3. Demonstrations of Using *Maple* in Calculus and Differential Equations

In this second introductory section we will give demonstrations of how *Maple* can be used in calculus and differential equations. Later, as you work through some of the lab sections, it may be helpful to return to this section to see how some of the code in *Maple* is actually used.

We will begin with some calculus operations that include algebraic manipulations.

1. Calculus Examples

In this section we will see how to differentiate, integrate, take limits, and manipulate algebraic expressions with *Maple*.
Consider the following sequence of operations relating to the function $x(t) = \mathbf{e}^{(-\alpha t)} \cos(\omega t + \theta)$.
What information is obtained in each step?

```
>
> restart;
> x := exp(-alpha*t)*cos(omega*t+theta);#Note that x is not declared to be a function but is an assigned variable. We will therefore refer to it at x below and not as x(t).
```

$$x := \mathbf{e}^{(-\alpha t)} \cos(\omega t + \theta)$$

```
> dx := diff( x, t );
```

$$dx := -\alpha\, \mathbf{e}^{(-\alpha t)} \cos(\omega t + \theta) - \mathbf{e}^{(-\alpha t)} \sin(\omega t + \theta)\, \omega$$

```
> dx2 := diff( x, t$2 );
```

$$dx2 := \alpha^2\, \mathbf{e}^{(-\alpha t)} \cos(\omega t + \theta) + 2\, \alpha\, \mathbf{e}^{(-\alpha t)} \sin(\omega t + \theta)\, \omega - \mathbf{e}^{(-\alpha t)} \cos(\omega t + \theta)\, \omega^2$$

```
> factor( dx );
```

$$-\mathbf{e}^{(-\alpha t)} (\alpha \cos(\omega t + \theta) + \sin(\omega t + \theta)\, \omega)$$

```
> Crit_point := solve( dx=0, t );
```

$$Crit\_point := -\frac{\theta + \arctan\left(\frac{\alpha}{\omega}\right)}{\omega}$$

```
> subs( t=Crit_point, dx2 );
```

$$\alpha^2\, \mathbf{e}^{\left(\frac{\alpha \left(\theta + \arctan\left(\frac{\alpha}{\omega}\right)\right)}{\omega}\right)} \cos\left(-\arctan\left(\frac{\alpha}{\omega}\right)\right) + 2\, \alpha\, \mathbf{e}^{\left(\frac{\alpha \left(\theta + \arctan\left(\frac{\alpha}{\omega}\right)\right)}{\omega}\right)} \sin\left(-\arctan\left(\frac{\alpha}{\omega}\right)\right) \omega - \mathbf{e}^{\left(\frac{\alpha \left(\theta + \arctan\left(\frac{\alpha}{\omega}\right)\right)}{\omega}\right)} \cos\left(-\arctan\left(\frac{\alpha}{\omega}\right)\right) \omega^2$$

```
> dx := simplify( %, trig );
```

$$dx := -\frac{\alpha^2 \mathbf{e}^{\left(\frac{\alpha\left(\theta + \arctan\left(\frac{\alpha}{\omega}\right)\right)}{\omega}\right)}}{\sqrt{\frac{\alpha^2}{\omega^2}+1}} - \frac{\mathbf{e}^{\left(\frac{\alpha\left(\theta + \arctan\left(\frac{\alpha}{\omega}\right)\right)}{\omega}\right)}\omega^2}{\sqrt{\frac{\alpha^2}{\omega^2}+1}}$$

```
>
```

Since the derivative will be negative (for real values of omega, alpha, and theta), this critical point must be a (local) maximum. The value of the function at this maximum is

```
> X[max] := simplify( subs( t=Crit_point, x ), trig );#Note the effect
  of [max]. It becomes a subscipt.
```

$$X_{max} := \frac{\mathbf{e}^{\left(\frac{\alpha\left(\theta + \arctan\left(\frac{\alpha}{\omega}\right)\right)}{\omega}\right)}}{\sqrt{\frac{\alpha^2+\omega^2}{\omega^2}}}$$

```
>
```

Can the same information be reached using the Second Derivative Test? (What is the second derivative of x?)

Indefinite and definite integration is obtained in a natural way.

```
> Int( x, t );
```

$$\int \mathbf{e}^{(-\alpha\, t)} \cos(\omega\, t + \theta)\, dt$$

```
> value( %  );
```

$$-\frac{\alpha\, \mathbf{e}^{(-\alpha\, t)} \cos(\omega\, t+\theta)}{\alpha^2+\omega^2} + \frac{\omega\, \mathbf{e}^{(-\alpha\, t)} \sin(\omega\, t+\theta)}{\alpha^2+\omega^2}$$

```
> simplify( % );
```

$$-\frac{\mathbf{e}^{(-\alpha\, t)} (\alpha \cos(\omega\, t+\theta) - \sin(\omega\, t+\theta)\, \omega)}{\alpha^2+\omega^2}$$

```
>
```

Note that the constant of integration is not included in these results.

Definite integrals are very similar. Here's an example taken from the CRC Tables of Integrals.

```
> Int( u/sin(u), u=0..Pi/2 ):#The colon means that the result is not
  displayed.
> % = value( % );
```

$$\int_0^{1/2\,\pi} \frac{u}{\sin(u)}\, du = 2\, Catalan$$

3
the) value of this integral is

```
> evalf( rhs(%) );
```

$$1.831931188$$

```
>
```

For further illustrations, consider the following two improper integrals.

```
> int( exp(-u^2), u=0..infinity );
```

$$\frac{1}{2}\sqrt{\pi}$$

```
> int( exp(-a*u^2), u=0..infinity );
Definite integration: Can't determine if the integral is convergent.
Need to know the sign of --> a
Will now try indefinite integration and then take limits.
```

$$\lim_{u\to\infty} \frac{1}{2}\frac{\sqrt{\pi}\,\mathrm{erf}(\sqrt{a}\,u)}{\sqrt{a}}$$

```
>
```

Note that *Maple* is unable to evaluate this limit until something is known about the parameter a.

```
> assume( a>0 );
  about( a );
Originally a, renamed a~:
  is assumed to be: RealRange(Open(0),infinity)
```

```
> int( exp(-a*u^2), u=0..infinity );
```

$$\frac{1}{2}\frac{\sqrt{\pi}}{\sqrt{a\sim}}$$

```
>
```

The following assignment is used to reset the name a to its unassigned state.

```
> a := 'a';
  about( a );
```

$$a := a$$

```
a:
  nothing known about this object
```

```
>
```

Here is a simple limit example.

```
> L := Limit( tan(theta), theta=Pi/2 );
```

$$L := \lim_{\theta\to(1/2\,\pi)} \tan(\theta)$$

```
> value( L );
```

$$\mathit{undefined}$$

```
> limit( tan(theta), theta=Pi/2, right );
```

$$-\infty$$

```
>
```

3

```
> restart;
```

Consider the proper rational function:

```
> f := (x^2-26*x-47)/(x^5+5*x^3+3*x^4+11*x^2-20);
```

$$f := \frac{x^2 - 26\,x - 47}{x^5 + 5\,x^3 + 3\,x^4 + 11\,x^2 - 20}$$

The antiderivative of f can be obtained by a variety of different means. Let's examine a few and compare the information obtained in each case.

```
> F := Int( f, x );
```

$$F := \int \frac{x^2 - 26\,x - 47}{x^5 + 5\,x^3 + 3\,x^4 + 11\,x^2 - 20}\,dx$$

The simplest, but least instructive, approach is to simply let *Maple* do all the work.
Note that the constant of integration is not included in indefinite integrals.

```
> F  = value( F );
```

$$\int \frac{x^2 - 26\,x - 47}{x^5 + 5\,x^3 + 3\,x^4 + 11\,x^2 - 20}\,dx =$$

$$-\frac{4}{3}\ln(x-1) + \frac{1}{3}\frac{1}{x+2} + \frac{23}{27}\ln(x+2) + \frac{13}{54}\ln(x^2+5) + \frac{91}{135}\sqrt{5}\arctan\left(\frac{1}{5}x\sqrt{5}\right)$$

or alternatively,

```
> int( f, x );
```

$$-\frac{4}{3}\ln(x-1) + \frac{1}{3}\frac{1}{x+2} + \frac{23}{27}\ln(x+2) + \frac{13}{54}\ln(x^2+5) + \frac{91}{135}\sqrt{5}\arctan\left(\frac{1}{5}x\sqrt{5}\right)$$

```
>
```

To emphasize integration techniques while still avoiding algebraic complications, we can utilize *Maple* to determine the partial fraction expansion of the integrand.

```
> fpf := convert( f, parfrac, x );
```

$$fpf := -\frac{4}{3}\frac{1}{x-1} - \frac{1}{3}\frac{1}{(x+2)^2} + \frac{23}{27}\frac{1}{x+2} + \frac{13}{27}\frac{7+x}{x^2+5}$$

From this partial fraction decomposition it is now a relatively simple matter to compute the integral BY HAND. The result can be compared with the previous answer.

```
> int( fpf, x );
```

$$-\frac{4}{3}\ln(x-1) + \frac{1}{3}\frac{1}{x+2} + \frac{23}{27}\ln(x+2) + \frac{13}{54}\ln(x^2+5) + \frac{91}{135}\sqrt{5}\arctan\left(\frac{1}{5}x\sqrt{5}\right)$$

process of finding a partial fraction expansion without fear of unmanageable algebra.

