

The ODE Architect

A Guide to the ODE Architect Solver Tool

Michael E. Moody
Department of Mathematics
Harvey Mudd College

Documentation for Version 1.0, July, 1998.
Revised, October 1999

ACKNOWLEDGEMENTS

I would like to thank Robert Borrelli, Courtney Coleman, and Joel Miller for their helpful comments and suggestions. The description of how to build a model of a simple pendulum that appears in Appendix B is due to John Lu.

TABLE OF CONTENTS

SECTION	PAGE
1 Introduction	1
2 The ODE Editor	1
3 The ODE Architect menus.....	4
4 The ODE Architect toolbar	8
5 The Input control panels.....	8
6 The ODE Architect output windows	12
7 Numeric output from the ODE Architect	12
8 The ODE Architect graph windows.....	13
9 Creating custom 2D plots	18
10 3D plots	19
11 Exploring and animating graphs.....	20
12 Model-based animations.....	21
13 The ODE Library.....	23
14 The Report Writer	23
15 The ODE Architect Expert Mode	24
16 Concluding remarks	25
Appendix A: Contents of the ODE Library	26
Appendix B: Building a model for a simple pendulum	29
Appendix C: Graphing a function	32
Quick Reference Guide	34

SECTION 1: INTRODUCTION

The ODE Architect Tool is a sophisticated numerical differential equations solver package with a convenient user interface. There are two modes of use for the tool: the **standard** mode and the **expert** mode. This tutorial will describe the standard mode as it exists in the July, 1998 release, Version 1.0.

Once you have installed your software and have started the ODE Architect program, you will note that the screen has six principal regions: the **menu bar** at the far upper right, the **tool bar** just underneath the menu bar, and four quadrants where various text, graphs, and other information is displayed. We will refer to these four areas as the **ODE Editor** (top-left), the **input controls** (bottom-left) and the **output windows** (top-right and bottom-right). Each will be described in turn.

SECTION 2: THE ODE EDITOR

Figure 1 shows a sample ODE Editor window. The window has scroll bars, should your text extend beyond the margins of the window. If a word, sentence or equation appears incomplete, try scrolling the window to see if there is more text. Most files in the ODE Library have extended comments which require scrolling down to read. The ODE Editor window can be enlarged by dragging the frame bar to the right.

In the ODE Editor the user enters differential equations to be solved. Systems of up to ten first-order ODEs can be entered. Higher order differential equations can be solved in the ODE Architect by converting them to an appropriate system of first-order equations. The ODEs can contain parameters, and the values of these parameters are assigned using the ODE Editor. Other functions of the independent variable or the dependent variables also can be defined. The ODE Editor allows the user to incorporate comments to document work, or otherwise record useful information.

The ODE Editor window is a text-editing environment. Think of it as a simple word processor for typing and editing ODEs. Editing commands from the menu bar **Edit** menu, such as cut and paste, search, and so on, are available when in the ODE Editor. When you first run ODE Architect, the ODE Editor window will be blank. To explore an ODE, you must either type in new equations, or load a previously stored equation file. We will now describe how to type in your own ODEs.

Typing comments into your equations file. First note that a pair of slashes, `//`, **comments out** the rest of the line on which they occur: all text after the `//` on that line will be ignored by the solver. Comments that extend over several lines begin with `/*` and end with `*/`. Here is an example:

```
/* This comment keeps on going,  
   and going,  
   and going. */
```

Typing equations, parameters and auxiliary variables. Except for comments, everything in the equation file is interpreted by the Architect as mathematical instructions for defining ODEs, parameters,

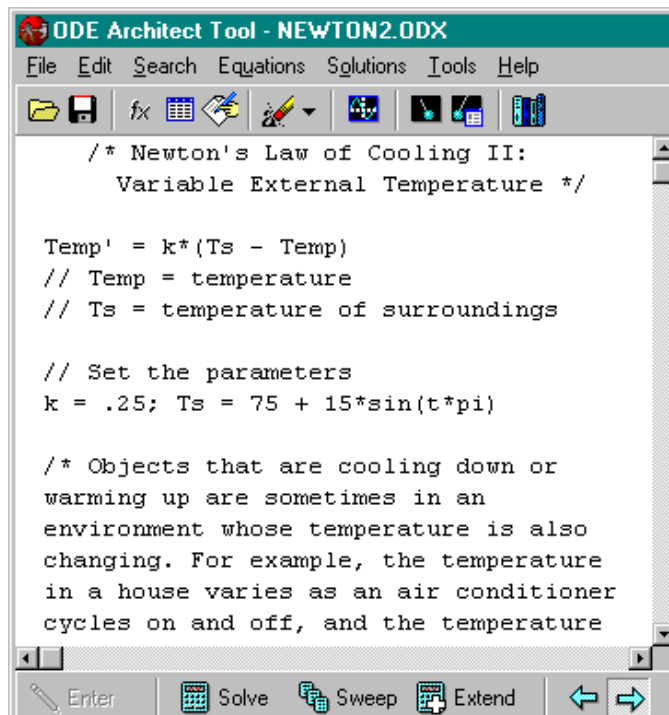


Figure 1: The ODE Editor window.

or auxiliary variables. Entering such definitions requires that you follow a set of rules, or syntax. The first important rule to note is that the arithmetic operations are indicated using $+$, $-$, $*$, $/$, and always must be typed explicitly. The symbol \wedge is used for exponentiation. The standard priorities for executing arithmetic operations applies; to be safe, always use parentheses to make your expressions unambiguous. The following table shows how to convert standard mathematical notation into correct ODE Editor expressions:

MATHEMATICAL EXPRESSION	ODE EDITOR EQUIVALENT
$2x$	<code>2*x</code>
$1 + \frac{a}{2}x$	<code>1 + (a/2)*x</code>
$\sqrt{ab - c \sin^{3/2} hx}$	<code>sqrt(a*b - c*(sin(h*x))^(3/2))</code>
$e^{\pi x}$	<code>exp(pi*x)</code>
$x \bmod(2\pi)$	<code>x mod(2*pi)</code>

Notice that the Architect supports most of the standard built-in functions that are found in a modern programming language, such as C. These functions include trigonometric functions and their inverses, exponential and logarithmic functions, integer and fractional part, and modular arithmetic. For a complete list, with corresponding Architect syntax, consult the Help files under the heading **Standard functions**.

Let's see how first-order ODEs are entered. *Example 1.* Consider the differential equation for Newton's Law of Cooling:

$$T' = k(T_s - T),$$

where T is the temperature of an object, k is a positive constant, and T_s is the temperature of the surroundings. To show features of the ODE Architect, we will suppose that the surrounding temperature is not constant, but varies periodically in time. For example, assume that

$$T_s = 75 + 15 \sin(\pi t).$$

To enter these equations for Newton's Law of Cooling, type the following into the ODE Editor:

```
/* Newton's Law of Cooling */
Temp' = k*(Ts - Temp) /* Always type ODEs FIRST, before other equations */
// Temp = temperature
Ts = 75 + 15*sin(pi*t)
// Ts = temperature of the surroundings
// Set the parameter:
k = 0.25
```

The initial conditions for the ODE are entered later, using the **Input Control Panel**. Note that the equations are typed in a natural way: derivatives are indicated by a "prime" (obtained by typing a left quotation mark) and equations are entered as if writing them on paper. One difference, however, is that all arithmetic operations are explicit, and all function arguments must be enclosed by parentheses. Thus you would always enter `sin(x)` in the ODE editor and not `sin x`, as might appear in your math book. Also note that the independent variable does not explicitly appear. The ODE Editor **always assumes that the independent variable is t** . Because of this convention, if you are using the Architect to solve an ODE that is written with some other letter for the independent variable, you must change the equation and use t . In our example, since t is "reserved" for the independent variable, we named our temperature variable `Temp`, rather than `T`; note that `Temp'` stands for $d\text{Temp}/dt$. It is a useful feature of the Architect that variable names can be descriptive.

There is one parameter in the differential equation (k), and one auxiliary function (T_s). They are entered symbolically in the ODE. The Architect knows that they are not dependent variables because they do not have a prime ($'$). Since the Architect solves ODEs numerically, you must assign values to all the parameters, and expressions to all auxiliary functions, before solving your equation. These assignments are also typed in a natural way, such as $k = 0.25$ and $T_s = 75 + 15*\sin(\pi*t)$. It is very useful in many applications of ODEs to be able to define an auxiliary function of the dependent or independent variables, such as our T_s function. These parameters and functions can be used in the differential equation, and they can be plotted in the graph windows.

Entering your equations. When you have finished typing your equations, click the mouse pointer on the **Enter** button in the toolbar just under the ODE Editor window. The equations will then be entered in the solver program of the ODE Architect. If you should forget to assign a value to a parameter or expression, the Architect will display an error message when you attempt to **Enter** them. You are now ready to begin exploring solutions to the ODEs.

Solving higher order equations and systems. The ODE Architect only solves first order equations or systems of first order equations. Higher order equations must be converted to a system of equations, which we will describe in our second example.

Example 2. Consider the differential equation for the motion of a nonlinear, damped pendulum:

$$x'' + bx' + a \sin(x) = 0,$$

where the mass of the pendulum is 1, x is the angular displacement, a is a parameter determined by the acceleration due to gravity and the length of the pendulum, and b is the damping coefficient. The Architect cannot solve this equation as it is written. To prepare the problem for the Architect, we convert the second-order equation displayed above to a system of first-order equations. If we let $y = x'$, then $y' = x''$ and the second order equation is then equivalent to the system

$$x' = y, \quad y' = -a \sin(x) - by,$$

where y is the angular velocity of the pendulum. This procedure can be used to convert an n^{th} -order equation to a system of n first-order equations. The energy E of this pendulum of mass 1 is an auxiliary variable that is physically important, where

$$E = \frac{1}{2}y^2 + a(1 - \cos x).$$

Enter the differential equations, parameters and the energy into the Architect using the ODE Editor as follows:

```
/* The Pendulum Equations */
x' = y // x = position
y' = -a*sin(x) - b*y // y = velocity
Energy = .5*y^2 + a*(1 - cos(x))
a = 10; b = 0.5 // b = 0 for undamped
/* Note that the energy function and the parameter values are
inserted after the ODEs */
```

The equations are again entered in a natural way: derivatives are indicated by a prime and the equations are entered as if writing them on paper. Each ODE is entered separately for the system. Notice how we have defined the energy. The energy does not appear in the differential equations, but

is a function of the dependent (or state) variables x and y . As noted previously, the ODE Architect allows you to define auxiliary variables; they may involve the dependent variables, the independent variable, or any other variables that you have defined.

We will note here that auxiliary variables can be defined *implicitly* in the ODE Architect. For example, we could have written our definition of the energy function this way:

$$.5*y^2 = \text{Energy} - a*(1 - \cos(x))$$

The Architect is able to determine the values of **Energy** even though it is not explicitly defined.

Once again note that the name of a variable does not have to be only one letter. The word **Energy** is used in our example to make it easy to remember the meaning of that function's definition. Notice from the example that more than one assignment can appear on a given line in the ODE Editor, as long as they are separated by a semicolon (e.g., $a = 10; b = 0.5$). Look at the example once more. Several of the lines end with comments. These comments are not necessary for the ODE Architect to solve the system. They are, though, a useful way to keep track of what you have done, so that if you should return to your work in the future, it will be easier to recall what you were doing.

