1) + f(n-2)$ with $f(1) = f(2) = 1$ for $n \in \mathbb{Z}^+$.

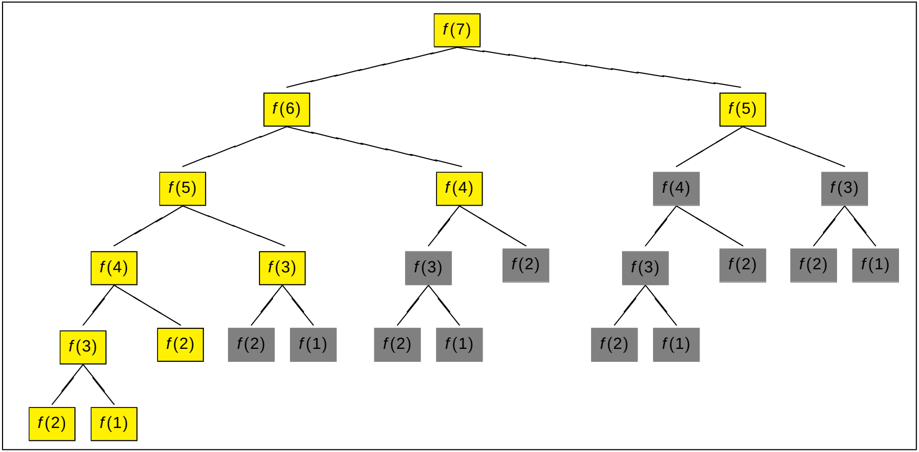
```
f:=proc(n)        {or we can use proc(n::posint) to ''type-check'' the arguments}
  if n<2 then 1
  elif n<3 then 1
  else f(n-1)+f(n-2) end if
  end proc
```

1. Execute
   ```
   T:=time(); f(N); (time()-T)*seconds;
   ```
   using $N = 20, 25$, and 30.
2. Change the function's first line to
   ```
   f:=proc(n) option remember;
   ```
   Redo 1. What happens? Why?

Recursive Fibonacci Tree



The calculation tree without "`option remember`"



The calculation tree with “`option remember`”

Maple has two main differentiators; use `diff` for expressions and `D` for functions. All symbolic derivatives are partial derivatives.

Definition (Differentiating an Expression)

Basic Form

$$\frac{\partial}{\partial var}\, expr = \mathtt{diff}(expr, var\,)$$

Extended Form

$$\frac{\partial^{(n_1+n_2+\dots)}}{\partial var_1^{n_1}\partial var_2^{n_2}\dots}\, expr = \mathtt{diff}(expr, var_1\$n_1, var_2\$n_2, \dots)$$

The inert form of `diff` is `Diff`. Try

```
Diff(sin(x), x$2006) = diff(sin(x), x$2006)
```

Derivatives

Definition (Differentiating a Function)

Basic Form

$$\frac{\partial}{\partial var} f = \texttt{D}(f)$$

Extended Forms

$$\frac{\partial^n}{\partial var^n} f = \texttt{(D@@n)}(f)$$

$$\frac{\partial^{(n_1+n_2+\dots)}}{\partial var_1^{n_1}\partial var_2^{n_2}\dots} = \texttt{D[1\$}n_1\texttt{, 2\$}n_2\texttt{, ...](f)}$$

There are two important equivalences relating `diff` and `D`:

- `diff(f(x),x)` $\equiv$ `D(f)(x)`
- `D(f)` $\equiv$ `unapply(diff(f(x),x),x)`

Subexpression Explosion

Subexpression Explosion is a major problem for computer algebra designers. The problem often occurs when differentiating.

Experiment (Exploding Taylor Coefficients)

Try to compute the Maclaurin series coefficients $\frac{d^n}{dx^n}\tan(x)$. *Try different values of* N *in:*

```
> N :=  ?
> d[1] := diff(tan(x), x)
> for i from 2 to N do
>   d[i] := diff(d[i-1], x)
>   end do
```

What do you observe about the number of terms and the size of the coefficients as N increases?