The key here is to enter the correct form for the partial fraction decomposition. This form depends upon the irreducible factors in the denominator of f.

```
> factor( denom( f ) );
```

$$(x-1)(x^2+5)(x+2)^2$$

```
>
```

The appropriate general form for the decomposition will be

```
> FORM := a/(x-1) + b/(x+2) + c/(x+2)^2 + (d+e*x)/(x^2+5);
```

$$FORM := \frac{a}{x-1} + \frac{b}{x+2} + \frac{c}{(x+2)^2} + \frac{d+e\,x}{x^2+5}$$

```
>
```

The second step is to determine the (linear) equations that define the constants a, b, c, d, and e.

```
> eqn := f = FORM;
```

$$eqn := \frac{x^2-26\,x-47}{x^5+5\,x^3+3\,x^4+11\,x^2-20} = \frac{a}{x-1} + \frac{b}{x+2} + \frac{c}{(x+2)^2} + \frac{d+e\,x}{x^2+5}$$

```
> simplify( % );
```

$$\frac{x^2-26\,x-47}{x^5+5\,x^3+3\,x^4+11\,x^2-20} = (a\,x^4+4\,a\,x^3+9\,a\,x^2+20\,a\,x+20\,a+b\,x^4+b\,x^3+3\,b\,x^2+5\,b\,x - 10\,b+c\,x^3-c\,x^2+5\,c\,x-5\,c+d\,x^3+3\,d\,x^2-4\,d+e\,x^4+3\,e\,x^3-4\,e\,x) \Big/ ((x-1)(x^2+5)(x+2)^2)$$

```
> collect( %, x );
```

$$\frac{x^2-26\,x-47}{x^5+5\,x^3+3\,x^4+11\,x^2-20} = ((a+b+e)\,x^4+(4\,a+3\,e+c+d+b)\,x^3+(9\,a+3\,b-c+3\,d)\,x^2 + (-4\,e+5\,b+5\,c+20\,a)\,x-5\,c-10\,b-4\,d+20\,a) \Big/ ((x-1)(x^2+5)(x+2)^2)$$

```
>
```

This equation will be satisfied (for all x, except 1 and -2) for any solution to the following system of five equations with five unknowns.

```
> eq1 := a+b+e=0;
  eq2 := 4*a+b+c+d+3*e;
  eq3 := 9*a+3*b-c+3*d = 1;
  eq4 := -4*e+5*b+5*c+20*a=-26;
  eq5 := -5*c-10*b-4*d+20*a = -47;
```

$$eq1 := a+b+e=0$$

$$eq2 := 4\,a+3\,e+c+d+b$$

$$eq3 := 9\,a+3\,b-c+3\,d=1$$

$$eq4 := -4\,e+5\,b+5\,c+20\,a=-26$$

$$eq5 := -5\,c-10\,b-4\,d+20\,a=-47$$

```
>
```

```
> COEFFS := solve( {eq1,eq2,eq3,eq4,eq5}, {a,b,c,d,e} );
```

$$COEFFS := \{ b = \frac{23}{27}, e = \frac{13}{27}, a = \frac{-4}{3}, d = \frac{91}{27}, c = \frac{-1}{3} \}$$

```
>
```

Substituting this result, which is a set, into the original form for the partial fraction decomposition yields

```
> subs( COEFFS, FORM );
```

$$-\frac{4}{3}\frac{1}{x-1} + \frac{23}{27}\frac{1}{x+2} - \frac{1}{3}\frac{1}{(x+2)^2} + \frac{\frac{91}{27} + \frac{13}{27}x}{x^2+5}$$

```
>
```

Let's confirm the answer.

```
> simplify( % );
```

$$\frac{x^2 - 26\,x - 47}{(x-1)\,(x+2)^2\,(x^2+5)}$$

The integral is now much simpler to evaluate by hand, but here it is one last time according to *Maple*.

```
> int( %%, x );
```

$$-\frac{4}{3}\ln(x-1) + \frac{23}{27}\ln(x+2) + \frac{1}{3}\frac{1}{x+2} + \frac{13}{54}\ln(27\,x^2 + 135) + \frac{91}{135}\sqrt{5}\,\arctan\left(\frac{1}{5}x\sqrt{5}\right)$$

```
>
```

Note that this same outline can be used for ANY partial fraction problem. The key steps are i) knowing the correct form for the decomposition and ii) understanding how to identify the conditions that the coefficients must satisfy. These are (relatively) high level concepts; the user is relieved of the rather complicated mechanic manipulations.

```
>
```

2. Differential Equation Demonstrations

In this section we will demonstrate the use of *Maple* in working with differential equations. The following topics are covered in the following subsections.

Direction Fields and Graphical Solutions (Sections 1.3 and 5.2 of the Nagle/Saff/Snider text)
Symbolic Solutions to Ordinary Differential Equations and Initial Value Problems (Chapter 4 of the Nagle/Saff/Snider text)
Expressions and Functions (Chapter 4 of the Nagle/Saff/Snider text)
Systems of Ordinary Differential Equations (Sections 5.2 and 5.3 of the Nagle/Saff/Snider text)
Numeric Solutions (Sections 3.6 and 5.6 of the Nagle/Saff/Snider text)
Series Solutions (Chapter 8 of the Nagle/Saff/Snider text)
Laplace Transforms (Chapter 7 of the Nagle/Saff/Snider text)

It is a good habit to remember to use restart when starting a new problem.

```
> restart;
```

Another good habit is to load the plots and DEtools packages at the top of all worksheets, just in case they will be needed at some point in the analysis.

34

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

```
> with( DEtools );
```

[*DEnormal*, *DEplot*, *DEplot3d*, *DEplot_polygon*, *DFactor*, *Dchangevar*, *GCRD*, *LCLM*, *PDEchangecoords*, *RiemannPsols*, *abelsol*, *adjoint*, *autonomous*, *bernoullisol*, *buildsol*, *buildsym*, *canoni*, *chinisol*, *clairautsol*, *constcoeffsols*, *convertAlg*, *convertsys*, *dalembertsol*, *de2diffop*, *dfieldplot*, *diffop2de*, *eigenring*, *endomorphism_charpoly*, *equinv*, *eta_k*, *eulersols*, *exactsol*, *expsols*, *exterior_power*, *formal_sol*, *gen_exp*, *generate_ic*, *genhomosol*, *hamilton_eqs*, *indicialeq*, *infgen*, *integrate_sols*, *intfactor*, *kovacicsols*, *leftdivision*, *liesol*, *line_int*, *linearsol*, *matrixDE*, *matrix_riccati*, *moser_reduce*, *mult*, *newton_polygon*, *odeadvisor*, *odepde*, *parametricsol*, *phaseportrait*, *poincare*, *polysols*, *ratsols*, *reduceOrder*, *regular_parts*, *regularsp*, *riccati_system*, *riccatisol*, *rightdivision*, *separablesol*, *super_reduce*, *symgen*, *symmetric_power*, *symmetric_product*, *symtest*, *transinv*, *translate*, *untranslate*, *varparam*, *zoom*]

Direction Fields and Graphical Solutions (Sections 1.3 and 5.2 of the Nagle/Saff/Snider text)

Many introductory courses begin by trying to develop the student's understanding of what a differential equation is, what it means for a function to solve an ODE, and how to perform some analysis directly from the differential equation. Graphical methods are commonly employed in these discussions. The *Maple* command DEplot, from the DEtools package, provides a comprehensive interface for most graphical needs.

To begin, consider the (linear) differential equation

```
> ODE := diff( y(x), x ) = x^2 - y(x);
```

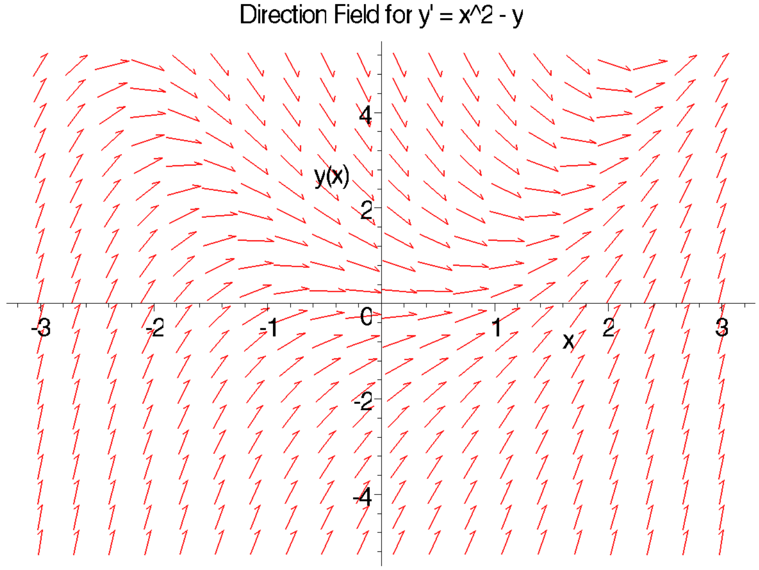
$$ODE := \frac{\partial}{\partial x} \mathrm{y}(x) = x^2 - \mathrm{y}(x)$$

Note that this is the first example used in Section 1.3 of Nagle, Saff, and Snider. The next few commands reproduce a few of the figures displayed on p.18.

The first three arguments to DEplot must provide, in order, a single n-th order ODE or a system of n first-order ODEs, a dependent variable or a list or set of dependent variables, and a range for the independent variable. The specific content of the plot is determined by all subsequent arguments to DEplot.

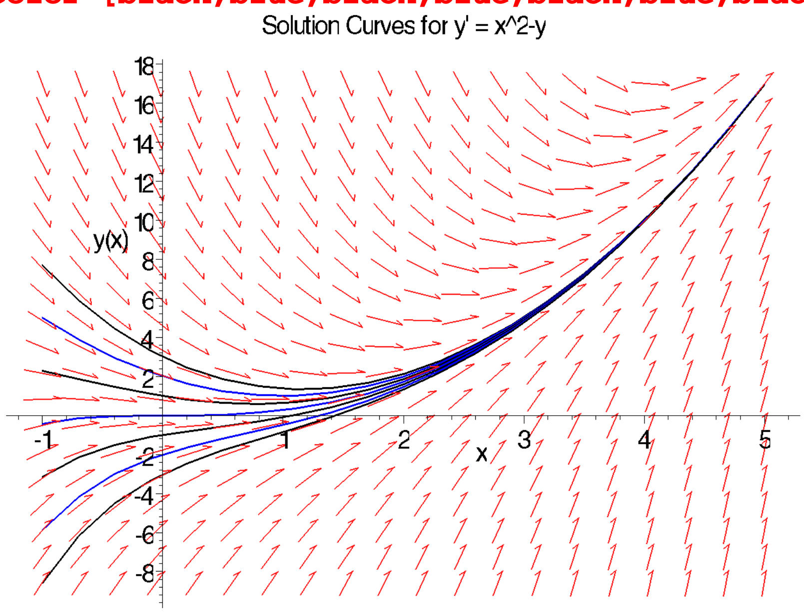
The most basic use of DEplot is to display a direction field for a differential equation. A request for a direction field is made by specifying both a range for the independent variable and the arrows = option.

```
> DEplot( ODE, y(x), x=-3..3, y=-5..5, arrows=THIN,
          title=`Direction Field for y' = x^2 - y` );
```