The Engineering Functions The ODE Architect supports a variety of built-in functions that are commonly used in engineering, such as step functions, square waves and triangular waves. These functions are used to model “on-off” or other forms of “switching” behavior. The syntax for using these functions is documented in detail (with accompanying graphs) in the on-line help, listed under “engineering functions.” For example, to produce a square wave function y of time with a period of 50 time units, and which has a value of 1 for the first 20 units of each period and the value 0 for the other 30, you would enter $y = \text{SqWave}(t, 50, 20)$ (Figure 2). The other engineering functions have a similar form.

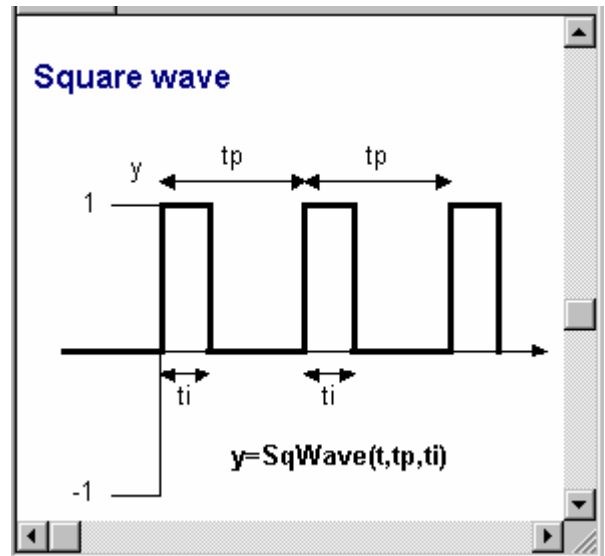


Figure 2: The Square Wave engineering function.

SECTION 3: THE ODE ARCHITECT MENUS

The Architect has seven principal menus: **File**, **Edit**, **Search**, **Equations**, **Solutions**, **Tools** and **Help** (Figure 3). Many functions controlled from these menus are obvious, and will not be explained in detail. For example, the **Edit** and **Search** menus offer simple features found in most word processing programs.

The File Menu. A sample **File** menu is shown in Figure 4. Four of the first five selections (**New**, **Open**, **Save**, **Save As**) function just like their counterparts in other commonly used programs. The ODE Architect allows you to save your equations into **equation files**. These files must have the extension **.odx**. From the **File** menu you can create new equation files, open old ones, and save your work. The last four equation files that you have opened are listed at the end of the **File** menu and can be reopened by selecting them. Warning: If you modify the mathematical content of an active equation file, such as by changing the value of a parameter, the previously computed solutions and their graphs will not correspond to the updated ODEs.



Figure 3: The ODE Architect menus.

The **Library** selection will open the ODE Library, which also can be opened using the Toolbar (discussed below). The ODE Library has many examples that span most of the common topics in a first course in differential equations. When you open a Library file, you will see commented equations in the equation editor, a short description of the application, and some graphs. If you examine several of these Library files, you will quickly see how to enter a variety of different types of equations. A complete list of Library files appears at the end of this document. Click **Open movie** to play stored avi files.

The **Expert Mode** selection from the **File** menu will open a version of the ODE Architect for the advanced user. The expert mode offers more features and control, but does not have a graphical user interface. For example, in **Expert** mode, the user can open several graph windows simultaneously and place them anywhere on the screen. Use the expert mode only after you have become familiar with the standard mode. See Section 15 for more discussion of the **Expert Mode**.

The Equations menu. The first selection in the **Equations** menu (Figure 5) provides an alternate way to **Enter** your ODEs; it also shows that the F4 key is a shortcut for entering equations. The last two items are more advanced features.

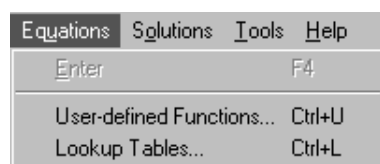


Figure 5: The **Equations** menu.

The **User-Defined Functions** menu item opens a window that allows the user to create custom functions using a scripting language (Figure 6). User-Defined functions (UDFs) are sub-programs written by the user to perform calculations. They can be used in many situations, including those which need calculations requiring logic (IF-THEN-ELSE statements) or iteration (FOR-DO looping), and by replacing a set of statements used several times in the equations with one statement. From the ODE Editor, UDF calls are included as an equation: `var = fname(par1, par2, ..., parN)`. The UDF with name `fname` performs a calculation using the values of the parameters `par1, par2, ..., parN` that are passed to it via the function call. The UDF returns a value that is assigned to the variable `var`. Here is an example of a UDF that computes the maximum of two numbers:

```
function max(a,b)
  if (a > b) then
    return a
  else
    return b
  endif
end
```

To use this UDF in your equations, you would type something like this: `z = max(x, y)`; the variable `z` would then be the maximum of variables `x` and `y`. Consult the on-line help files for more detailed information about UDFs.

Lookup tables. The last menu selection in the **Equations** menu opens the **Lookup Tables** control panel. The user can input numerical data to



Figure 4: The **File** menu.

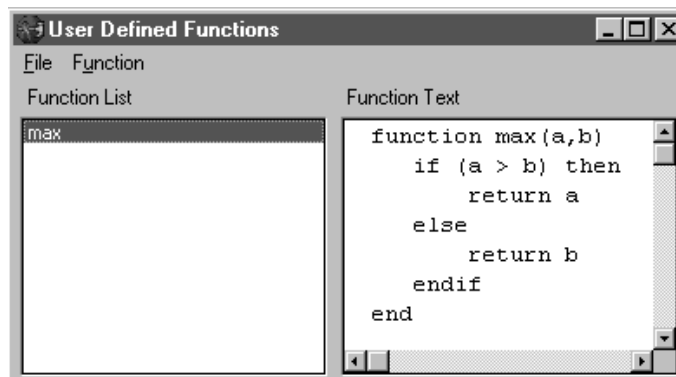


Figure 6: The **UDF Control Panel**.

the Architect using Lookup Tables (see Figure 7).

	1	2
1	1941	0.68
2	1942	0.57
3	1943	0.57
4	1944	0.61
5	1945	0.44
6	1946	0.25
7	1947	0.13
8	1948	0.19

Figure 7: A Look-up Table.

graphs that you produce in the Architect can be pasted from the graph windows into the report writer. Cutting and pasting graphs is explained below along with other graph editing features. This version of the **Report Writer** does not support an equation editor; you should therefore type equations into your report in the same way that you would type them into the **ODE Editor** of the Architect. We also note that you may use other word-processing environments to create your reports. If your word-processor can paste from the Windows Clipboard, then you can cut and paste your Architect graphs into that program. A more full-featured word processor will also have better mathematical equation editing features that you may find useful.

The **Model Builder** is a program used to create model-based animations. Models can be constructed using various geometric objects, such as polygons, ellipses, lines and text (see Figure 9). The properties of these objects, such as position, scale, rotation and so on, can be dynamically linked to variables that appear in a system of ODEs. Animation “scripts” can also be written to control more complex interactions between the model and ODEs. The resulting model image will then be animated in response to solving the ODEs. Many of the Architect Library files have animations associated with them; the animations for the Multimedia ODE Architect were also produced using the Model Builder.

Consult Appendix B of this manual for more discussion and an example of building a model of a simple pendulum. Section 12 describes how to link or connect a model to a set of differential equations, and how to save model animations as movie files to replay outside the Architect program. Detailed instructions for the Model Builder are also found in its on-line help facility. If you like programming computers, you may enjoy using this tool to create animations for ODEs or applications that interest you.

The **Discrete Tool** is a separate program for exploring discrete dynamical systems. With the **Discrete Tool** (Figure 10), the user can iterate one-dimensional maps that have one parameter and two-dimensional maps with two parameters. For a pre-defined collection of discrete complex maps, the Julia and Mandelbrot sets can be explored. The **Discrete Tool** complements the ODE tools in the Architect, and provides a tool for analyzing discrete dynamical systems, drawing bifurcation diagrams, and exploring complex chaotic dynamics.

This feature is documented in the on-line help and by explicit example in Module 1 of the Multimedia ODE Architect. As suggested by Figure 7, the user can directly enter a table of numerical data, or can import that data from outside the Architect.

There are three built-in functions for referring to the data in the table: **LOOKUP**, **LOOKUPVAL** and **LOOKUPLIN**, which return, respectively, exact table values, one-dimensional interpolated values, and two-dimensional interpolated values.

The Tools menu. There are three additional programs (“tools”) available on this menu: the Model Builder, the Report Writer, and the Discrete Tool (Figure 8). Selecting the **Report Writer** will open the Windows Word Pad application. You may use the **Report Writer** to type your lab report. Any

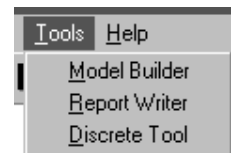


Figure 8: The Tools menu.

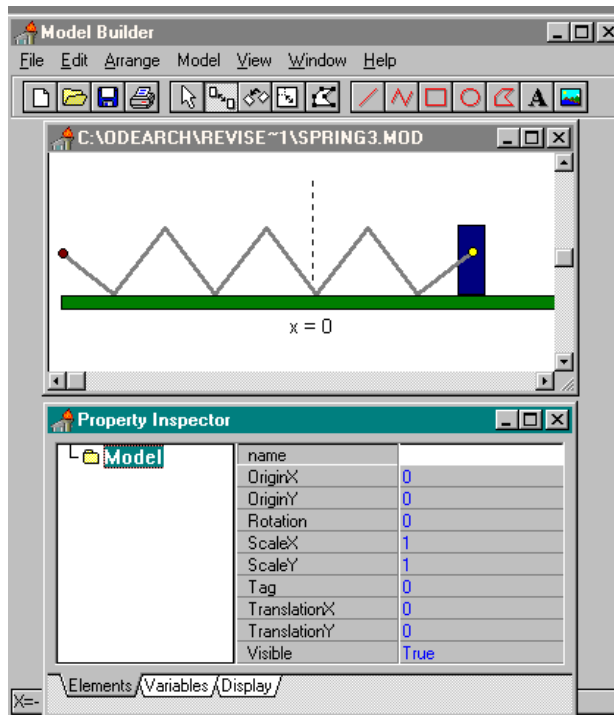


Figure 9: The Model Builder.