The main Maple function for indefinite integration is `integrate`, usually called via `int`.

Definition (Integrating an Expression)

Indefinite Integral

$$\int f(x)\,dx = \texttt{int(f(x), x)}$$

Definite Integral

$$\int_a^b f(x)\,dx = \texttt{int(f(x), x=a..b)}$$

The definite integral attempts to use the Fundamental Theorem of Calculus when possible. Note, there is an inert form: `Int`.

Computer Indefinite Integration

To see the techniques Maple uses for integrating a function, increase the value of `infolevel[int]`.[1]

Experiment (Indefinite Integration Methods)

Enter:

```
> infolevel[int] := 5
> y := 1/(x^5+2*x^2+1)
> Int(y, x) = int(y, x)
```

Try integrating several different functions.

The names *Liouville, Hermite, Horowitz, Rothstein, Trager,* and *Risch* figure prominently. See Bronstein's *Symbolic Integration I* (2nd ed, Springer-Verlag, 2005: ISBN: 3-540-21493-3)

[1] In Maple 10, either use 'Worksheet Mode' or 'Expand Document Blocks' to see the workings of `int`.

Numeric Integration

There are two ways to ask Maple for a numeric integral.

Definition (Numerically Integrating a Function)

Fundamental Theorem — if possible

$$\int_a^b f(x)\,dx = \texttt{int(f(x), x=a..b)}$$

Numeric Quadrature

$$\int_a^b f(x)\,dx \approx \texttt{evalf(Int(f(x), x=a..b))}$$

See `?evalf,int` for the numerical methods Maple implements.

Dangers in “Freshman Integration”

Experiment

1. *Define* $y := \dfrac{\sqrt{2}}{1 + \sin^2(t)}$.
2. *Define* $Y := \int y\, dt$.
3. *Graph y and Y together with*

```
plot([y, Y], t=0..2*Pi, discont=true)
```

Question: Using the Fundamental Theorem with Y, *what is* $\int_0^{2\pi} y\, dt$?

4. *Use Maple to do the calculation in two ways:*

```
> int(y, t=0..2*Pi)
> evalf(Int(y, t=0..2*Pi))
```

Question: What happened?

Day 5: Summation, Series, & Limits

Maple has several commands that create sequences and series.

Definition

Sequence $\{expr\}_{ndx_1}^{ndx_n}$ is

`seq(`$expr, ndx = ndx_1..ndx_n$`)`

`seq(`$expr, ndx$ `in` $< set \mid list >$`)`

Sum $\sum_{ndx_1}^{ndx_n} expr$ is

`sum(`$expr, ndx = ndx_1..ndx_n$`)`

`sum(`$expr, ndx$ `in` $< set \mid list >$`)`

Enter in Maple:

1. $\{1, 3, 5, 7, \ldots, 99\}$, $\{1, 0, -1, 0, 1, 0, -1, 0, 1, 0\}$, $\{2, 4, 6, \ldots\}$
2. The sum of odds from 1 to 99, $\displaystyle\sum_{k=1}^{1000} \frac{1}{k}$, $\displaystyle\sum_{k=1}^{\infty} \frac{1}{k^N}$ for $N = 1, 2, 3, \ldots$

Side Bar: The Ratio Test

Theorem (The Ratio Test)

Let $\sum_{n=0}^{\infty} a_n$ *be a series of positive terms. Define* $\rho = \lim_{n\to\infty} \frac{a_{n+1}}{a_n}$*. Then,*

- *if* $\rho < 1$*, we have* $\sum a_n$ *converges.*
- *if* $\rho = 1$*, the test fails.*
- *if* $\rho > 1$*, we have* $\sum a_n$ *diverges.*

Example (Does $\sum_{n=1}^{\infty} \frac{n!}{n^n}$ converge or diverge?)

```
> a := n -> n!/n^n
> r := n -> a(n+1)/a(n)
> rho := limit(r(n), n, infinity)
> S := sum(a(n), n=1..infinity)
```

What about: $\sum \frac{1}{2n+1}$, $\sum \frac{1}{n\ln(n)}$, $\sum \frac{n}{\sqrt{2n+1}}$, $\sum \frac{\alpha^n}{n!}$, $\sum \frac{n^n}{n!}$

Limit

Maple's `limit` generally uses a series expansion to calculate a value.