The solution curves through specified points are produced when the fourth argument is a set of initial conditions. If the arrows= none is included instead of arrows = thin, only the solution curves are displayed. The linecolor = gives the colors of the solution curves.

```
> DEplot( ODE, y(x), x=-1..5,
          { [0, -3], [0, -2], [0, -1], [0, 0], [0, 1], [0, 2], [0, 3]
  },arrows= thin,title=`Solution Curves for y' =
  x^2-y`,linecolor=[black,blue,black,blue,black,blue,black] );
```



Only slight alterations are needed for higher-order equations, or for systems. To illustrate, consider the predator-prey system

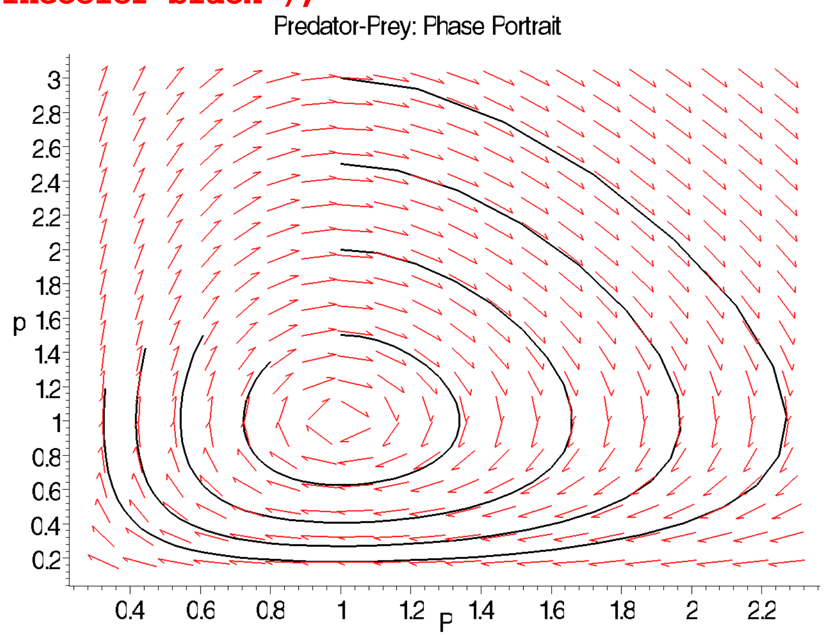
```
> SYS := [ diff( P(t), t ) = p(t)*P(t) - P(t),
           diff( p(t), t ) = 2*p(t) - 2*P(t)*p(t) ];
```

$$SYS := \left[\frac{\partial}{\partial t}\mathrm{P}(t) = \mathrm{p}(t)\,\mathrm{P}(t) - \mathrm{P}(t), \frac{\partial}{\partial t}\mathrm{p}(t) = 2\,\mathrm{p}(t) - 2\,\mathrm{p}(t)\,\mathrm{P}(t)\right]$$

where P denotes the size of the predator population and p is the prey.

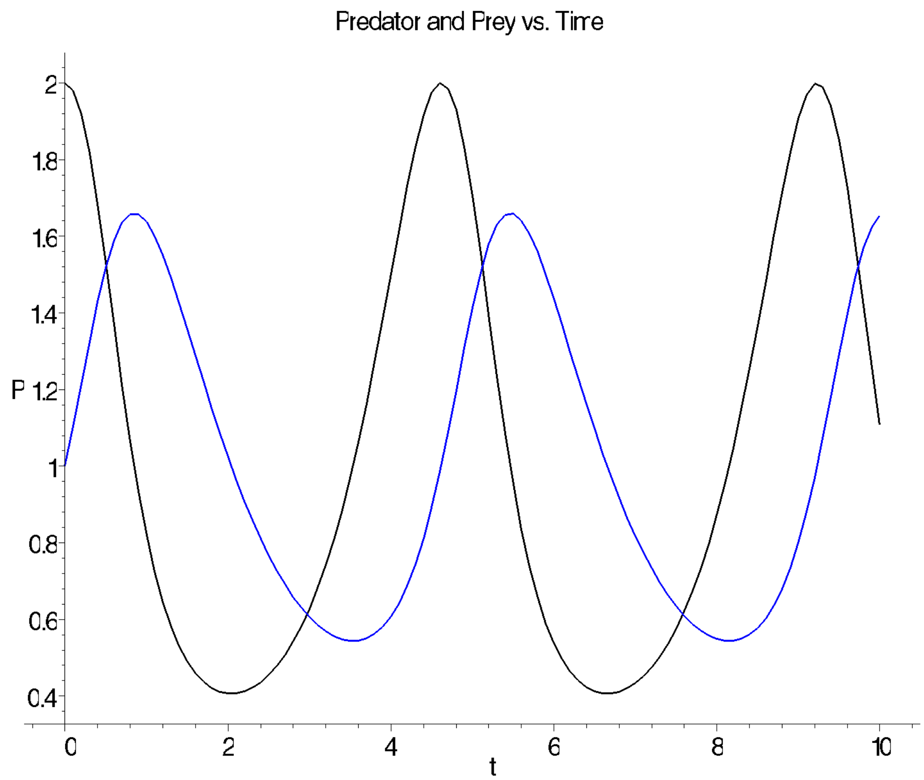
A sampling of solution curves in the phase-space can be created by specifying a set of initial conditions using the *$ operator* and the desired *scene*. (See help on $ to see how this operator functions as a real time-saver to generate the initial data.) The *stepsize=0.1* is included because the default stepsize produces a rather rough graph.

```
> DEplot( SYS, [P(t),p(t)], 0..4, { [0,1,k/2] $ k=2..6 }, scene=[P,p],
          stepsize=0.1, title=`Predator-Prey: Phase
  Portrait`,linecolor=black );
```



Plots of the individual solutions can be obtained by changing the scene. Here, individual plots of the predator and prey are created, and then displayed in a single plot using the *display* command from the *plots* package.

```
> plotp := DEplot( SYS, [P(t),p(t)], 0..10, { [0,1,2] }, scene=[t,p],
                   stepsize=0.1,linecolor=black):
  plotP := DEplot( SYS, [P(t),p(t)], 0..10, { [0,1,2] }, scene=[t,P],
                   stepsize=0.1,linecolor = blue):
  display( { plotp, plotP }, title=`Predator and Prey vs. Time`);
```



This is only the beginning of what can be done with DEplot and the DEtools package. Please consult the on-line help for additional information for a complete description of the available options and examples.

```
> ?DEtools
> ?DEtools[DEplot]
>
```

Symbolic Solutions to ODEs and IVPs (Chapter 4 of the Nagle/Saff/Snider text)

While *Maple* is quite happy to accept the inputs to its commands in almost any form, it is practically, esthetically, and pedagogically preferable to use descriptive names for the individual parts of a problem (e.g., the differential equation and boundary condition). To illustrate, let's find the general solution to x'' + 4x' + 4x = $2\,t\,\mathbf{e}^{(-2\,t)}$. The differential equation can be specified as

```
> ODE := diff( x(t), t$2 ) + 4*diff( x(t), t ) + 4*x(t) =
  2*t*exp(-2*t);
```

$$ODE := \left(\frac{\partial^2}{\partial t^2}\mathrm{x}(t)\right) + 4\left(\frac{\partial}{\partial t}\mathrm{x}(t)\right) + 4\,\mathrm{x}(t) = 2\,t\,\mathbf{e}^{(-2\,t)}$$

and the general solution is found using dsolve.

```
> GSOLN := dsolve( ODE, x(t) );
```

$$GSOLN := \mathrm{x}(t) = \frac{1}{3}t^3\,\mathbf{e}^{(-2\,t)} + \_C1\,\mathbf{e}^{(-2\,t)} + \_C2\,t\,\mathbf{e}^{(-2\,t)}$$

Note that *Maple* introduces the constants _C1 and _C2 as arbitrary constants.

The solution to an initial value problem can be obtained by including initial conditions in a set containing the differential equation. For example, the initial conditions x(1)=0, x'(1)=1 can be implemented as

```
> IC := x(1)=0, D(x)(1)=1;
```

$$IC := x(1) = 0, D(x)(1) = 1$$

Note the use of D in the specification of the derivative condition (higher order derivative conditions can be specified using @@, *Maple's* composition operator, e.g., (D@@2)(x)(0)=3). Then, the initial value problem is

```
> IVP := { ODE, IC };
```

$$IVP := \left\{ \left(\frac{\partial^2}{\partial t^2} x(t)\right) + 4\left(\frac{\partial}{\partial t} x(t)\right) + 4\, x(t) = 2\, t\, \mathbf{e}^{(-2\,t)}, D(x)(1) = 1, x(1) = 0 \right\}$$

and its solution is found using dsolve.

```
> SOLN := dsolve( IVP, x(t) );
```

$$SOLN := x(t) = \frac{1}{3} t^3\, \mathbf{e}^{(-2\,t)} + \frac{1}{3} \frac{(2\,\mathbf{e}^{(-2)} - 3)\, \mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}} - \frac{(\mathbf{e}^{(-2)} - 1)\, t\, \mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}}$$

```
>
```

Expressions and Functions (Chapter 4 of the Nagle/Saff/Snider text)

Notice how the output from *dsolve* is a *Maple* equation (the reason for this choice will be apparent when considering the solution to a system of differential equations). While there are specific cases where these equations are useful, most circumstances call for either an expression or a function. The key is to understand the structure of the output from *dsolve*.