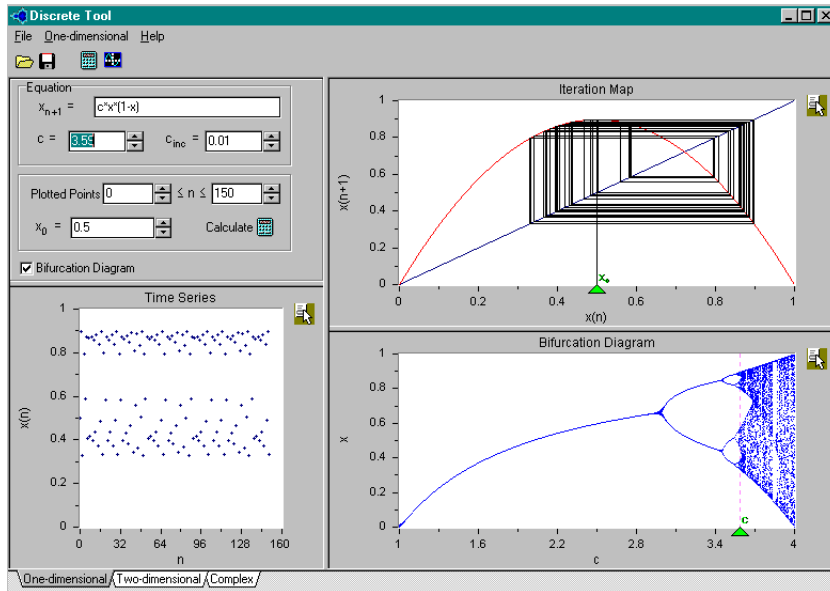


Figure 10: The Discrete Tool screen.

The Solutions menu. The selections in this menu (Figure 11) control the numerical solver and allow the user to delete, explore, or animate solutions; keyboard shortcuts are indicated for most of these functions. The **Solutions** menu collects in one place several functions that are normally accessed from other parts of the Architect. Rather than explain them at this point, they will be discussed in Section 5, where the details about solving your ODEs are presented.

Solutions	Tools	Help
Solve		F5
Sweep		F6
Extend Current		F7
✓ Forward Direction		Ctrl+F
Backward Direction		Ctrl+B
Clear Last Solve		
Clear Current Run		
Clear All Runs		
Animate...		F9
Explore		F10

Figure 11: The **Solutions** menu.

SECTION 4: THE ODE ARCHITECT TOOLBAR

The Architect Toolbar is a row of icons below the main menu list (Figure 12). They provide quick access to many features of the Architect; details of the individual features will be explained in the relevant sections in this guide. At this point, we will simply give an overview of the Toolbar functions.

The first two buttons on the Toolbar provide, respectively, shortcuts for opening and saving equation files; they substitute for **File** menu operations. The third button opens the user-defined functions window, and the fourth button is a shortcut to the lookup table feature. The next button (the fifth) sends the user to the Report Writer.

The Toolbar button with the pencil eraser icon pops up a menu with controls for deleting the solutions you have computed. Moving across the Toolbar, the next button opens the **Explore** tool, which allows the user to examine solutions in detail, with all numerical information displayed in a table. The next two Toolbar buttons control the animation features of the Architect, and the last button on the Toolbar opens the ODE Library, which is an extensive collection of ODE examples and applications.



Figure 12: The ODE Architect Toolbar.

SECTION 5: THE INPUT CONTROL PANELS

There are four input control panels that are accessible by clicking on their respective tabs at the bottom of the window. They are the **IC**, **Sweep**, **Solver** and **Equilibrium** control panels.

The IC Panel. Initial conditions and integration parameters for solving an ODE are set in the IC Control Panel (Figure 13). It is a four-step process to obtain a solution curve for the ODE. First, set the **Integration** parameters; second, enter the **Initial Conditions**; third, set the **Solve Direction** (arrows at the top of the IC panel); fourth, press the **Solve** button.

On the right of the IC Control Panel are the **Integration** parameter settings. You must specify an interval, which will be the length of time over which the solver will compute solutions each time the **Solve** button is clicked. How long this time should be will depend on the ODEs; you must be careful when setting the interval. Many users choose a short interval if they think the solutions might become huge very quickly. Very often, the physical problem itself provides insight for a good solve interval; this applies to the settings in Figure 7,

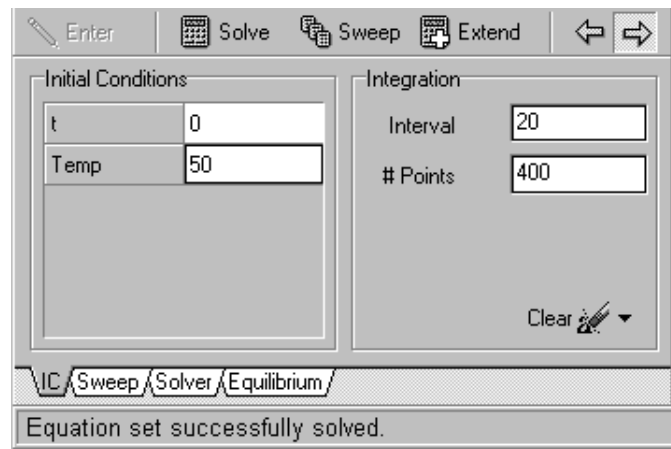


Figure 13: The Initial Conditions (IC) control panel.

for the Newton’s Law of Cooling example. In any case, just make your best guess and try it! You will know from the graph of the solution whether you should change the interval.

In addition to the integration interval, you must also specify the number of points that the Architect should plot on the graph. This number is **not the number of points at which the solution is calculated**. The numeric solution algorithms are adaptive and decide for themselves how many points, and at what locations, the solution is computed. The number of points, evenly spaced in time, at which the solution should be **plotted**, is under your control. Set this number in the **#Points** field; a value between 250 and 500 usually works well. For solution curves that extend over large intervals of time, or for curves that change rapidly, a smoother graph can often be obtained by increasing the number of points plotted. The number of points plotted should be less than 5,000.

Now that you have set the integration parameters, it is time to set the initial conditions (IC) for the ODEs. There are two ways to set the ICs. They can be typed directly into the input fields of the table of variables on the left side of the IC panel. There will be a field corresponding to t and fields for all dependent variables (those with a prime) that you defined in your equations using the Equation Editor.

If the graph window is in **Solve** mode (described below), initial conditions are supplied to the solver by clicking in the graph window. Each time the user clicks in the graph window, a new solution is computed and graphed using the position of the “click” as the IC. Notice that when the cursor is in an active (two-dimensional) graph window, its coordinates are recorded as a pair of numbers at the far bottom-right of the Architect screen. This information can be helpful when you input graphical ICs or when deciding what ICs to type in directly.

Now that you have entered your ICs and set the Integration parameters, you must decide which direction to solve. The **Solve Direction Arrows** which appear in the bar just above the IC panel set these directions. If the right arrow at the top of the IC panel is pressed, solutions will be computed forward in time from the IC; if the left arrow is pressed, they will be computed backward in time from the initial time. Note that changing the solve direction does not “undo” the previous solve.

Now that you have entered the ICs and set the integration parameters and the solve direction, just press the **Solve** button that appears above the IC panel (Figure 14). The Architect will attempt to solve your system of ODEs for one integration (or solve) interval in the indicated direction. As the Architect computes the solution, a message box appears alerting you to the solver’s progress. Should the solver encounter problems, you will be able to abort the calculation from this dialog box.

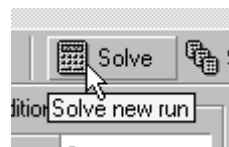


Figure 14: The **Solve** button.

You should also be aware that some ODEs are very “nice” when solved forward in one direction, but are very sensitive to the solve interval setting when solved in the other direction. For example, the simple ODE $x' = -x$ with IC $x(0) = 1$ has the solution $x(t) = e^{-t}$. This solution decays to zero as t increases forward in time, but grows exponentially fast as t decreases backward in time; thus, a large value for the **Interval** will work for forward solutions, but the solution would then “blow up” when computed backward.

You may use **keyboard shortcuts** for the **Solve Direction** arrows: control-F sets the forward direction and control-B sets the backward direction. Pressing F5 is equivalent to clicking the **Solve** button. These shortcuts are especially useful when viewing a solution graph that has been maximized to occupy the entire computer screen, thereby obscuring the controls.

Extending solutions, runs, and the current run. It is important to understand how the Architect keeps track of your work. Suppose you have entered a new IC for a given ODE, set the solve direction, and pressed **Solve**. The Architect will compute the solution, and the results for the basis for the **current run**. Suppose you want to see the solution for another increment of time; to do so, press the

Extend button, which is also on the bar just above the IC panel. The Architect will then extend the solution for another integration interval in the direction that the **Solve Direction Arrow** indicates. The solution can be repeatedly extended, and you can change the solve direction as often as you want. The Architect joins the results into a single solution, called a **run**. When you change the IC and solve again, you are creating a new run. As each run is computed, the Architect keeps a complete record of them, which can be viewed under the **Data** tab on the graph window (described below). For purposes of selecting solution curves for inspection, or deleting solutions, it is important to distinguish between the results of the most recent solve, the current run, and other runs you may have computed.

Deleting results. On the lower right of the IC panel is a small pencil-eraser icon marked **Clear**. Pressing this button pops-up the menu shown in Figure 15. The pencil-eraser icon on the tool bar does the same thing.

The user can clear the result of the immediately previous **Solve**; this function will erase the extensions to a run. If you have extended a solution several times, the extensions can be cleared, one at a time, in the reverse order they were created by repeatedly pressing **Clear Last Solve**. The entire **current run** can be deleted by pressing the designated key, and all computed runs can be erased at once (be careful with that!).

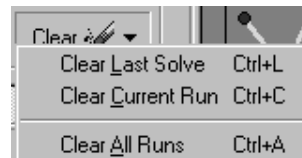


Figure 15: Menu for deleting solutions.

The Solver Panel. The **Solver** tab leads to the controls for selecting the solver method (Figure 16). The solver routines of the ODE Architect were designed by Professor Larry Shampine of Southern Methodist University. It is a research-quality program, capable of solving a wide variety of differential equations. The default is the Runge-Kutta (4, 5) Pair method; it usually will work well. For some ODEs you may have to change the method to obtain accurate solutions. For example, the ODEs for the so-called Oregonator chemical reaction (see the ODE Library under Chemical Models) are “stiff” and the BDF method of solution works best.

You will have to experiment with these different methods—if RK does not work, then try another method. The **Solver** panel also allows the experienced user to override the solver’s internal setting for the maximum time step that it will take while computing a numerical solution.

Lastly, we point out that the Euler method is provided mainly for instructional purposes, and is not recommended for solving most initial value problems.

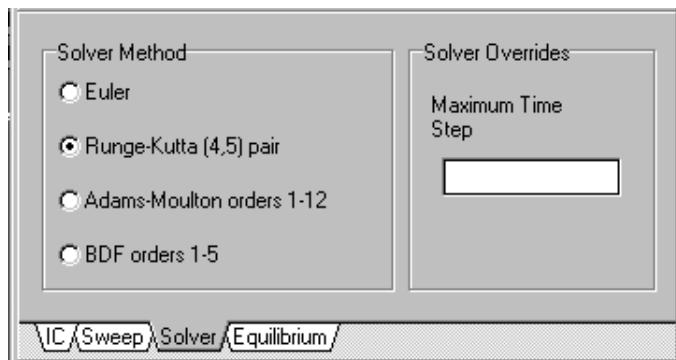


Figure 16: The Solver Control Panel.

The Sweep control panel. Many differential equations are written with parameters, as the following example shows:

Example 3. The ODE

$$n' = an \left(1 - \frac{n}{100} \right),$$

has one parameter, a . Of course, to solve the ODEs numerically in any instance requires that the parameters have been set to specific values. When studying ODEs, we are often interested in how the solution curves change, *for a given initial condition*, as the parameters are varied.