Experiment

```
> Limit(sin(x)/x, x=0)=limit(sin(x)/x, x=0)
> Limit(sin(1/x), x=0)=limit(sin(1/x), x=0)
> Limit(ln(x),x=infinity)=limit(ln(x),x=infinity)
> Limit(signum(x),x=0,left)
    =limit(signum(x),x=0,left)
> Limit(signum(x),x=0,right)
    =limit(signum(x),x=0,right)
> Limit(sin(1/x), x=0)=limit(sin(1/x), x=0)
```

Explain the difference in answers from:

```
> limit(tan(n*Pi*x),x=1)
> limit(tan(n*Pi*x),x=1) assuming n::posint
```

Indefinite Sums

There is a theory of indefinite sums, or anti-differences, paralleling indefinite integrals, or anti-derivatives.

Experiment

Define

```
L := N -> [seq(Sum(n^k,n)=sum(n^k,n), k=1..N)].
```

Then display results as a column vector with

```
Vector(L(8))
```

and

```
Vector(map(factor, L(8))).
```

What do you observe? Compare with $\int n^k dn$.

As a starting point, look up Finite Difference Calculus.

Reciprocal Sums

Experiment

Define the falling factorial function:

```
FF := n -> product(k-j, j=0..n).
```

Now create a function that sums the reciprocals of the falling factorials:

```
S:=j ->
  Sum(1/FF(j), k=j+1..n) = sum(1/FF(j), k=j+1..n)
```

Consider the results in

```
Vector([seq(S(i), i=0..5)])
```

Questions:

1. *What is the pattern?*
2. *What is a formula for the constant term?*
3. *Give a formula* $S_n(j)$ *for* $\displaystyle\sum_{k=j+1}^{n} \frac{1}{k(k-1)\ldots(k-j)}$. *Then* $S_\infty(j) = ?$

Maple has a large variety of methods to produce plots of functions, expressions, and data. Maple can also make animated graphs. The images can be exported in many different formats. Maple 9 expanded plotting capabilities making all graphs into "smartplots."

Definition

2-d `plot(` *expr*, $<$*var*=*range*, *options*$>$`)`

`plot(` *fcn*, $<$*range*, *options*$>$`)`

`plot([[`x_1,y_1`],[`x_2,y_2`],...],` $<$*range*, *options*$>$`)`

3-d `plot3d(` *expr*, *var*=$range_1$, *var*=$range_2$, $<$*options*$>$`)`

`plot3d(` *fcn*, $range_1$, $range_2$, $<$*options*$>$`)`

`plot3d([[`x_1,y_1,z_1`],[`x_2,y_2,z_2`],...]` *var*=$range_1$, *var*=$range_2$, $<$*options*$>$`)`

See `?plot,options` and `?plot3d,options`.

Examples

Example

Try the the following graphs; investigate changing options.

```
> plot(sin, -2*Pi..2*Pi)
> plot({sin(x),cos(x)}, x=-2*Pi..2*Pi,
    color=[red,blue])
> plot(tan(x), x=-2*Pi..2*Pi)
> plot(tan(x), x=-2*Pi..2*Pi, -10..10,
    discont=true)
> plot3d(x^2-y^3, x=-2..2, y=-1..1, axes=frame,
    style=patchnogrid)
> plot3d(sin(x*y/3), x=-Pi..Pi,
    y=-sqrt(Pi^2-x^2)..sqrt(Pi^2-x^2))
```

Packages, I

There are two packages of functions for plotting:`plottools`.