Let's illustrate with the same example as above, with initial conditions specified, so that the constants refer to the values of the solution and its first derivative at t=1.

```
> IC := { x(1)=c1, D(x)(1)=c2 };
```

$$IC := \{ x(1) = c1, D(x)(1) = c2 \}$$

Note that these initial conditions are specified as a set, not an expression sequence. Thus, different manipulations are required to express the final IVP as a set.

```
> IVP := { ODE } union IC;
```

$$IVP := \left\{ \left(\frac{\partial^2}{\partial t^2} x(t)\right) + 4\left(\frac{\partial}{\partial t} x(t)\right) + 4\, x(t) = 2\, t\, \mathbf{e}^{(-2\,t)}, x(1) = c1, D(x)(1) = c2 \right\}$$

It is easiest to obtain the solution as an expression; it's just the right-hand side of the object *Maple* returns from dsolve.

```
> X := rhs( dsolve( IVP, x(t) ) );
```

$$X := \frac{1}{3} t^3\, \mathbf{e}^{(-2\,t)} - \frac{1}{3} \frac{(-2\,\mathbf{e}^{(-2)} + 3\,c1 + 3\,c2)\, \mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}} + \frac{(-\mathbf{e}^{(-2)} + 2\,c1 + c2)\, t\, \mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}}$$

When a Maple function is desired, it is most expedient to use unapply.

```
> XX := unapply( rhs( dsolve( IVP, x(t) ) ), t );
```

$$XX := t \to \frac{1}{3} t^3 \mathbf{e}^{(-2\,t)} - \frac{1}{3} \frac{(-2\,\mathbf{e}^{(-2)} + 3\,c1 + 3\,c2)\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}} + \frac{(-\mathbf{e}^{(-2)} + 2\,c1 + c2)\,t\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}}$$

To illustrate a potential advantage of the use of unapply, consider a situation in which you wish to include the constants as arguments to the function, e.g., x(t;c1,c2) , where c1 and c2 are the values of the function and its first derivative at t=1. Then you would use

```
> XXX := unapply( rhs( dsolve( IVP, x(t) ) ), t, c1, c2 );
```

$$XXX := (t, c1, c2) \to \frac{1}{3} t^3 \mathbf{e}^{(-2\,t)} - \frac{1}{3} \frac{(-2\,\mathbf{e}^{(-2)} + 3\,c1 + 3\,c2)\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}} + \frac{(-\mathbf{e}^{(-2)} + 2\,c1 + c2)\,t\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}}$$

and the resulting function, XXX, can be used to find the solution passing through the point (1,0) with unit slope as follows

```
> XXX(t,0,1);
```

$$\frac{1}{3} t^3 \mathbf{e}^{(-2\,t)} - \frac{1}{3} \frac{(-2\,\mathbf{e}^{(-2)} + 3)\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}} + \frac{(-\mathbf{e}^{(-2)} + 1)\,t\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}}$$

The same computation using X and XX would appear as

```
> subs( c1=0, c2=1, X );
```

$$\frac{1}{3} t^3 \mathbf{e}^{(-2\,t)} - \frac{1}{3} \frac{(-2\,\mathbf{e}^{(-2)} + 3)\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}} + \frac{(-\mathbf{e}^{(-2)} + 1)\,t\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}}$$

```
> subs( c1=0, c2=1, XX(t) );
```

$$\frac{1}{3} t^3 \mathbf{e}^{(-2\,t)} - \frac{1}{3} \frac{(-2\,\mathbf{e}^{(-2)} + 3)\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}} + \frac{(-\mathbf{e}^{(-2)} + 1)\,t\,\mathbf{e}^{(-2\,t)}}{\mathbf{e}^{(-2)}}$$

Next, consider the problem of verifying that the solution is correct.

```
> simplify( subs( x(t)=X, ODE ) );
```

$$2\,t\,\mathbf{e}^{(-2\,t)} = 2\,t\,\mathbf{e}^{(-2\,t)}$$

```
> simplify( subs( x(t)=XX(t), ODE ) );
```

$$2\,t\,\mathbf{e}^{(-2\,t)} = 2\,t\,\mathbf{e}^{(-2\,t)}$$

```
> simplify( subs( x(t)=XXX(t,c1,c2), ODE ) );
```

$$2\,t\,\mathbf{e}^{(-2\,t)} = 2\,t\,\mathbf{e}^{(-2\,t)}$$

Note that, because both x and XX are functions of a single variable, it is also possible to use *simplify (subs(x=XX, ODE));*. The same is not true for XXX, as this function requires three arguments.

To conclude this discussion, let's plot the solutions that have critical points (i.e., c2=0) at (1,-1), (1,1), and (1,3) using each of X, XX, and XXX.

```
> S1 := { seq( subs( c1=C, c2=0, X ), C={-1,1,3} ) };
```

$$S1 := \{\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}-3)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}-2)t e^{(-2t)}}{e^{(-2)}},$$
$$\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}+3)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}+2)t e^{(-2t)}}{e^{(-2)}},$$
$$\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}+9)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}+6)t e^{(-2t)}}{e^{(-2)}}\}$$

```
> S2 := { seq( subs( c1=C, c2=0, XX('t') ), C={-1,1,3} ) };#The use of 't' is explained below.
```

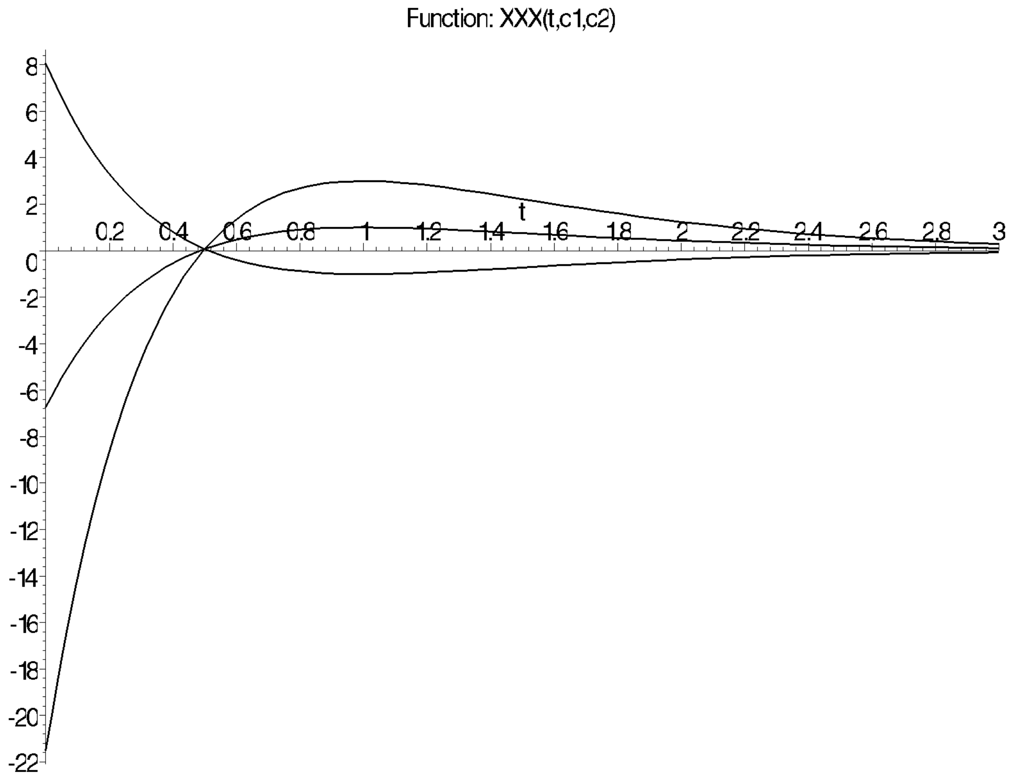
$$S2 := \{\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}-3)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}-2)t e^{(-2t)}}{e^{(-2)}},$$
$$\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}+3)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}+2)t e^{(-2t)}}{e^{(-2)}},$$
$$\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}+9)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}+6)t e^{(-2t)}}{e^{(-2)}}\}$$

```
> S3 := { seq( XXX('t',C,0), C={-1,1,3} ) };
```

$$S3 := \{\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}-3)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}-2)t e^{(-2t)}}{e^{(-2)}},$$
$$\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}+3)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}+2)t e^{(-2t)}}{e^{(-2)}},$$
$$\frac{1}{3}t^3 e^{(-2t)} - \frac{1}{3}\frac{(-2e^{(-2)}+9)e^{(-2t)}}{e^{(-2)}} + \frac{(-e^{(-2)}+6)t e^{(-2t)}}{e^{(-2)}}\}$$

Note that the use of single quotes (') is needed so that the evaluation of t is delayed until after the *subs* command is completed.

```
plot( S3, t=0..3, title=`Function: XXX(t,c1,c2)` );
```



The structure of this solution (linear combination of a basis of solutions to the homogeneous equation plus a particular solution) is evident in the solutions we have found. The individual components can be easily extracted and manipulated. Here we demonstrate how a general solution is used to determine the solution of a specific initial value problem. Once the general solution is found

```
> GSOLN := rhs( dsolve( ODE, x(t) ) );
```

$$GSOLN := \frac{1}{3} t^3 \mathbf{e}^{(-2\,t)} + \_C1\, \mathbf{e}^{(-2\,t)} + \_C2\, t\, \mathbf{e}^{(-2\,t)}$$

the two equations that must be satisfied by C1 and C2 can be constructed:

```
> eq1 := subs( t=1, GSOLN=3 );
  eq2 := subs( t=1, diff(GSOLN,t)=0 );
```

$$eq1 := \frac{1}{3} \mathbf{e}^{(-2)} + \_C1\, \mathbf{e}^{(-2)} + \_C2\, \mathbf{e}^{(-2)} = 3$$

$$eq2 := \frac{1}{3} \mathbf{e}^{(-2)} - 2\, \_C1\, \mathbf{e}^{(-2)} - \_C2\, \mathbf{e}^{(-2)} = 0$$

and solved

```
> solC := solve( { eq1, eq2 }, { _C1, _C2 } );
```

$$solC := \{\_C1 = \frac{1}{3} \frac{2\,\mathbf{e}^{(-2)} - 9}{\mathbf{e}^{(-2)}}, \_C2 = -\frac{\mathbf{e}^{(-2)} - 6}{\mathbf{e}^{(-2)}}\}$$

These solutions obviously simplify (but require an application of *expand* to force all simplifications).

```
solC := simplify(expand( solC ));
```

$$solC := \{\_C1 = \frac{2}{3} - 3\,\mathbf{e}^2,\ \_C2 = -1 + 6\,\mathbf{e}^2\}$$