Sweeping a parameter means sequentially varying the value of that parameter, starting at an initial value and ending at a prescribed value, with a fixed increment between successive values.

Suppose we want to “sweep” the parameter a in the above example equation. Select the **Sweep** tab to bring up the Sweep Control Panel (Figure 17).

Look at the middle set of controls, marked Sweep 1. Since we are sweeping only one parameter, we use only these controls. Notice the text field at the left with a pull-down menu button and the word `<None>`. This box displays the name of the first variable to be swept. Before we have chosen a variable, the box reads `<None>`. Press the pull-down menu button: a list of all the available variables will be displayed (a and n in this case). Select from the list the variable that you want to sweep. Next, enter a starting value for the variable in the **Start** field, and an ending value for the variable in the **Stop** field. Finally, enter the total number of values of the parameter that you want the solver to sweep in the **#Points** field. For example, if a is your sweep variable, and you selected **Start** 0.5, **Stop** 2.5, **#Points** 5, then the solver will solve the equations a total of five times, using the current IC setting on the IC control panel. The five values of a are evenly spaced starting at $a = 0.5$ and ending at $a = 2.5$, or $a = 0.5, 1.0, 1.5, 2.0, 2.5$. Then press **Sweep** on the bar at the top of the output control panel. After computing the five solutions, the Architect will automatically graph them all on the same set of axes.

If you select a dependent variable as a sweep variable, the Architect **assumes that you want to sweep the initial value of that variable**. This feature is useful when drawing phase portraits or other situations where multiple solution curves corresponding to different ICs are desired. For example, to draw curves corresponding to values of $n(0) = 10, 20, 30, 40,$ and 50 , select n as the sweep variable, set **Start** to 10, **Stop** to 50, and sweep 5 points.

Lastly, we point out that it is possible to sweep two different variables simultaneously. When two different variables are swept, they may be swept independently through all pairwise combinations, called a **Dual (Matrix)** sweep. If **Dual (Linear)** is selected, the first value for **sweep variable 1** is paired with the first value for **sweep variable 2**, and a solution curve with these values for the variables is plotted. The second value for **sweep variable 1** is paired with the second value for **sweep variable 2** and a new solution curve is plotted. This pattern continues until all the plots have been made.

The Equilibrium input control panel. The ODE Architect will numerically estimate equilibrium points for autonomous systems. Press the **Equilibrium** tab on the **Input Control Panels** to access this feature. To locate an equilibrium, the Architect needs an initial guess for its location. You must enter your guess as values of all the dependent variables. Once you have done so, press the **Calculate** button on the panel. The Architect will proceed from your guess to find a nearby equilibrium point. The results are displayed under the **Equilibrium** tab of the output windows on the right half of the Architect screen. These outputs will be described in more detail below.

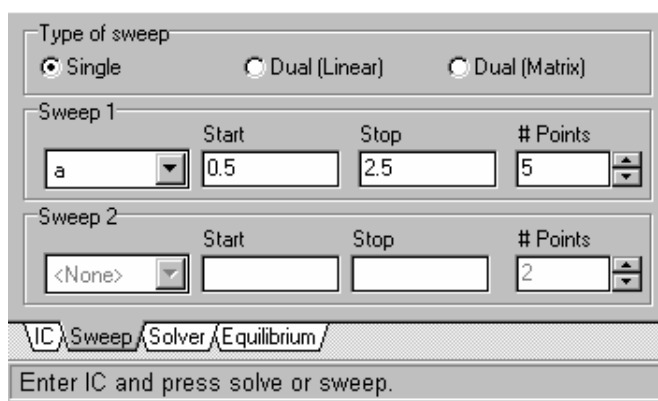


Figure 17: The Sweep control panel.

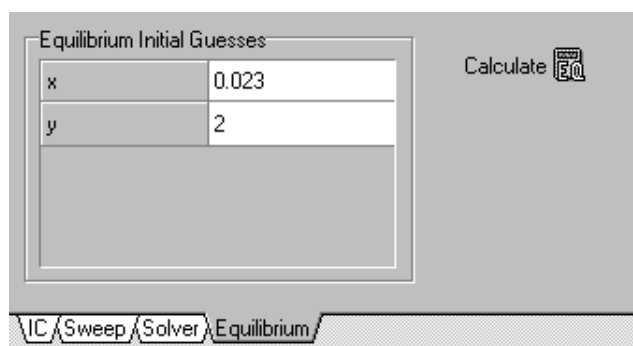


Figure 18: The Equilibrium control panel.

SECTION 6: THE ODE ARCHITECT OUTPUT WINDOWS

The right half of the ODE Architect is devoted to outputs. These outputs consist of graphics, numerical data, and model-based animations. The size of these windows can be adjusted by dragging the edges of the window frames to any desired position. The tabs on the output windows are used to quickly change between various graph windows, the equilibrium output window, the solution data window, and the model window. We will describe these various windows in turn, beginning with the numeric outputs.

SECTION 7: NUMERIC OUTPUT FROM THE ODE ARCHITECT

There are two numerical output windows: the **Equilibrium output** and the **Data output** windows.

The Equilibrium output panel. As the Architect computes equilibrium points, the results are recorded in a table that appears under the **Equilibrium** tab in the output panels. This window is shown in Figure 19. Each equilibrium is assigned a reference number, and the results of the calculation are indicated to the right. Sometimes the Architect is unsuccessful in locating an equilibrium point based on your initial guess.

	x	y	Results
1	0.5	0.5	Successful
2	1	0	Successful
3	0	0	Successful
4	0	2	Successful

t-x / t-y / x-y / 2D / 3D / Equil / Data / Model /

Figure 19: The equilibrium output window.

If you click on the menu icon that appears on the right of the **Equilibrium** output window (Figure 20, or right-click on the window), a menu of controls appears. Using these controls you can set how much information is displayed in the output window and you can delete selected equilibrium records or clear all of them.

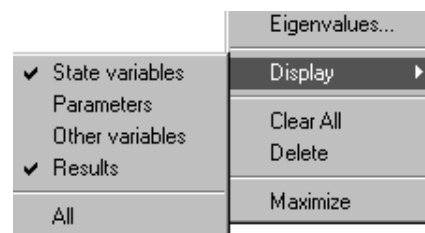


Figure 20: The equilibrium output menu.

Finally, the Architect will numerically approximate the eigenvalues and eigenvectors of the Jacobian matrix corresponding to each of the equilibrium points for an autonomous system. If the system is planar, the Architect will (usually) classify the equilibrium point as to stability and type (*e.g.*, attracting node). To obtain this information, click on the desired equilibrium point record to select it, and then select the **Eigenvalues** item from the menu.

Eigenvalues and Eigenvectors			
		1	2
EigenValues	Saddle	-0.7844	0.1594
	Equilibrium	EigenVector	EigenVector
x	0.5	-0.8693	0.6042
y	0.5	-0.4944	-0.7968
	Jacobian		
x		-0.5	-0.5
y		-0.375	-0.125

Figure 21: Eigenvalues and eigenvectors of the Jacobian, and classification of an equilibrium point.

The information will be displayed in a new window (Figure 21). You will have to examine this data for each equilibrium point individually. As you can see from the figure, the information is tabular. The “*x*” and “*y*” coordinates of the equilibrium point are shown; your variable names may be different from *x* and *y* and the table will show the correct names. The approximate Jacobian matrix for the equilibrium point is represented in the lower-right 2×2 array of entries in the table. The eigenvalues of the matrix are written across the first row of the table.

The corresponding eigenvectors are displayed immediately above the Jacobian, and below the eigenvalues, in column form. Lower-case “i” denotes the imaginary unit. **Warning:** The Architect may misclassify an equilibrium point of a nonlinear, planar, autonomous system, particularly if the eigenvalues of the Jacobian are pure imaginary. In this case, nonlinear terms may determine the nature of the equilibrium point.

The Data output panel. As mentioned previously, the Architect records all information about the solution to all runs. This information is displayed in the **Data** panel of the output windows (Figure 22).

The values of t , the dependent variables, the auxiliary variables, and all parameters are displayed for each point that appears on the graph of a given solution curve. A menu allows the user to select how much of this information is displayed in the window.

Notice the series of numbered tabs at the bottom of the **Data** window. Each tab corresponds to the data for a given solution run. By clicking on a tab, you can view the data for a run; that run then is “active” and you can extend the solution, reverse the solve direction, and so on for the run. The data for each run is arranged in order of increasing time. The active tab also corresponds to the **current run**.

	Temp	Ts	t	Temp'
IC	50	75	0	
1	50	75	0	
2	50.5346	78.745	0.0803213	
3	51.131	82.2528	0.160643	
4	51.781	85.3013	0.240964	
5	52.4725	87.6973	0.321285	
6	53.1901	89.2891	0.401606	
7	53.9161	89.9758	0.481928	

Figure 22: The data output panel.

SECTION 8: THE ODE ARCHITECT GRAPH WINDOWS

There are several graph windows in the output panels. They can be resized by dragging the frames to any desired position. The tabs on the upper set of graph panels and the lower set are (almost) independent of each other. This independence makes it easy to compare several different plots to each other. A typical graph window is displayed in Figure 23, which shows orbits (or trajectories) for the damped pendulum equations in Example 2. The curves in the figure correspond to sweeping the initial value of y from 10 to 20 with the initial value of x kept fixed at 0.

The tabs on the graph windows are used to quickly change between a standard set of plots. There is a tab for plotting each dependent variable against t , the so-called **component plots**. If there are two or more dependent variables, there will also be a tab to graph a parametric “phase plot” of the first two dependent variables in your system of differential equations (see Figure 23).

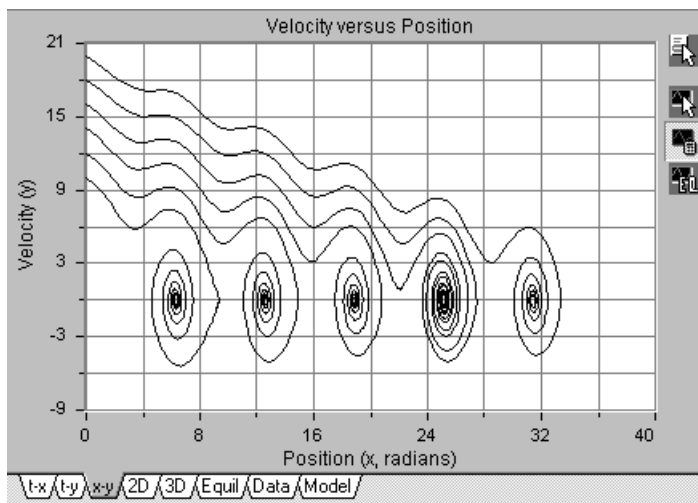


Figure 23: A typical graph window.

The 2D and 3D tabs are for special plotting purposes. The 2D tab on each graph window is used

to plot user-selected combinations of variables, or auxiliary variables. This feature allows the user to customize two-dimensional plots, such as plotting all dependent variables versus t on the same axes. Lastly, the 3D tab selects the three-dimensional plot window. Although this tab is always present, it will be most useful when the system of ODEs is at least two dimensional.

Now that you are oriented to the kind of graphs that are available, we will discuss in more detail how you can manipulate the graphs, and how to manage the graphical information in these output windows. We will begin by describing the **Graph Mode** controls.