Definition (The `plots` package)

```
> with(plots)
```

[*Interactive, animate, animate3d, animatecurve, arrow, changecoords, complexplot, complexplot3d, conformal, conformal3d, contourplot, contourplot3d, coordplot, coordplot3d, cylinderplot, densityplot, display, display3d, fieldplot, fieldplot3d, gradplot, gradplot3d, graphplot3d, implicitplot, implicitplot3d, inequal, interactive, interactiveparams, listcontplot, listcontplot3d, listdensityplot, listplot, listplot3d, loglogplot, logplot, matrixplot, multiple, odeplot, pareto, plotcompare, pointplot, pointplot3d, polarplot, polygonplot, polygonplot3d, polyhedra_supported, polyhedraplot, replot, rootlocus, semilogplot, setoptions, setoptions3d, spacecurve, sparsematrixplot, sphereplot, surfdata, textplot, textplot3d, tubeplot*]

Packages, II

Definition (The `plots` package)

```
> with(plottools)
```

[*arc, arrow, circle, cone, cuboid, curve, cutin, cutout, cylinder, disk, dodecahedron, ellipse, ellipticArc, hemisphere, hexahedron, homothety, hyperbola, icosahedron, line, octahedron, parallelepiped, pieslice, point, polygon, project, rectangle, reflect, rotate, scale, semitorus, sphere, stellate, tetrahedron, torus, transform, translate, vrml*]

Example (A composite graph of $f(x, y) = x^2 - y^3$.)

```
> p[f] := plot3d(f, -1..1, -1..1, style=
   patchnogrid, shading=zhue):
> p[c] := contourplot3d(f, -1..1, -1..1, style=
   contour, contours=14, thickness=2, color=black):
> display([p[f], p[c]], axes=boxed)
```

Animations

There are two methods for making animated graphs: via a parameter and from a sequence of images.

Example

```
> animate(plot, [k*sin((k+1)*Pi*x),x=-Pi..Pi], k=1..10)

> the_frames := [seq(plot(x^k, x=-1..1), k=1..10)]:
  display(the_frames, insequence=true)

> f := (x,y,k) -> -sin(k/30*Pi)
    * cos(x^2+y^2)*exp(-(x^2+y^2)/3)
  rngs := x=-Pi..Pi, y=-sqrt(Pi^2-x^2)..sqrt(Pi^2-x^2)
  opts := style=patchcontour, orientation=[40,60]
  animate(plot3d, [f(x,y,k), rngs], k=1..60, frames=90,
    opts)
```

Look at the Maple code for generating the Lotka-Volterra graph (on slide 2). Could this plot be animated?

Day 7: Solving Algebraic & Differential Equations

Solving equations is an active area of CAS research. Very sophisticated techniques involving algebraic ideals and extensions have been implemented. Maple's main solving functions include:

Definition (Solve and its Variants)

- `solve`: main solving function: `solve(`*eqns, vars*`)`
- `dsolve`: differential equation solver
- `fsolve`: numeric solver
- `isolve`: solve for integer solutions
- `LinearSolve`: linear matrix/system solver (in `LinearAlgebra`)
- `msolve`: solve equations in $\mathbb{Z}_m$
- `pdsolve`: solve partial differential equations
- `rsolve`: recurrence equation solver

The Quadratic Formula

Experiment

Set `QuadraticEquation` *to* $ax^2 + bx + c = 0$ *and then solve.*

```
> QuadraticEquation := a*x^2+b*x+c=0
> solve(QuadraticEquation, x)
```

Compare the solution above to

```
> solve(QuadraticEquation)
```

What is the difference?

Do the same with the depressed cubic equation: $x^3 + px + q = 0$.

Along with normal solutions, we see many *special functions* results.

Example

```
> solve(x*exp(x)=a, x)
```

$$\mathrm{LambertW}(a)$$

```
> S := x -> int(sin(Pi/2*t^2), t=0..x):
```
(Graph S)

```
> s1 := solve(S(x) = 7/10);
  evalf(s1);
  s2 := fsolve(S(x) = 7/10);
```

$$s1 := \mathrm{RootOf}(10\,\mathrm{FresnelS}(\_Z) - 7)$$

$$-.5718592176 - .8807572177\,\mathrm{I}$$

$$s2 := 1.333735601$$

Differential Equations

Many differential equations can be solved.