Floating point approximations to these constants are obtained using *evalf:*

```
> evalf( solC );
```

$$\{\_C1 = -21.50050163,\ \_C2 = 43.33433659\}$$

The solution to the IVP is found to be

```
> IVPsoln := factor( subs( solC, GSOLN ) );
```

$$IVPsoln := \frac{1}{3}\mathbf{e}^{(-2\,t)}\,(t^3 + 2 - 9\,\mathbf{e}^2 - 3\,t + 18\,t\,\mathbf{e}^2)$$

```
> simplify( IVPsoln - subs( c1=3, c2=0, X ) );
```

$$0$$

```
> simplify( IVPsoln - subs( c1=3, c2=0, XX(t) ) );
```

$$0$$

```
> simplify( IVPsoln - XXX(t,3,0) );
```

$$0$$

To conclude, let's see three different ways in which the terms contributing to the homogeneous and particular solutions can be obtained. Working directly from the general solution we find

```
> Xp := subs( _C1=0, _C2=0, GSOLN );
```

$$Xp := \frac{1}{3}t^3\,\mathbf{e}^{(-2\,t)}$$

```
> Xh := GSOLN - Xp;
```

$$Xh := \_C1\,\mathbf{e}^{(-2\,t)} + \_C2\,t\,\mathbf{e}^{(-2\,t)}$$

```
> X1 := subs( _C1=1, _C2=0, Xh );
```

$$X1 := \mathbf{e}^{(-2\,t)}$$

```
> X2 := subs( _C1=0, _C2=1, Xh );
```

$$X2 := t\,\mathbf{e}^{(-2\,t)}$$

Alternatively, the same information can be obtained directly from *dsolve*.

```
> ODEh := lhs(ODE)=0;
```

$$ODEh := \left(\frac{\partial^2}{\partial t^2}\mathrm{x}(t)\right) + 4\left(\frac{\partial}{\partial t}\mathrm{x}(t)\right) + 4\,\mathrm{x}(t) = 0$$

```
> IC1 := x(0)=0, D(x)(0)=1;
  X1 := rhs( dsolve( { ODEh, IC1 }, x(t) ) );
```

$$IC1 := \mathrm{x}(0) = 0,\ \mathrm{D}(x)(0) = 1$$

$$X1 := t\,\mathbf{e}^{(-2\,t)}$$

```
> IC2 := x(0)=1, D(x)(0)=0;
  X2 := rhs( dsolve( { ODEh, IC2 }, x(t) ) );
```

$$IC2 := \mathrm{x}(0) = 1,\ \mathrm{D}(x)(0) = 0$$

$$X2 := \mathbf{e}^{(-2\,t)} + 2\,t\,\mathbf{e}^{(-2\,t)}$$

```
> ICp := x(0)=0, D(x)(0)=0;
  Xp := rhs( dsolve( { ODE, ICp }, x(t) ) );
```

$$ICp := x(0) = 0, D(x)(0) = 0$$

$$Xp := \frac{1}{3} t^3 e^{(-2 t)}$$

Although, to be honest, in this case it is probably simplest to explicitly identify the appropriate terms in the general solution and explicitly define the components of the solution.

```
> x1 := exp(-2*t);
  x2 := t*exp(-2*t);
  xp := t^3/3 * exp(-2*t);
> 
```

This example hopefully illustrates that while almost all manipulations can be done in Maple, some steps should still be done by hand.

Additional information about *dsolve* is available from the on-line help (*?dsolve*). Note, in particular, that *dsolve* may return an implicit solution or a solution in parametric form. Explicit solutions can, sometimes, be coerced by using the optional argument *explicit*. Other optional arguments are discussed later in this supplement.

```
> ?dsolve
```

Systems of ODEs (Sections 5.2 and 5.3 of the Nagle/Saff/Snider text)

Systems of ODEs are analyzed in a completely parallel manner. The system of equations is specified as a set (with or without initial conditions); the dependent variables are also specified as a set. To illustrate, consider the (linearized) model of a pendulum: $\frac{d^2 \theta(t)}{dt^2} + 3 \theta(t) = 0$. This second-order equation is equivalent to the system of first-order equations

```
> SYS := { diff( theta(t), t ) = v(t), diff( v(t), t ) = -3*theta(t) };
```

$$SYS := \{ \frac{\partial}{\partial t} \theta(t) = v(t), \frac{\partial}{\partial t} v(t) = -3 \theta(t) \}$$

$$SYS := \{ \frac{\partial}{\partial t} \theta(t) = v(t), \frac{\partial}{\partial t} v(t) = -3 \theta(t) \}$$

in terms of the dependent variables

```
> FNS := { theta(t), v(t) };
```

$$FNS := \{ \theta(t), v(t) \}$$

The general solution is found to be

```
> SOL := dsolve( SYS, FNS );
```

$SOL :=$

$$\{ v(t) = \cos(\sqrt{3}\, t) \_C1 - \sqrt{3} \sin(\sqrt{3}\, t) \_C2, \theta(t) = \frac{1}{3}\sqrt{3} \sin(\sqrt{3}\, t) \_C1 + \cos(\sqrt{3}\, t) \_C2 \}$$

We are now in a position to understand why it is important that *dsolve* returns (a set of) equations. Because a set is unordered, there is no reason to expect the elements of the solution set to appear in the same order each time *Maple* is executed. That is, there is no "first" term in the solution vector. There is, nonetheless, a very simple means to ensure that the appropriate term from the solution is extracted. It is still possible to obtain the individual component as either an expression

```
> SOLV := subs( SOL, v(t) );
```

$$SOLV := \cos(\sqrt{3}\,t)\,\_C1 - \sqrt{3}\,\sin(\sqrt{3}\,t)\,\_C2$$

or as a function

```
> SOLT := unapply( subs( SOL, theta(t) ), t );
```

$$SOLT := t \to \frac{1}{3}\sqrt{3}\,\sin(\sqrt{3}\,t)\,\_C1 + \cos(\sqrt{3}\,t)\,\_C2$$

Most systems of ODEs do not have explicit solutions. *Maple* has no difficulties producing series solutions for systems. We illustrate using the corresponding nonlinear pendulum equation.

```
> SYSnl := { diff( theta(t), t ) = v(t),
             diff( v(t), t ) = -3*sin( theta(t) ) };
```

$$SYSnl := \{\frac{\partial}{\partial t}\mathrm{v}(t) = -3\,\sin(\theta(t)), \frac{\partial}{\partial t}\theta(t) = \mathrm{v}(t)\}$$

Only a small number of terms will be needed to compare with the linearized solution.

```
> Order := 3:
> SOLnl := dsolve( SYSnl, FNS, type=series );
```

$$SOLnl := \{\mathrm{v}(t) = \mathrm{v}(0) - 3\,\sin(\theta(0))\,t - \frac{3}{2}\cos(\theta(0))\,\mathrm{v}(0)\,t^2 + \mathrm{O}(t^3),$$

$$\theta(t) = \theta(0) + \mathrm{v}(0)\,t - \frac{3}{2}\sin(\theta(0))\,t^2 + \mathrm{O}(t^3)\}$$

```
> SOLnlV := subs( SOLnl, v(t) );
```

$$SOLnlV := \mathrm{v}(0) - 3\,\sin(\theta(0))\,t - \frac{3}{2}\cos(\theta(0))\,\mathrm{v}(0)\,t^2 + \mathrm{O}(t^3)$$

Comparisons with the solution to the linearized equation can be performed using the series command, as follows

```
> series( SOLV, t=0 );
```

$$\_C1 - 3\,\_C2\,t - \frac{3}{2}\_C1\,t^2 + \mathrm{O}(t^3)$$

```
>
```

The real power of *Maple* for systems of ODEs is seen in its handling of numerical solutions. This procedure will be demonstrated in the next section.

Numeric Solutions (Sections 3.6 and 5.6 of the Nagle/Saff/Snider text)

Numeric solutions to initial value problems can be used in a variety of ways in an introductory course, including direction fields, qualitative behavior of solutions, existence theory, and regularity theory. The *dsolve* command can be used for each of these topics. The default method used is rkf45, a *Maple* implementation of the Fehlberg fourth-fifth order Runge-Kutta method.

Notice that there are no built-in *Maple* procedures for Euler's method, improved Euler's method and other typical numerical methods discussed in an introductory course. These methods are easily implemented in *Maple*, as will be demonstrated in a later section.

To begin our discussion, consider the initial value problem

```
> ODE := diff( x(t), t ) = x(t)^2 + t:
  IC  := x(0)=0:
  IVP := { ODE, IC };
```

$$IVP := \{ x(0) = 0, \frac{\partial}{\partial t} x(t) = x(t)^2 + t \}$$

While this problem does have an explicit solution, it cannot be found using methods typically found in an introductory course. But, it is reasonable to ask if a solution does exist for all t > 0. Numerical solutions can be used, with caution, to discuss questions of this type.

The inclusion of the optional argument *type=numeric* is the basic interface to the numerical version of *dsolve*.

```
> numSOLN := dsolve( IVP, x(t),type=numeric);
```

numSOLN := **proc**(*rkf45_x*) ... **end**

The output from *dsolve* is now quite different. What this is saying is that numSOLN is a *Maple* procedure that can be used to compute the solution at a given (numeric) value of t. For example:

```
> numSOLN( 0 );
```

$$[t = 0, x(t) = 0]$$

```
> numSOLN( 1/10 );
```

$$\left[t = \frac{1}{10}, x(t) = .005000500074847118\right]$$

Notice that the output from the procedure created by *dsolve* is a list of equations with one equation for the independent variable and one equation for each dependent variable. The structure of this solution is ideal for use with the *subs* command to extract specific components of a solution or even to use the results in subsequent calculations. But first, let's evaluate the solution at more points.

```
> numSOLN( 0.25 );
```