The Graph Mode Controls. The buttons along the right side of the graph windows are the **Graph Mode** controls (Figure 24). When you have produced a graph, you may want to modify it or do further analysis.

For example, you may want to locate equilibrium points. When the bottom of these buttons is pressed (the one with the EQ on the icon), the graph window is in **Equilibrium Mode**. All clicks in the window are interpreted as initial guesses and passed to the **Equilibrium Tool**. When a mode button is pressed, its background changes to let you know which mode is active.

If the button with the small calculator icon is pressed (just above the **Equilibrium Mode** button), the Architect is in **Solve Mode**. When in **Solve Mode**, all mouse clicks in the graph window are interpreted as inputs to the solver, and are passed as initial conditions. Clicking on the graph in this mode, then, generates new solution runs. One still can enter ICs by hand in the **IC Control Panel**. Both methods of entering ICs work together.



Figure 24: Graph Mode controls.

Suppose you have produced a graph with several plots on it, and you would like to recall the details of a particular plot (or run). To pick out a particular plot from several, click the **Selection Mode** button, which has the small arrow pointing to a graph on its icon (just above the **Solve Mode** button). When in **Selection Mode**, clicks in the graph window are interpreted as attempts to select a plot to become the new **current run**. If you click exactly on any portion of a particular solution curve, that curve will become the new current run. If you do not click squarely on a plotted curve, the Architect will do its best to select the nearest run to that point, although sometimes it will make mistakes. It is best to click on or very close to the curve that you want to select. Once you have selected a run, you can view the data for that run, or delete it, or extend it—whatever you want to do.

The top button on the **Graph Mode Controls** has a small arrow pointing to a menu. Clicking on this button places the graph window into **Plot Control Mode**, and produces a menu of options (Figure 25). When in **Plot Control Mode**, the user can control most aspects of the appearance of the graph: direction field, color, line style, line thickness, data markers, text and symbol markers, axis scales, zoom, graph window size, copy graph, print graph and so on. A right button click on the graph window itself will also summon the **Plot Control Mode** menus. We will now discuss in more detail the various ways you can customize your plots and edit their appearance.

Several controls on the **Plot Control Mode** menu work in obvious ways. Select the **Maximize** control to fill the entire computer screen with a graph. To return the graph to its original size, click the right mouse button on the expanded graph to call up the **Plot Control Mode** menu and use the **Restore size** option, which has replaced **Maximize**. To

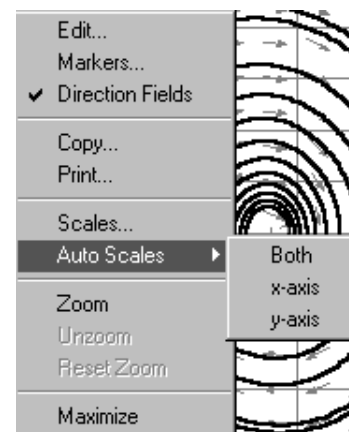


Figure 25: The Plot Control Mode menu.

“zoom in” on a portion of the graph, use the **Zoom** control: the mouse pointer will change to a magnifying glass icon. Click on the graph to do a standard zoom on the surrounding region. Alternatively, hold down the left button on the mouse and drag a box around a region of the graph window to zoom in on precisely that region. You may use **Zoom** repeatedly. Use the **Unzoom** option to return to the immediately previous zoom level, and use the **Reset Zoom** control to return directly to the original scale.

The **Copy** control will copy the graph to the Windows clipboard; you can select from several sizes, and the graph can then be pasted into your document (see Figure 48). In this release of the Architect, only 2D graphs can be copied to the Windows clipboard. The **Print** command will send the graph to the currently selected printer on your computer system. Clicking on **Print** will print the graph that is in the active window (Figure 26). Depending on how your computer is configured, you may be able to **Print** the graph to a postscript file, or to a PDF file, so that the image can be incorporated into other applications or documents. The ability to print to files will depend on the user having installed appropriate software (such as Adobe Acrobat for PDF), and is not part of the Architect.

The other menu options, **Edit...**, **Markers**, **Direction Fields**, **Scales**, and **Auto-Scales**, will be explained in more detail.

Drawing a Direction Field. For a planar autonomous system, or a first-order equation, the Architect will draw a direction field on the appropriate graph window. Select this option from the **Plot Control Mode** menu. If it is inappropriate to draw a direction field, such as for a non-autonomous system, the **Direction Field** option will be dimmed on the menu. The appearance of the direction field can be adjusted using the **Edit** control that appears on the **Plot Control Mode** menu. When the direction field tab is selected on the **Edit...** menu, controls appear for setting the density of direction field segments on the plot, the shape of the segments, their color, and so on. The control panel for adjusting the properties of the direction field is shown in Figure 27.

The Edit... Control. When the **Edit...** control is selected, the **Edit Graph** window is opened (see Figure 28). When this window opens, the **Plot** tab is active. From the left half of the **Plot** tab panel, you can control the color of the curves, their width and their line style. Changing the settings in this window will affect all curves in the graph window. The right half of the **Plot** panel controls the placement of symbols, called **Data Markers**, at regular time points on a plot. There are three controls. The **Style** control is a menu of data marker shapes. The default data marker style is **<None>**: there are no data markers. Select one of these symbols for your data marker. Now use the **Size** control and set a size for the symbol. In the **Every** field, set a frequency with

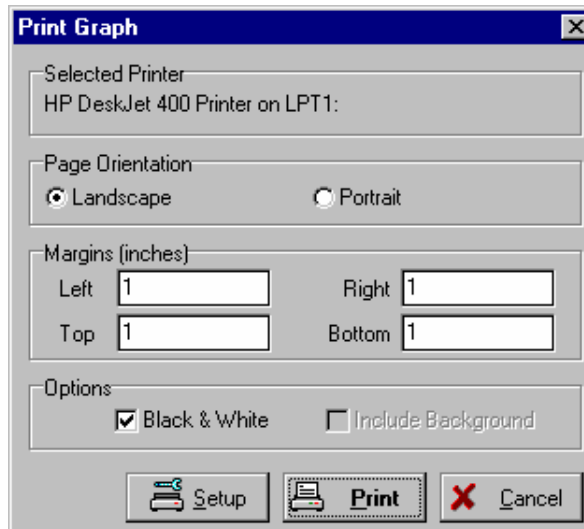


Figure 26: Printing graphs from the Architect.

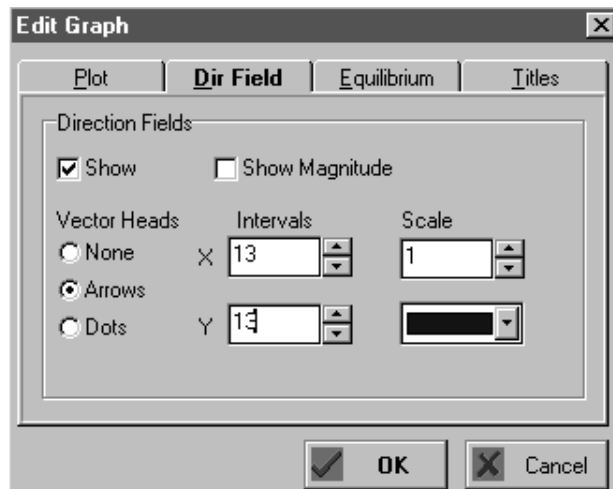


Figure 27: Editing a direction field.

which the Architect will place a data marker on a solution graph.

For example, if your graph has 100 points, and you set a data marker every 9 points, then there will be 12 data markers on the final graph evenly spaced in time. The starting point (always) and the end point (often) of a graph have a data marker.

Data markers are particularly useful for phase plots, in that their spacing conveys information about the speed of the system along an orbit. The orbit is traced faster in those places where the markers are farther apart. Figure 29 shows a plot with data markers of an orbit of the pendulum equations (Example 2), with $x(0) = 0, y(0) = 10$. To remove the data markers, reset the data marker style to <None>.

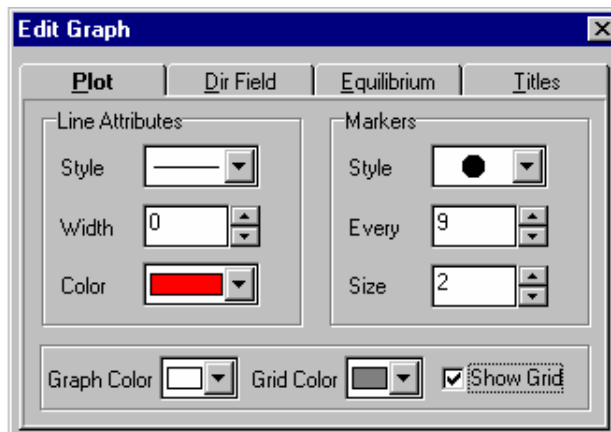


Figure 28: Editing the default graphs.

Sometimes it is desirable to plot a graph using only data markers—a graph without the plotted points interpolated with line segments. To obtain such a plot, simply select the line style for the plot to <None> (the first style in the line attributes list), and choose a data marker symbol to plot, as described above. The graph will now be plotted only with data markers. You can adjust the frequency with which the markers are plotted to get a nice looking graph.

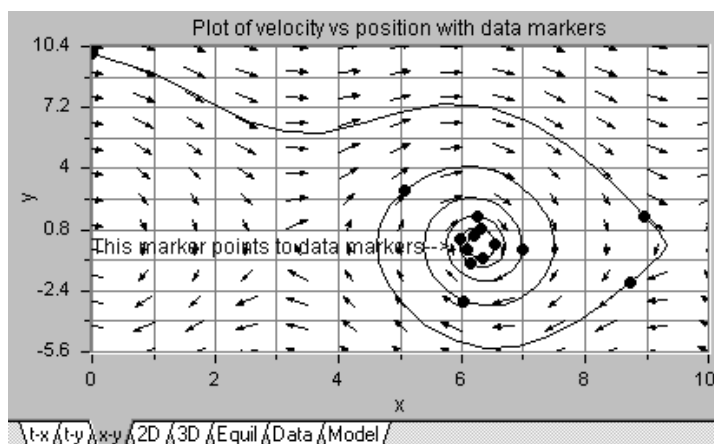


Figure 29: A plot with data markers and direction field.

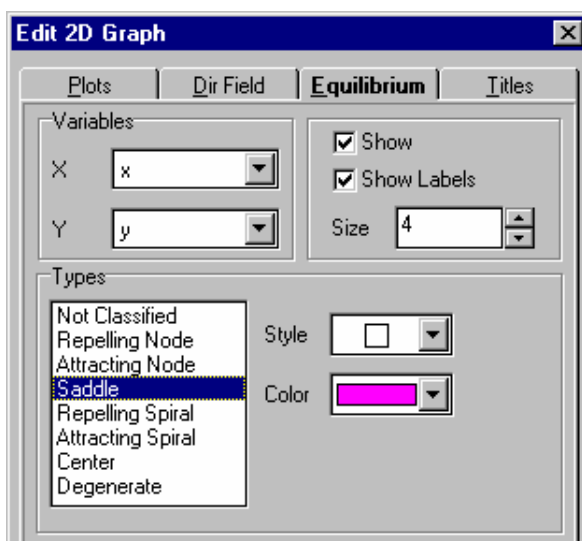


Figure 30: Editing the equilibrium markers.