Example

```
> eq := (D@@2)(y)(x) + 2*D(y)(x) + y(x) = 0
> dsolve(eq, y(x))
```

$$y(x) = \_C1 e^{-x} + \_C2\, e^{-x} x$$

```
> dsolve({eq, y(0)=0, D(y)(0)=2}, y(x))
```

$$y(x) = 2\, e^{-x} x$$

```
> sys := {D(y)(x)=y1(x), D(y1)(x)=-2*y1(x)-y(x)}
> dsolve(sys, {y(x), y1(x)})
```

$$\{y(x) = e^{-x}(\_C1 + \_C2 x),\ y1(x) = -e^{-x}(\_C1 + \_C2 x - \_C2)\}$$

Also investigate `?dsolve,numeric` and `?dsolve,series`.

Recurrence Equations

Discrete differences lead to *recurrence equations*.

Example

```
> eq := phi(n) = phi(n-1)+phi(n-2)
> inits := phi(0)=1, phi(1)=1
> rsolve({eq, inits}, {phi})
```

$$\left\{\phi(n) = \left(-\frac{1}{10}\sqrt{5}+\frac{1}{2}\right)\left(-\frac{1}{2}\sqrt{5}+\frac{1}{2}\right)^n + \left(\frac{1}{2}+\frac{1}{10}\sqrt{5}\right)\left(\frac{1}{2}+\frac{1}{2}\sqrt{5}\right)^n\right\}$$

```
> eqL := u(n) = u(n-1) + 1/10*u(n-1)*(10-u(n-1)):
> U := rsolve({eqL, u(0)=0.1}, u(n), 'makeproc'):
> pts := [seq([k, U(k)], k=0..15)]
> plot(pts)
```

Day 8: The *Student[LinearAlgebra]* Package

The `Student[LinearAlgebra]` package is designed to handle computations and assist learning elementary linear algebra concepts. This package has 77 functions, 9 *interactive tutors*, and a “matrix builder.” Users may also choose to use the Matrix palette to enter matrices. Load the package via `with(Student[LinearAlgebra])`.

Example (The Tutors)

The *tutors* are Maplet-based introductions. Try each one:

EigenPlotTutor	EigenvaluesTutor
EigenvectorsTutor	GaussianEliminationTutor
GaussJordanEliminationTutor	InverseTutor
LinearSolveTutor	LinearSystemPlotTutor
LinearTransformPlotTutor	

Matrix and Vector Operations

Definition

Vectors are entered as:

```
   column:  Vector([1,2,3])        or  <1,2,3>
      row:  Vector[row]([1,2,3])   or  <1|2|3>
   by fcn:  Vector(3, i -> 2*i)
```

Matrices are entered as:

```
   by col:  Matrix([<1,3>,<2,4>])  or  <<1,3>|<2,3>>
   by row:  Matrix([[1,2],[3,4]])  or  <<1|2>,<3|4>>
   by fcn:  Matrix(2,2, (i,j) -> i+j-1)
```

- Matrix and vector addition and multiplication uses: + and .
- Vector cross product uses: &x
- Matrix power is: ^
- Adding a scalar to a matrix uses an implicit identity

Examples, I

Experiment

Load the `Student[LinearAlgebra]` *package. Try:*

```
> M := RandomMatrix(3,4, generator=-3..3)
> GenerateEquations(M, [x[1],x[2],x[3]])
> GaussianElimination(M)
> ReducedRowEchelonForm(M)
```

Repeat this several times. What do you observe?

Experiment

Investigate the visualization commands:

ApplyLinearTransformPlot CrossProductPlot EigenPlot
LeastSquaresPlot LinearSystemPlot LinearTransformPlot
PlanePlot ProjectionPlot VectorSumPlot

Examples, II

Load the `Student[LinearAlgebra]` package.

Experiment

Try:

```
> M := RandomMatrix(3,3, generator=-1..1)
> p := CharacteristicPolynomial(M, lambda)
> pM := eval(p, lambda=M)
> value(pM)
> Eigenvectors(M, output=list)
> RowSpace(M)
> ColumnSpace(M)
```

Repeat this several times. What do you observe?