$$[t = .25, x(t) = .03129892370880233]$$

```
> numSOLN( 1 );
```

$$[t = 1, x(t) = .5571617678870111]$$

```
> numSOLN( 0 );
```

$$[t = 0, x(t) = 0]$$

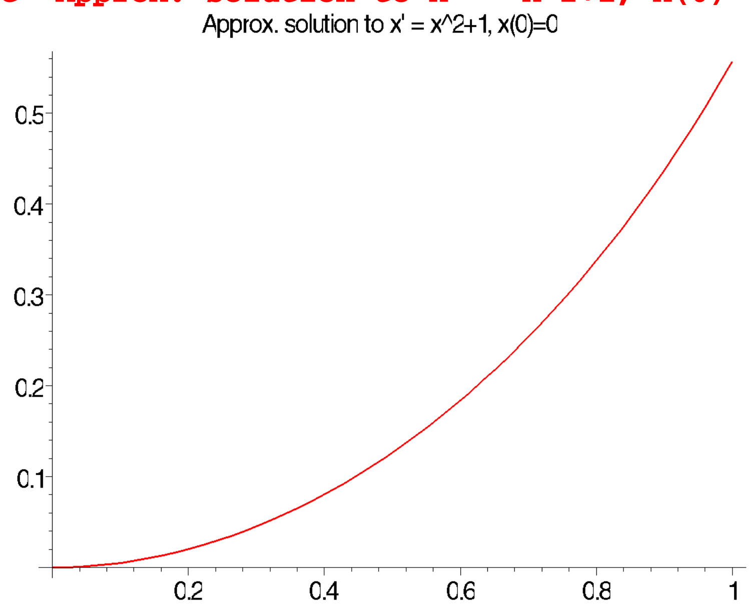
```
> numSOLN( 1 );
```

$$[t = 1, x(t) = .5571617684443276]$$

This last result is a little distressing -- why have the values of the solution changed from the first time the solution was evaluated? To explain this change, it is necessary to realize that each call to *numSOLN* uses the results of the most recent computations as the initial conditions for the current call. Thus, the first evaluation of the solution at t=1 used the computation at .25 and the second at 0. The difference between the two values gives an indication of the accumulation error for this solution. To reset the solution to the original initial condition, simply re-execute the *dsolve* command. This design characteristic can take some time to adjust to, but it is quite reasonable for most uses of a numerical solution.

The plot of a numerical solution can be created in a number of different ways. One of the simplest ways is to use the *odeplot* command from the *plots* package, which must be loaded prior to use. The basic syntax for *odeplot* expects the first argument to be the output from numeric dsolve. The second argument should be a list of two or three expressions involving the dependent and independent variables. For example:

```
> numSOLN := dsolve( IVP, x(t), type=numeric ):
  odeplot( numSOLN,  [t,x(t)] , 0..1,
           title=`Approx. solution to x' = x^2+1, x(0)=0` );
```



Both numeric *dsolve* and *odeplot* work equally well with higher-order equations and with systems. The basic operations for a higher-order problem will be illustrated using the damped nonlinear pendulum.

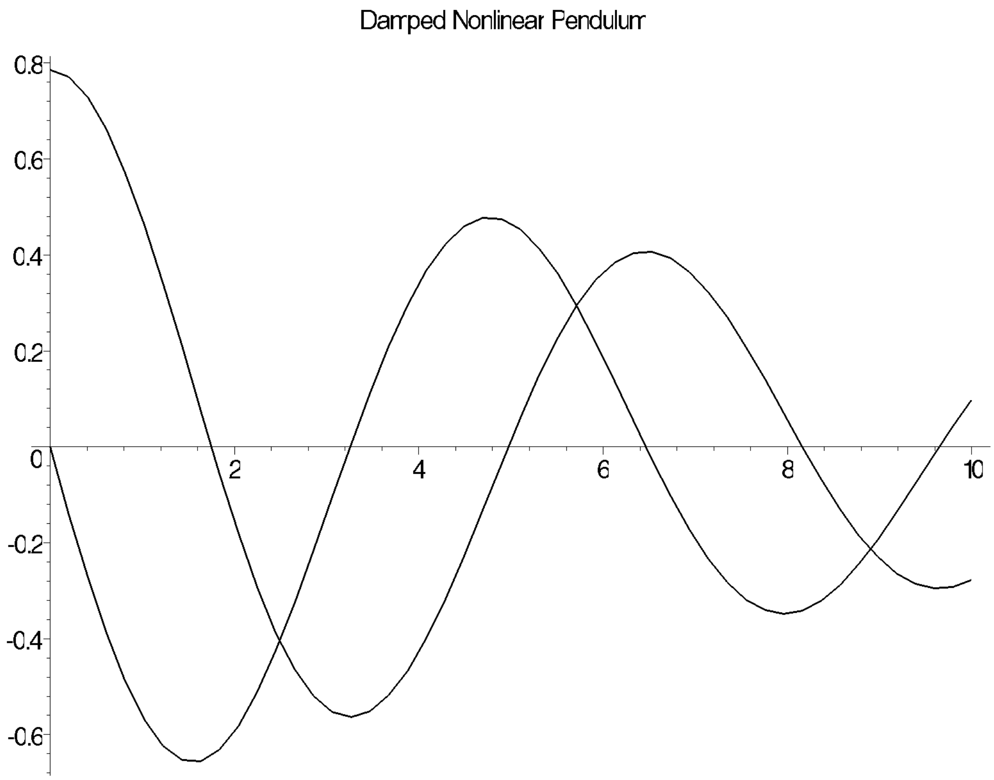
```
> ODE := diff( theta(t), t$2 ) + 0.2*diff( theta(t), t ) +
  sin(theta(t)) = 0:
  IC := theta(0)=Pi/4, D(theta)(0)=0:
  IVP := { ODE, IC };
```

$$IVP := \{\left(\frac{\partial^2}{\partial t^2}\theta(t)\right) + .2\left(\frac{\partial}{\partial t}\theta(t)\right) + \sin(\theta(t)) = 0, \theta(0) = \frac{1}{4}\pi, \mathrm{D}(\theta)(0) = 0\}$$

```
> PEND := dsolve( IVP, theta(t), type=numeric ):
```

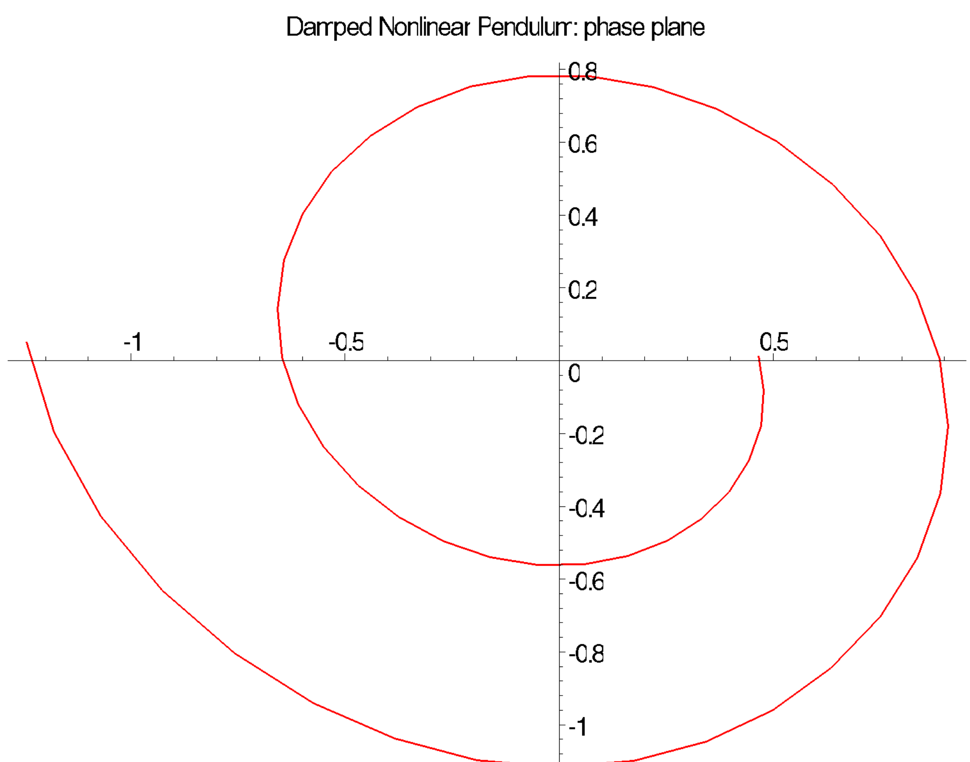
Multiple graphs are included in one plot when the second argument to *odeplot* is a list of lists. (Note how the derivative of the solution is selected using *diff*.)

```
> odeplot( PEND, [ [ t, theta(t) ], [ t, diff(theta(t),t) ] ], 0..10,
           title=`Damped Nonlinear Pendulum` );
```



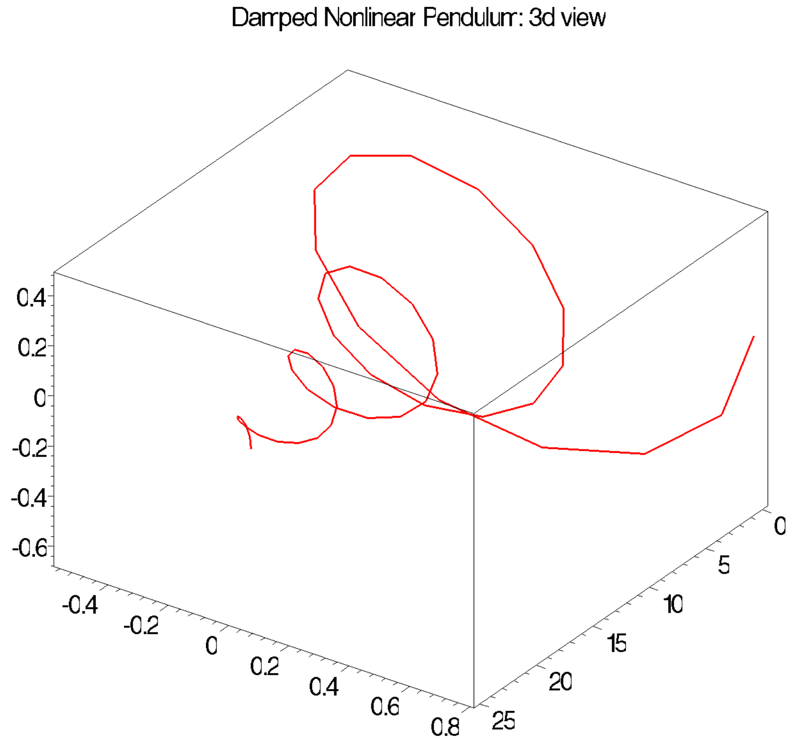
The phase portrait is also easy to create.

```
> odeplot( PEND, [ diff(theta(t),t), theta(t) ], -5..5,
           title=`Damped Nonlinear Pendulum: phase plane` );
```



A three-dimensional view can sometimes be quite illustrative.

```
> odeplot( PEND, [ t, theta(t), diff(theta(t),t) ], 0..25,
            title=`Damped Nonlinear Pendulum: 3d view`,
  axes=BOXED,color=red );
```



You can now click on the image and rotate it by dragging the mouse.