The **Graph Color** button on the **Plot** tab controls the background color (Note: use white for the background color if the graph is to be pasted into a report). The two **Grid** buttons allow you to put a rectangular grid of any color over the plot (see Figure 23 and Figure 29).

The other three tabs on the **Edit Graph** menu open control panels to edit the direction field characteristics, the markers for equilibrium points, and to add titles.

For example, you may edit a direction field to change the density of points at which field segments are plotted, change the color, and select the kind of segment that is plotted, such as an arrow or a plain line (Figure 27). You may also choose to scale the length of the direction field segments to be proportional to their slope.

The Architect has a default set of symbols to label various equilibrium points; these symbols and their colors can be changed by the user using the control panel shown in Figure 30.

Lastly, the **Titles** panel gives the user control over the labels on the axes and title of the plot (Figure 31). These labels and title are entered into the appropriate text fields. Note that on this control panel “X-axis” and “Y-axis” refer to the labels on the horizontal and vertical axes, respectively. Your actual variable names may be different from X and Y .

The Scales Control. Unless you override the default, the Architect will **Auto Scale** your plots. After the solver has computed the solution, a scale is chosen so that the entire plot is visible. If you want to specify a **custom** scale for a plot, select the **Scales** control on the Graphic Control menu. Doing so will open the **Graph Scales** window (Figure 32). It is a tabbed window, with tab settings for the various dependent variables that your ODE has defined.

To change the scale, select the appropriate variable tab, (X means horizontal, Y means vertical), disable the **Auto Scale** box, and then enter the minimum and maximum values that you want the graph window to display for that variable. Keep in mind that with **Auto Scale** disabled, all or portions of a solution curve might fall **outside** the specified plot window and will not appear on the display.

It is usually a good idea to leave **Auto Scale** as the default, and only change the scales when it is necessary. Warning: If the variables in your solutions become very large over a portion of the solve interval, the scale of the solution curve produced by **Auto Scale** might be a poor one. In such cases, either set your own scale or alter the solver controls to produce a better plot. The **Graph Scales** window also has controls for modifying the placement of tick marks and labeling on the axes. Select the **Log** option to create a plot with a logarithmic scale for that variable.

The Markers... Tool. It is often very useful to record information about your solutions on their graphs. The **Marker** tool can be used to add various symbols and text at any location in the graph window. To add a marker, first bring up the **Edit ...** menu, and select **Marker**. The window that appears in Figure 33 will open. Click **Add** to add a new marker to the plot.

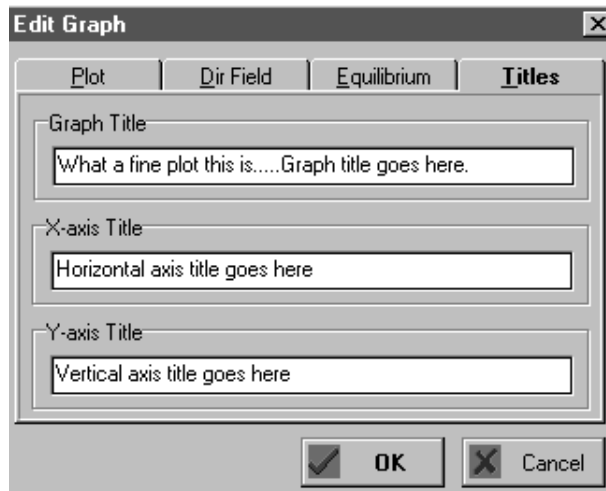


Figure 31: Adding titles to a plot.

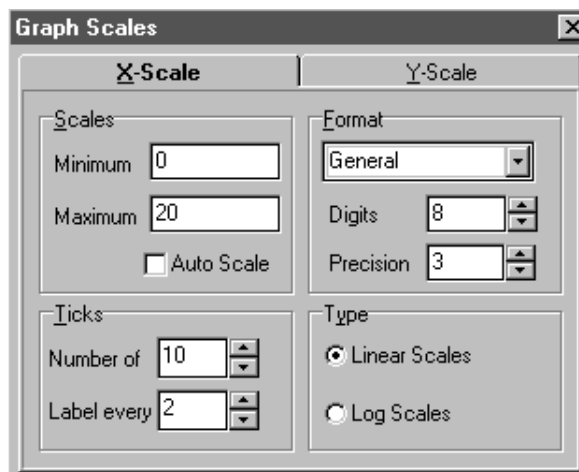


Figure 32: Control panel for graph scales.

Click **Add** to add a new marker to the plot.

There are controls to place special symbols, lines, or text at prescribed positions on the screen. Just type the horizontal (X) and vertical (Y) coordinates using the scales from the graph, and the text, line or symbol will appear there. For example, the commands shown in Figure 33 will place the words **Marker Text** on the plot at the location $X=1$, $Y=1$. The color, size and style of these markers can be adjusted. A marker can be deleted by selecting it from the list, and then clicking the **Remove** button.

SECTION 9: CREATING CUSTOM 2D PLOTS

It is possible to create plots with graphs of different variables plotted at once, with custom graphical features to make the plots more informative or visually interesting. The graph in Figure 34 shows a customized plot from the Library file *Cold Pills II* that has several different features.

To create such a plot, select the 2D tab in a graph window. When the 2D tab is selected, the window is initially blank. Click on the menu icon, or use a right button click in this blank window to open the **Plot Control Mode** menu and select **Edit...** The **Edit 2D Graph** window will open (Figure 35). This window has controls for creating customized plots. There are four panels under the tabs. The **Dir Field**, **Equilibrium** and **Titles** panels function just as described above for default plots.

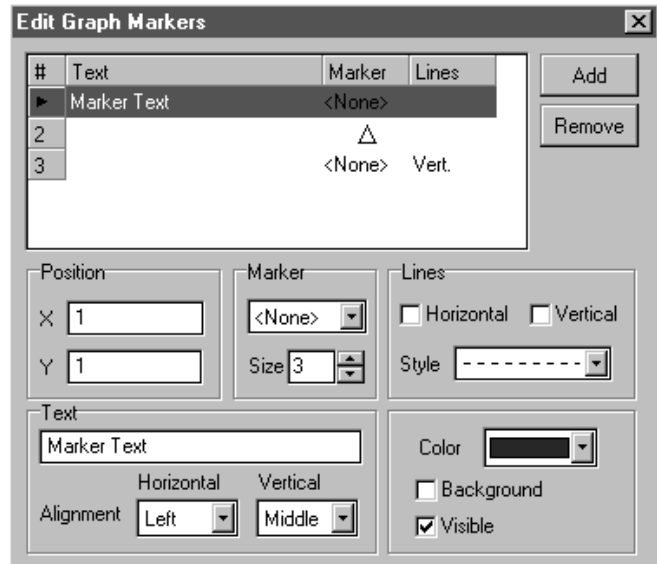


Figure 33: Adding markers to a plot.

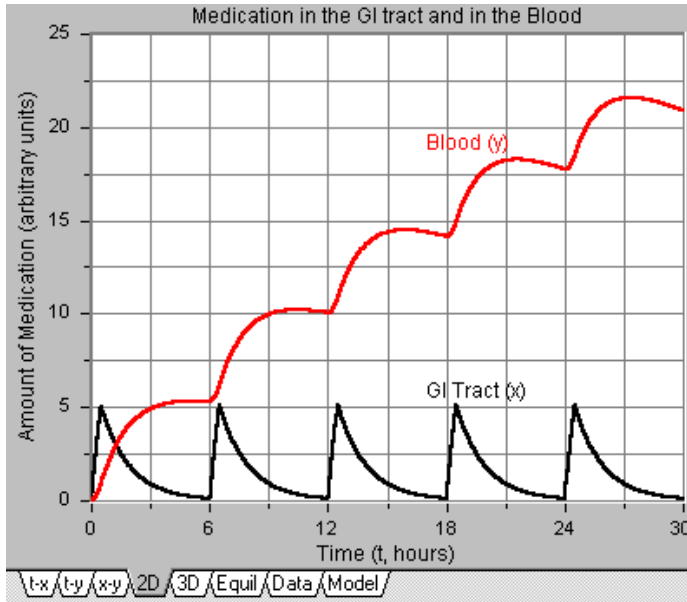


Figure 34: A custom plot with several features.

The **Plot** panel has a control to select the variable that will be displayed on the horizontal axis; the default is t , and the pull-down menu has a list of all variables that can be plotted. Up to five different variables can be simultaneously plotted on the vertical axis. Their colors and other attributes can be independently edited. To plot a variable on the vertical axis, just enter or select from the menu a name for the variable, and it will be added to the set of variables to be plotted. In Figure 35, the variables $Temp$ and Ts of Example 1 are both to be plotted against t . By selecting the line-style option of **<None>**, a curve, and adding Markers to the plot, the graph can be rendered “discretely” and not interpolated as a solid curve. To view an example of creating a plot of this type, look at the custom 2D graphs in the library file called “Coupled Oscillators: Entrainment” which is in the folder “Miscellaneous Models”. This feature can also be useful in creating Poincare time-section plots.

The Architect has a default color scheme so that the different curves on the vertical axis will have distinct colors. The user can modify the color scheme. State variables or auxiliary variables that may have been defined can be plotted, as in Figure 34.

SECTION 10: 3D PLOTS

Each of the graphics windows has a panel with the tab 3D. This panel will display three-dimensional plots. Default values for two of the axes are t and the first of the state variables. The user is free to change these two axes and to choose a variable for the third axis (Figure 36). The Architect will normally highlight the coordinate planes in a 3Dplot, and the plot of the 3D curve is projected on these planes. The planes, the projected curves, and the 3D curve itself are all different colors. You can then compare the projections of the 3D plot to their corresponding two-dimensional plots—just open a component plot or phase plot in the other graph window. It is possible to customize the plot, to eliminate the coordinate planes and projections, change the colors, and so on. In Figure 36, we have turned off the coordinate-plane projections in order to make the printed graphic simpler.

As you can see from Figure 36, there is a control panel in the margin at the right of the plot. This panel allows you to manipulate the 3D graph. For example, if you press one of the four buttons labeled **Views**, the observation point from which the 3D graph is viewed will change to the indicated pre-set position.

To rotate the individual axes of your plot, click on the corresponding control buttons (the up-down triangles) that are located in the **Rotate Axis** portion of the panel. A “free rotation” of the graph can be obtained by dragging the mouse pointer directly in the plot window. This process will rotate all three axes simultaneously. Zoom in on the plot by pressing the right **Zoom** control button, and zoom out by pressing the left button.

Important: the Architect needs to have a way to refer to the three axes of the plot in a general way. For example, on the buttons for setting the viewpoint, small pictures are drawn of an X - Y - Z coordinate system. *Do not interpret these references to mean, literally, that the variables in the plot are named X , Y , and Z .* The Architect denotes one of its directions as X , another by Y and the last by Z for its internal referencing scheme. The actual variables that are plotted, and their names, will come from your particular ODEs. This fact will be reflected in the text labels that the Architect will draw at the end of each axes on the plot itself. You may change the variable that is plotted on a given axis by using the 3D editing **Menu**, shown in Figure 37.