Look at the full `LinearAlgebra` package.

Projects, I

Project (Limits)

Discuss limits in terms of the ϵ-δ definition. Use Maple to calculate limits and graph examples illustrating ϵ-δ arguments.

Project (Continuity)

Discuss continuity in terms of the limit definition. Use Maple to calculate limits and graph examples.

Project (Differentiation)

Discuss differentiation in terms of the limit definition. Use Maple to calculate limits and graph examples.

Projects, II

Project (Integration: Riemann Sums)

Discuss integration in terms of Riemann sums. Use Maple to calculate Riemann sums and graph examples.

Project (Integration: Liouville)

Discuss integration techniques using Liouville's "Integration in finite terms" model. Use Maple to calculate integrals via Liouville's method and integrate examples with `infolevel` *set to* 3.

Project (Convergence Tests)

Define the standard convergence tests in Maple. Show graphs of converging and diverging series.

Projects, III

Project (Tangent Line Animation)

Create a Maple procedure that inputs a function and draws a tangent lines to successive points in the domain.

Project (Maple Cobweb Animation)

Create a Maple procedure that inputs a function and draws a "cobweb diagram" in stages.

Project (Newton's Method Animation)

Create a Maple procedure that inputs a function and a starting point, then animates Newton's method.

Projects, IV

Project (Systems of Recurrence Equations)

Investigate the system of recurrence equations

$$\left\{ \begin{array}{lcl} x_n & = & x_{n-1} + (1.5x_{n-1} - x_{n-1}y_{n-1}) \cdot \Delta t \\ y_n & = & y_{n-1} + (-3y_{n-1} + x_{n-1}y_{n-1}) \cdot \Delta t \end{array} \right\}$$

letting $N = 400$ *&* $\Delta t = 0.02$ *for several different choices of* x_0 *and* y_0 *between* 4 *and* 10 *(include the point* $(x_0, y_0) = (3.0, 1.5)$*).*
Create plots with:

1. `pts_nx := [seq([k,x(k)], k=0..N)]:`
2. `pts_ny := [seq([k,y(k)], k=0..N)]:`
3. `pts_xy := [seq([x(k),y(k)], k=0..N)]:`
4. `pts_nxy := [seq([n,x(k),y(k)], k=0..N)]:`

Describe your results.

Do you recognize this well-known system?

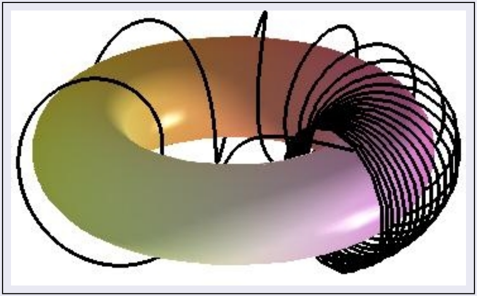
Projects, V

Project (Coiled Helix)

Produce a plot like the one below of the helix

$$f(t) = \begin{bmatrix} (3 + 1.5\cos(15/(1.1 - t))) \times \sin(2\pi t) \\ (3 + 1.5\cos(15/(1.1 - t))) \times \cos(2\pi t) \\ 1.5\sin(15/(1.1 - t)) \end{bmatrix}, t = 0..1$$

coiled around a torus. Discuss coordinate transformations.



Projects, VI

Project (Systems of Equations)

Discuss Reduced Row Echelon Form. Use this technique to solve systems of equations showing examples of over- & under-determined and consistent & inconsistent systems.

Project (Determinants)

Using `RandomMatrix`, generate enough matrices to estimate the probability that the determinant of a random 5×5 matrix M with entries $-10 < m_{ij} < +10$ is nonzero; i.e. that M is nonsingular.

Project (Eigenspaces)

Define and calculate eigenvalues and eigenvectors. Show how to find an eigenbasis. Explain the geometric significance of eigenvectors.