There are times when the standard output from numeric *dsolve* is not suitable for further processing. The *output*= and *value*= optional arguments provide some flexibility. Specifying *output=listprocedure* will cause *dsolve* to return a *Maple* list of procedures.

```
> PEND2 := dsolve( IVP, theta(t), type=numeric ,output=listprocedure );
```

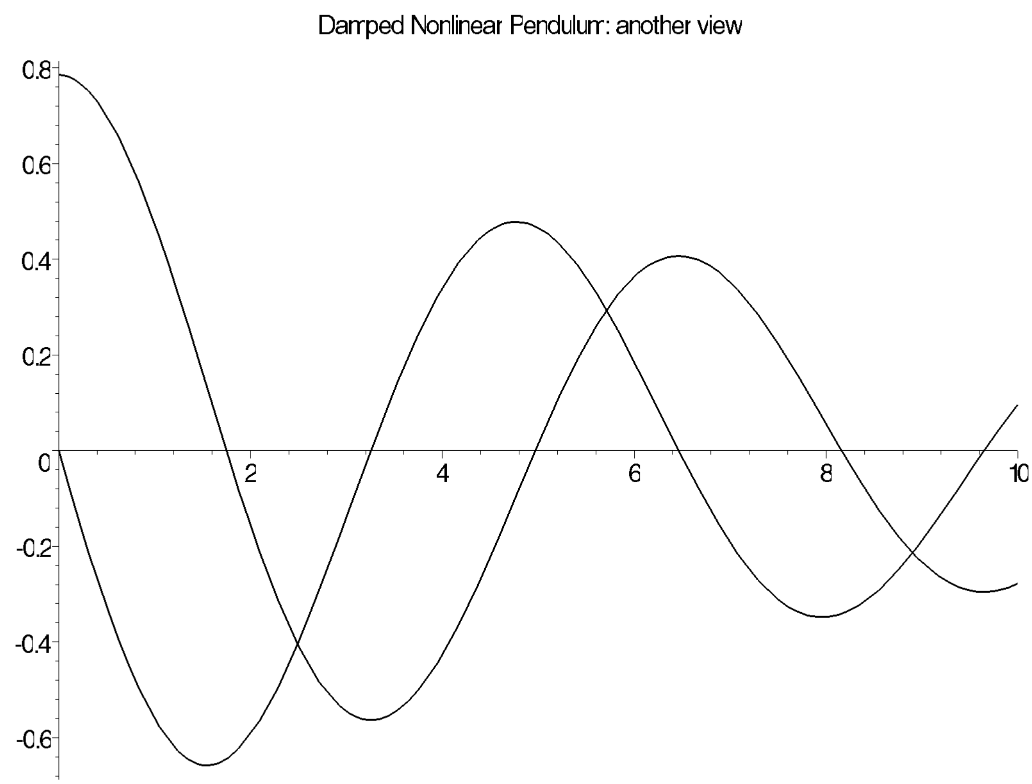
$$PEND2 := \left[t = (\mathbf{proc}(t) \ \ldots \ \mathbf{end}), \theta(t) = (\mathbf{proc}(t) \ \ldots \ \mathbf{end}), \frac{\partial}{\partial t}\theta(t) = (\mathbf{proc}(t) \ \ldots \ \mathbf{end}) \right]$$

```
> THETA := subs( PEND2, theta(t) );
  dTHETA := subs( PEND2, diff(theta(t),t) );
```

$$THETA := \mathbf{proc}(t) \ \ldots \ \mathbf{end}$$

$$dTHETA := \mathbf{proc}(t) \ \ldots \ \mathbf{end}$$

```
> plot( { THETA, dTHETA }, 0..10,
        title=`Damped Nonlinear Pendulum: another view` );
```



The *value*= argument is used when the value of the solution is needed only for specified values of the independent variable. This method can be particularly useful when creating a table of data.

```
> PEND3 := dsolve( IVP, theta(t), type=numeric,
                   value=array([ i/100. $ i=320..330 ]) );
```

$$PEND3 := \left[\begin{array}{c} \left[t, \theta(t), \frac{\partial}{\partial t}\theta(t)\right] \\ \left[\begin{array}{ccc} 3.200000000 & -.5620403790 & -.02994513342 \\ 3.210000000 & -.5623128991 & -.02456030038 \\ 3.220000000 & -.5625316139 & -.01918415046 \\ 3.230000000 & -.5626966125 & -.01381712091 \\ 3.240000000 & -.5628079882 & -.008459647176 \\ 3.250000000 & -.5628658387 & -.003112162912 \\ 3.260000000 & -.5628702662 & .002224900014 \\ 3.270000000 & -.5628213769 & .007551111503 \\ 3.280000000 & -.5627192815 & .01286604321 \\ 3.290000000 & -.5625640950 & .01816926855 \\ 3.300000000 & -.5623559366 & .02346036265 \end{array}\right] \end{array}\right]$$

Extracting information from this table can be frustrating. Note that the structure of PEND3 is a two-dimensional (column) array. The first element of PEND3 is the three-dimensional (row) vector which describes the contents of the matrix contained in the second element of PEND3.

```
> VARS := evalm( PEND3[1,1] );
```

$$VARS := \left[t, \theta(t), \frac{\partial}{\partial t}\theta(t) \right]$$

```
> VARS[2];
```

$$\theta(t)$$

It is also possible to nest the indices to directly access entries in this data structure. For example, the pendulum is at rest sometime between t=3.25 and t=3.26. At this time the angle describing the pendulum's position is approximately

```
> rest := ( PEND3[2,1][6,2] + PEND3[2,1][7,2] )/2;
```

$$rest := -.5628680525$$

which, converted to degrees, is

```
> evalf( rest * 180/Pi );
```

$$-32.24996384$$

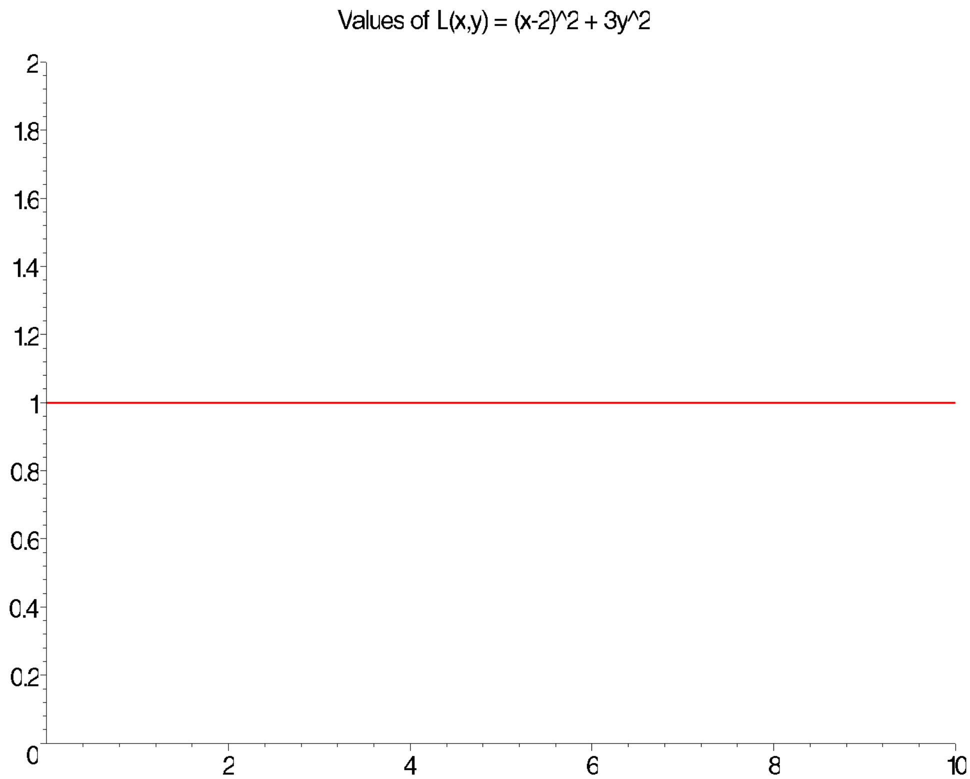
To illustrate the use of numeric *dsolve* for a system, consider the second-order system

```
> ODE := diff( x(t), t ) = 3*x(t)^2*y(t) - 6*y(t)^2,
         diff( y(t), t ) = - x(t)^3 + 2*x(t)^2 + 2*x(t)*y(t) - 4*y(t):
  IC := x(0) = 1, y(0)=0:
  IVP := { ODE, IC };
```

$$IVP := \left\{ \frac{\partial}{\partial t}x(t) = 3\,x(t)^2\,y(t) - 6\,y(t)^2, \frac{\partial}{\partial t}y(t) = -x(t)^3 + 2\,x(t)^2 + 2\,x(t)\,y(t) - 4\,y(t), x(0) = 1, y(0) = 0 \right\}$$

The level curves of the (Liapunov) function $L(x, y) = (x-2)^2 + 3\,y^2$ are trajectories for this system. That is, $L(x(t), y(t)) = L(x_0, y_0)$ for all t>0. This can be illustrated using numeric *dsolve* and *odeplot* as follows.

```
> numSOLN := dsolve( IVP, [ x(t), y(t) ], type=numeric ):
  odeplot( numSOLN, [ t, (x(t)-2)^2 + 3*y(t)^2 ],
  0..10, title=`Values of L(x,y) = (x-2)^2 + 3y^2`,view=[0..10,0..2] );
```



Note that if the *view* = is omitted, then the plot looks a little surprising at first glance. A close examination of the default vertical range selected by *Maple* explains this apparent anomaly.

The preceding examples illustrate a few of the options that are available in numeric *dsolve*. Full details are available from the on-line help pages for

```
> ?dsolve
> ?dsolve[numeric]
> ?dsolve[dverk78]
```

and

```
> ?plots[odeplot]
```

Series Solutions (Chapter 8 of the Nagle/Saff/Snider text)

Some differential equations, and systems, are most approachable using power series techniques. *Maple* provides access to this method by way of the *type=series* optional argument to *dsolve*.