If you click on the **Menu** button, a pop-up appears with five selections. **Print** will send the graph to your printing device, whereas **Maximize** will expand the image to fill your screen, and **Show only current run** has the obvious effect.

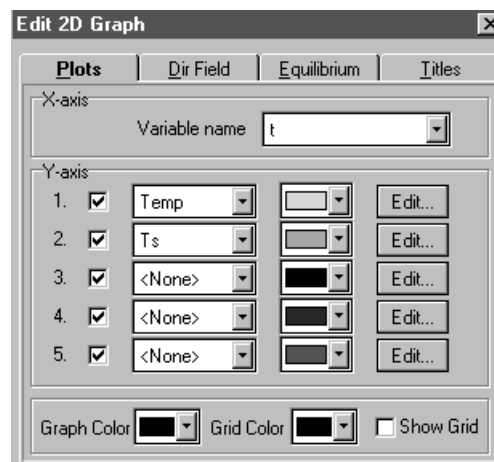


Figure 35: Customizing a 2D graph.

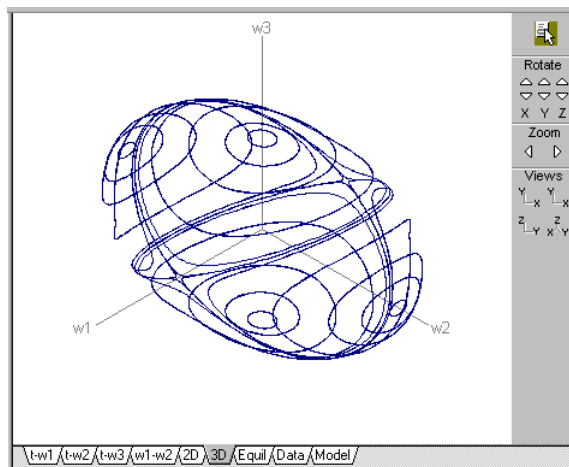


Figure 36: A 3D plot from the Library file Spinning Bodies I: The Momentum Ellipsoid.

Selecting the **Scales** option will open a window from which the scales of the three axes can be set by the user, which will override the auto-scale default. Select the **Edit** option to open the **Edit 3D Plot** window. The top-left quadrant of this window is used to specify which variables from your ODEs are plotted against the Architect's X , Y and Z axes. The lower left quadrant of the window has fields you can use to add customized labels to the axes.

The top-right portion of the **Edit 3D Plot** window is used to set which of the two-dimensional projections are displayed (the default is all three). The controls at the bottom-right of the window are used to set the colors for the coordinate planes (the **Background** buttons), the axes, the projected curves (the **XY**, **YZ**, **XZ** buttons) and the 3D curve itself (the **XYZ** button). All these parts of the plot can be independently adjusted to suit your fashion sense.

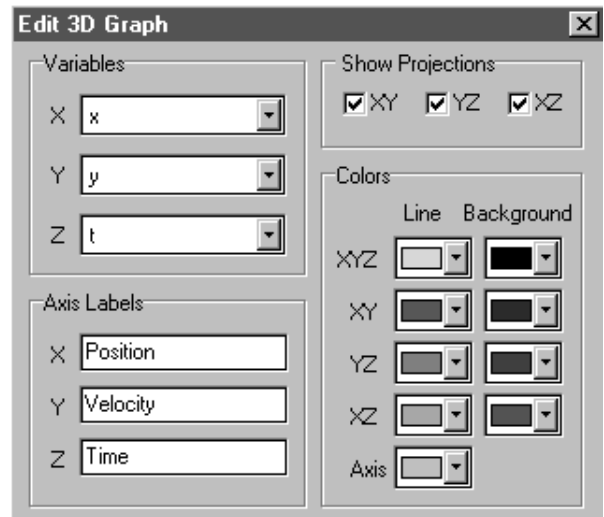


Figure 37: The **Edit 3D Plot** window.

SECTION 11: EXPLORING AND ANIMATING GRAPHS

Explore. When solution curves have been plotted, you may want to see the numerical information that corresponds to the graph. The **Explore** tool allows you to view all the relevant numerical information corresponding to a graph. To explore a solution curve, first use the selection tool and highlight the curve in the graph window (or else click on the **Data** tab and click the numbered tab for the curve to be explored). Then click on the **Explore** icon on the toolbar or select **Explore** from the **Solutions** menu; the F10 key is a shortcut to **Explore**. The dialog box that appears in Figure 38 will open. This box displays a table whose entries are the values of any parameters that are in your equations, the value of the independent variable t , the values of the dependent variable(s) in the plot, and the index of the data point that appears in the solution data table. At the bottom of the window there is a scroll bar. As the button in that bar is scrolled or dragged, a “cross-hair” moves along the graph, and the data in the **Explore** box is updated. If you select a different curve in the plot while the **Explore** window is open, the data in the window will change to that for the new curve. You can explore any graphs with this tool, even 3D graphs.

Animate. When the ODE Architect plots a solution curve, or series of solution curves, it first computes the complete solution, and then graphs the result. Graphs are not plotted as the points are calculated for a few reasons. First, the final scale of the graph is not generally known in advance; computing the solution before plotting allows the Architect to select a scale for the plot that would allow the entire solution to be viewed. If the scale is poor, the user can either adjust the scales on the plot or change the solver settings and solve again. Second, there is some information in watching the **Solver Progress** box (Figure 39) as a solution is computed. If there is a problem in computing the solution, the user will know

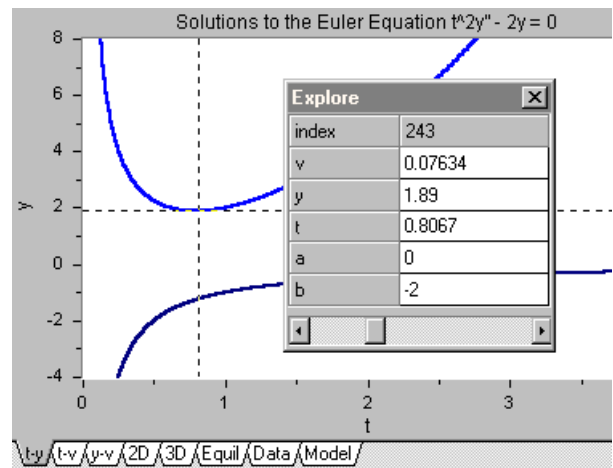


Figure 38: Exploring a plot.

at what time that occurred, and can halt the progress of the solver and re-adjust the solver parameters.

In any event, the Architect draws its solutions essentially instantaneously, and it is impossible to see them being drawn: they just appear. However, it is often useful to watch a solution curve as it is sketched, especially if that curve is an orbit in the phase plane. The **Animate** tool allows you to re-plot, or animate, solution curves with an adjustable “frame speed.” Families of solutions, such as produced by sweeping a parameter, can also be plotted, one at a time, at a user-adjustable animation rate.

To animate a plot, select **Animate** from the **Solutions** menu (the shortcut is the F9 key), and open the **Animation settings** dialog box, shown in Figure 40. This box allows the user to select whether the animation should be in **time** or by **runs**. A time animation will sketch the graph of the currently selected run, forward in time, at a rate controlled by the scroll bar on the right of the box.

Animating the plot using the **runs** setting will “play back” the entire set of runs, curve by curve, in the order they were created; this feature is most often used to make a movie of a family of solutions created by a parameter sweep. How the movie appears depends on whether the **Selection** tool in the **Plot Controls** is active or not. If it is active, all of the separate graphs remain on the screen, but are dimmed. As the animation proceeds, the graphs are successively highlighted. If the **Selection** tool is not active, the screen is erased and graphs of the different runs appear one at a time, in sequence, without the dimmed background image of the entire set of graphs.

The **toolbar** has two animation control buttons, one for opening the **Animation Settings** box and the other to initiate an animation in the active graph window; see Figure 41. Just click on them to access the animation tools directly. To stop an animation before it ends, click again on the animate button on the tool bar (the button on the left).

SECTION 12: MODEL-BASED ANIMATIONS

Models. Using the ODE Architect Model Builder, the user can create drawings or representations of physical models. The elements of such drawings have properties, such as their size, location, rotation and so on. These properties can be linked to dynamical variables in a system of ODEs, so that when the ODE is solved, the corresponding model will be animated. We call such objects **model based animations**.

Appendix B describes how to create a model using the Model Builder; more documentation on constructing models is found in the Help files of the Model Builder. Once a model has been constructed, it can be loaded into the Architect using the **Model** panel in the output side of the Architect. Click on the **Model** tab, and then right-click in the window. A dialog box matching Figure 42 will appear. From this box a model can be loaded (or unloaded). The other menu selections are for creating links between model variables and your ODEs, and for making a movie of your animation, so that it can be played with a Windows movie viewer, independently of

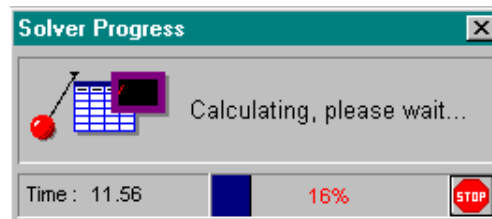


Figure 39: Solver progress window.

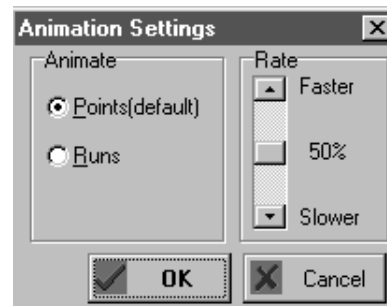


Figure 40: Settings for controlling animations.



Figure 41: Animation tools.

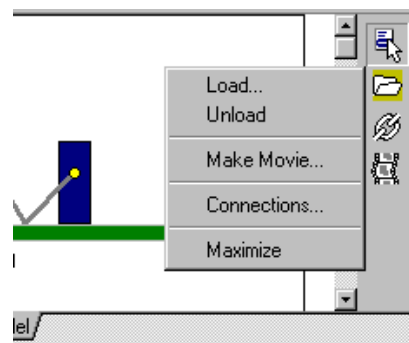


Figure 42: Model panel menu.

the ODE Architect.

When you select **Load**, you will open a window like the one in Figure 43. This window has two panels with tabs. The **Gallery** panel has a number of models that have already been designed, for compartment modeling, electric circuits, springs, pendulums, and so on. They can be loaded and linked to your own ODEs. The **Disk** tab allows you to load any model that you may have created or otherwise stored on your system.