Consider the differential equation, in normal form,

```
> ODE := diff( y(x), x$2 ) + 1/x * diff( y(x), x ) - 1/(x^2+1) * y(x) =
  0;
```

$$ODE := \left(\frac{\partial^2}{\partial x^2}\,y(x)\right) + \frac{\frac{\partial}{\partial x}\,y(x)}{x} - \frac{y(x)}{x^2+1} = 0$$

The first few terms of the power series, centered at x=0, can be obtained with the command

```
> dsolve( ODE, y(x), type=series );
```

$$y(x) = \_C1\left(1+\frac{1}{4}x^2-\frac{3}{64}x^4+O(x^6)\right) + \_C2\left(\ln(x)\left(1+\frac{1}{4}x^2-\frac{3}{64}x^4+O(x^6)\right)+\left(-\frac{1}{4}x^2+\frac{1}{128}x^4+O(x^6)\right)\right)$$

To obtain additional terms in the sequence, modify the system constant *Order*.

```
> Order := 10:
> Y9 := dsolve( ODE, y(x), type=series );
```

$$Y9 := y(x) = \_C1\left(1+\frac{1}{4}x^2-\frac{3}{64}x^4+\frac{5}{256}x^6-\frac{175}{16384}x^8+O(x^{10})\right)+\_C2\Bigg(\ln(x)\left(1+\frac{1}{4}x^2-\frac{3}{64}x^4+\frac{5}{256}x^6-\frac{175}{16384}x^8+O(x^{10})\right) + \left(-\frac{1}{4}x^2+\frac{1}{128}x^4+\frac{1}{1536}x^6-\frac{265}{196608}x^8+O(x^{10})\right)\Bigg)$$

Note that x=0 is a regular singular point for this equation. Note also the structure of this solution. The presence of the logarithmic factor is enlightening; that same term also complicates the conversion of this solution from an equation whose RHS is of type *series* into an expression that can be plotted, etc.

Series solutions centered at other points can be requested by specifying a full set of initial conditions at the desired value of the independent variable. Here, the series expansion of the solution centered at x=-1 will be produced.

```
> IVP := { ODE, y(-1)=c1, D(y)(-1)=c2 };
```

$$IVP := \left\{\left(\frac{\partial^2}{\partial x^2}y(x)\right)+\frac{\frac{\partial}{\partial x}y(x)}{x}-\frac{y(x)}{x^2+1}=0,\ y(-1)=c1,\ D(y)(-1)=c2\right\}$$

```
> Order:=6:
> Y5 := dsolve( IVP, y(x), type=series );
```

$$Y5 := y(x) = c1 + c2\,(x+1) + \left(\frac{1}{4}c1+\frac{1}{2}c2\right)(x+1)^2 + \left(\frac{5}{12}c2+\frac{1}{6}c1\right)(x+1)^3 + \left(\frac{11}{96}c1+\frac{1}{3}c2\right)(x+1)^4 + \left(\frac{1}{12}c1+\frac{127}{480}c2\right)(x+1)^5 + O((x+1)^6)$$

Here are two ways of obtaining the series solutions for two linearly independent solutions to this ODE. First, it is possible to simply specify two different sets of values for the constants c1 and c2.

```
> Y5a := subs( c1=1, c2=0, rhs(Y5 ) );
  Y5b := subs( c1=0, c2=1, rhs(Y5 ) );
```

$$Y5a := 1 + \frac{1}{4}(x+1)^2 + \frac{1}{6}(x+1)^3 + \frac{11}{96}(x+1)^4 + \frac{1}{12}(x+1)^5 + \mathrm{O}((x+1)^6)$$

$$Y5b := x+1 + \frac{1}{2}(x+1)^2 + \frac{5}{12}(x+1)^3 + \frac{1}{3}(x+1)^4 + \frac{127}{480}(x+1)^5 + \mathrm{O}((x+1)^6)$$

Alternatively, *Maple* can separately *collect* the terms involving c1 and c2. But, before this *collect* can be used, the solution must be converted to a polynomial. (As long as the O term is present, the solution is an equation whose RHS is of type *series*.)

```
> Y5 := convert( rhs(Y5), polynom );
```

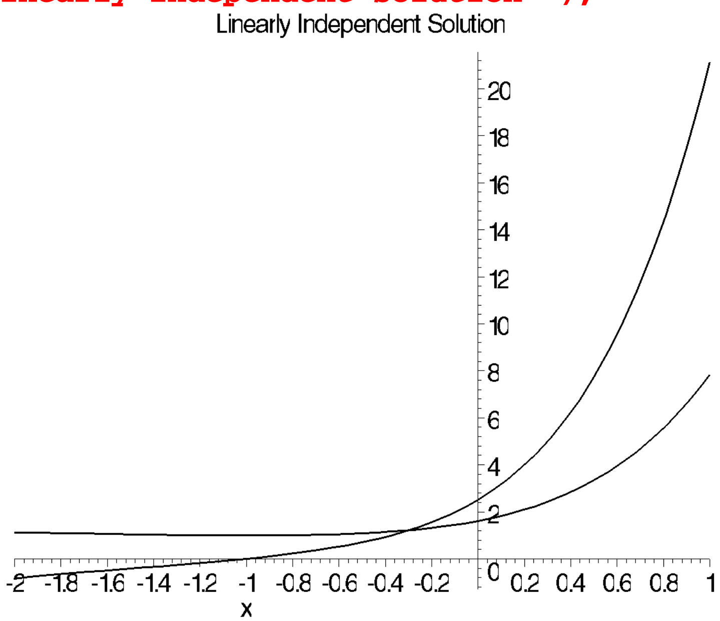
$$Y5 := c1 + c2\,(x+1) + \left(\frac{1}{4}c1 + \frac{1}{2}c2\right)(x+1)^2 + \left(\frac{5}{12}c2 + \frac{1}{6}c1\right)(x+1)^3 + \left(\frac{11}{96}c1 + \frac{1}{3}c2\right)(x+1)^4 + \left(\frac{1}{12}c1 + \frac{127}{480}c2\right)(x+1)^5$$

```
> Y5 := collect( Y5, [ c1, c2 ] );
```

$$Y5 := \left(\frac{1}{4}(x+1)^2 + \frac{1}{6}(x+1)^3 + \frac{11}{96}(x+1)^4 + 1 + \frac{1}{12}(x+1)^5\right)c1 + \left(\frac{5}{12}(x+1)^3 + x + 1 + \frac{1}{2}(x+1)^2 + \frac{127}{480}(x+1)^5 + \frac{1}{3}(x+1)^4\right)c2$$

Note that the ordering of the terms in this last expression is less than optimal. (Controlling the order of terms in a polynomial is not a simple task in *Maple*.)

```
> plot( map( convert, { Y5a, Y5b }, polynom ), x=-2..1,
        title=`Linearly Independent Solution` );
```



Laplace Transforms (Chapter 7 of the Nagle/Saff/Snider text)

All uses of Laplace transforms should begin by reading the integral transform library into the *Maple* session.

```
> with(inttrans);
```

$$[addtable, fourier, fouriercos, fouriersin, hankel, hilbert, invfourier, invhilbert, invlaplace, invmellin, laplace, mellin, savetable]$$

The two commands that form the Laplace transform pair are *laplace* and *invlaplace*. These commands can be used explicitly or implicitly to solve a differential equation. To illustrate both approaches, consider the problem of determining the general solution to

```
> ODE := diff( y(t), t$2 ) + 3*diff( y(t), t ) - 4 * y(t) = 0;
```

$$ODE := \left(\frac{\partial^2}{\partial t^2}\mathrm{y}(t)\right) + 3\left(\frac{\partial}{\partial t}\mathrm{y}(t)\right) - 4\,\mathrm{y}(t) = 0$$

One approach is to begin by applying the Laplace transform to this equation and solving for the Laplace transform, Y(s), of the solution y(t).

```
> lapODE := laplace( ODE, t, s );
```

$$lapODE := s\,(s\,\mathrm{laplace}(\mathrm{y}(t), t, s) - \mathrm{y}(0)) - \mathrm{D}(y)(0) + 3\,s\,\mathrm{laplace}(\mathrm{y}(t), t, s) - 3\,\mathrm{y}(0) - 4\,\mathrm{laplace}(\mathrm{y}(t), t, s) = 0$$

```
> SOLY := solve( lapODE, laplace( y(t), t, s ) );
```

$$SOLY := \frac{s\,\mathrm{y}(0) + \mathrm{D}(y)(0) + 3\,\mathrm{y}(0)}{s^2 + 3\,s - 4}$$

Next, the inverse Laplace transform is applied to find the solution y(t).

```
> SOLy := invlaplace( SOLY, s, t );
```

$$SOLy := \frac{1}{5}\mathbf{e}^{(-4\,t)}\,\mathrm{y}(0) - \frac{1}{5}\mathbf{e}^{(-4\,t)}\,\mathrm{D}(y)(0) + \frac{4}{5}\mathbf{e}^{t}\,\mathrm{y}(0) + \frac{1}{5}\mathbf{e}^{t}\,\mathrm{D}(y)(0)$$

or, as follows, in a form that is sometimes more useful.

```
> collect( SOLy, [ y(0), D(y)(0) ] );
```

$$\left(\frac{1}{5}\mathbf{e}^{(-4\,t)} + \frac{4}{5}\mathbf{e}^{t}\right)\mathrm{y}(0) + \left(-\frac{1}{5}\mathbf{e}^{(-4\,t)} + \frac{1}{5}\mathbf{e}^{t}\right)\mathrm{D}(y)(0)$$

Alternatively, a one-step solution is possible with the use of the *method=laplace* optional argument to *dsolve*.

```
> dsolve( ODE, y(t), method=laplace );
```

$$y(t)=\frac{1}{5}e^{(-4t)}\,y(0)-\frac{1}{5}e^{(-4t)}\,D(y)(0)+\frac{4}{5}e^{t}\,y(0)+\frac{1}{5}e^{t}\,D(y)(0)$$

For additional assistance, consult the on-line help for *dsolve*

```
> ?dsolve
```

as well as the help pages for the individual commands

```
> ?inttrans[laplace]
> ?inttrans[invlaplace]
```

```
>
```