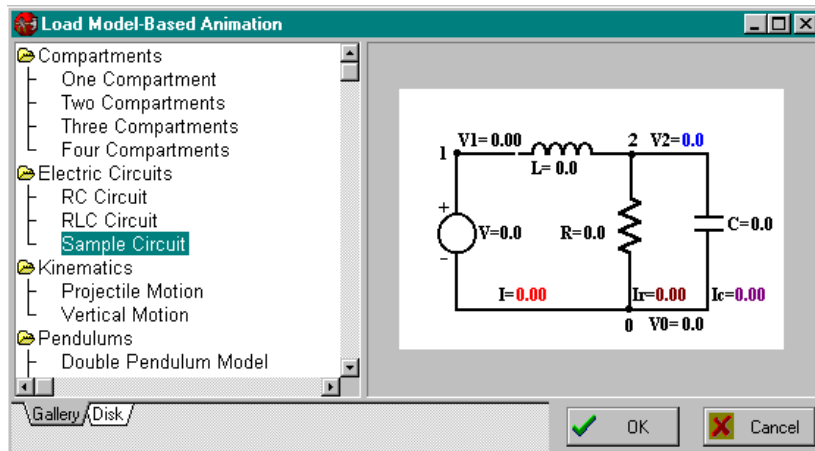


Figure 43: Loading models: the Gallery.

Figure 44 shows an example of a model of a spring-mass system, which appears in the Library file titled **Resonance I: The Undamped Linear Spring**, located in the Physical Models folder. The **connections** menu item opens the **Link Model Variables** window (Figure 45). From this window you can designate how the dynamic connections are made between ODE variables and model variables.

For example, in the Library file **Resonance I: The Undamped Linear Spring**, the position of the rectangular “mass” in the model has been assigned the variable name x . The width of the spring is adjusted in terms of x to “track” the mass. In the corresponding ODEs, the variable x represents the displacement of the mass from equilibrium. These two variables are then linked together, so that the solution to the ODE will animate the model. We note that you may link variables of different names—the dual use of x in this example does not mean that only variables of the same name can be linked.

Once the connections have been made, solving or animating ODE solutions will actuate the model. This feature of the Architect can be very helpful in relating the solutions of an ODE, and their graphs, to a tangible physical system. Sometimes you may wish to create an animation and use it outside the ODE Architect. The Architect allows you to do this by making a movie using the **avi** format. This movie can be stored, and

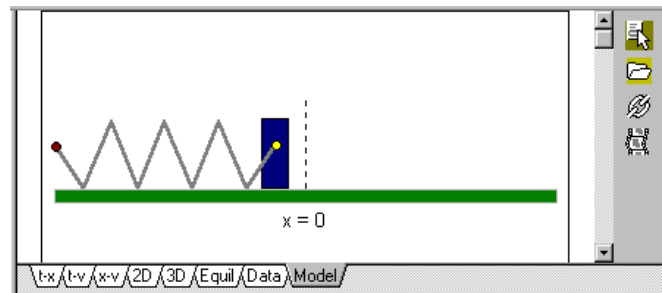


Figure 44: A model-based animation.

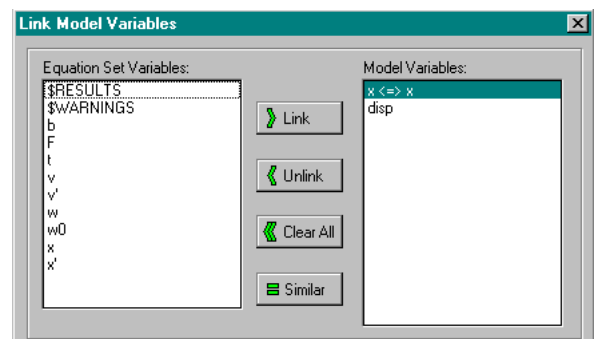


Figure 45: Linking model and ODE variables.

exported to be played back using a Windows movie player. For example, you could create an animation and then put a movie of it on your web page. When you select the **Make Movie** option from the **Model** menu (or click on the film icon at the right of the **Model** output panel), a dialog box (Figure 46) will appear.

Now type in a name for your movie, such as `my_movie.avi`, and adjust the settings for the frame rate that you want. Close the window, and the animation will run on the Architect; it will be slower than normal, as the Architect is now capturing the animation to your `avi` file. When the animation is finished, you will find the `avi` file on your disk and can play the movie back. You can load previously created movies from the **File** menu; they will load in a movie viewer window so that you can play it back as you wish.

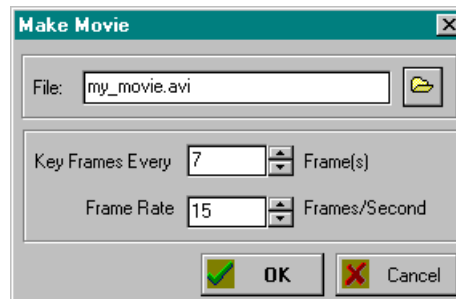


Figure 46: Creating a movie.

SECTION 13: THE ODE LIBRARY.

The ODE Architect has a library of over 100 ODEs. To open a library file, select the **Library** button on the **Toolbar**, or the **Library** option in the **File** menu. A window will open that shows the library file system (Figure 47). The ODE files are arranged in folders: **First Order Equations**, **Physical Models**, **Chemical Models**, **Population Models**, **Golden ODEs**, **Higher Dimensional Systems**, **2D Linear Systems**, **Miscellaneous Models**, **Bifurcations**, **Model-Based Animations** and **Tutorial Files**. Some ODE files are present in more than one of these folders. Open a folder and double-click on the description of an ODE file to load that file. The ODE files define an ODE or ODEs, assign values to the parameters, and display an interesting solution curve or curves. Many files suggest parameter values or initial conditions for further exploration, and several have animations.

We should point out that for some of the ODE models in the library, the graphs that automatically display were created after a careful exploration of parameter values, initial conditions, and solver settings. It would be easy for the user, when exploring these models, to change a setting, re-solve the model and produce unexpected results. Of course, such exploration is to be encouraged, and the user can always reopen the library file to return the model to its original state. You can **Clear** results at any time, and just use the equations in the file for your individual starting point.

If you edit a **Library** file and want to save your changes, you must use a different file name. The **Library** files are write-protected. A list of library files and their titles appears in the Appendix to this guide.

SECTION 14: THE REPORT WRITER

The Architect's **Report Writer** tool can be used to write reports for computer assignments or otherwise record your work. As described above, when you click on the **Report Writer** tool on the **Toolbar**, the Windows Word Pad will open. You can type your report into this window. Perhaps more importantly, you can paste graphs that you have produced using the Architect into your report. When you **Copy** a graph using the graph editing controls, you will open the dialog box shown in Figure 48. Use this window to set the physical size that you want the plot to have when it is pasted into your

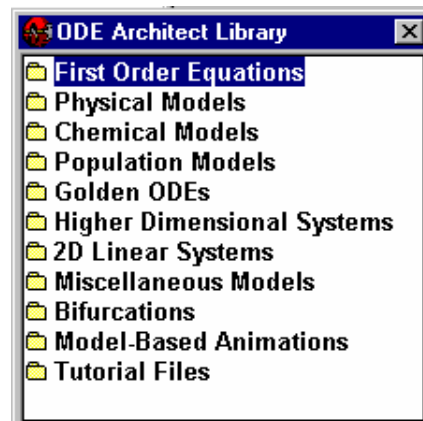


Figure 47: The ODE Library window.

report. At this point, you can choose to have the graphic converted to black-and-white, removing whatever colors that you may have used for your computer display, or it can be copied into the paste buffer exactly as you have created it. Depending on your printing environment, you may wish to follow this suggestion: before copying your graph to the clipboard, change the graph color (that is, the background color) to white, and the line color to black. We make this suggestion because it is often better for printing purposes to have black-on-white than white-on-black or other such color-intensive schemes.

Once you have copied your graph to the clipboard, re-open your report and paste it in using the Word Pad editing commands. It is possible to paste your graphs into word processing programs other than the Word Pad, such as Microsoft Word and Word Perfect, if you prefer to write your reports using such software. These other word processing programs would offer the advantage of being able to type more complicated mathematical equations and symbols. You can now print your graph as part of your report narrative.

SECTION 15: THE ODE ARCHITECT Expert MODE

As mentioned briefly above, the ODE Architect has a second mode, which we call the **Expert Mode**. When in **Expert Mode**, the user gives up much of the simplicity of the graphical user interface for enhanced flexibility of input and output control. For example, Figure 49 shows a session in **Expert Mode**: note the multiple graph windows, an animation window, and the various data windows. In **Expert Mode**, the user can have multiple sets of ODEs active at once, and can plot on one graph solutions from different sets. From the **Expert mode**, the numeric data for solutions can be cut-and-pasted into other programs, such as a spreadsheet or word processor.

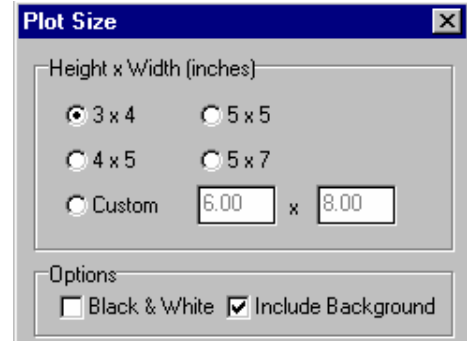


Figure 48: The Plot Size window for copying plots to paste buffer.

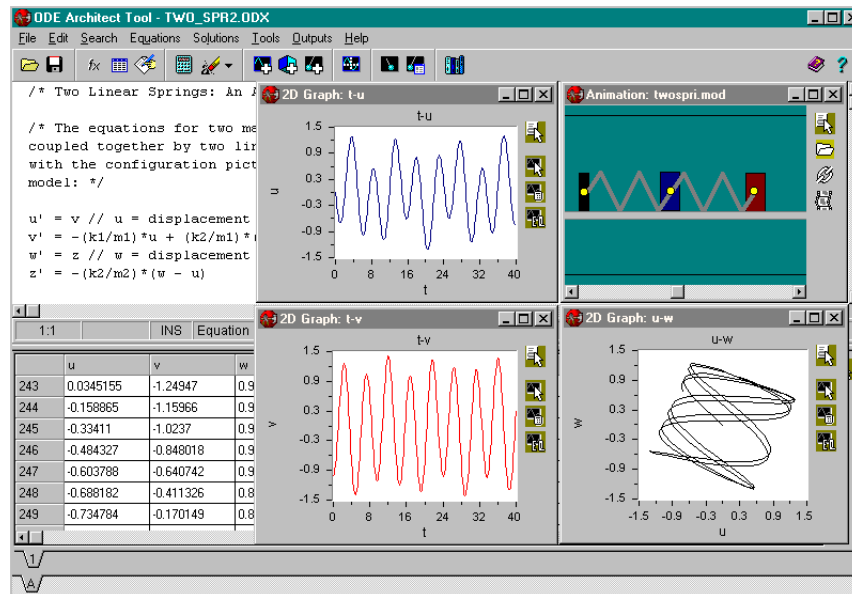


Figure 49: The ODE Architect Expert Mode.

The standard mode of the Architect has most of the features of **Expert Mode**, but not all. Once you

are comfortable with the standard features of the Architect, you should be able to master the **Expert** mode with little difficulty.

SECTION 16:. CONCLUDING REMARKS.

The ODE Architect is designed to be an easy-to-use ODE solver for use in course work, as a tool for practicing scientists and engineers, and as a tool for mathematical research. As you use the Architect you may have suggestions for improvements or features we should consider in future versions. We would appreciate your comments, as we want the ODE Architect to be the best tool we can fashion. To help you report your comments to us, the following URL points to the ODE Architect home page, where you will find a simple electronic form you can use: <http://www.math.hmc.edu/codee/architect>. Have fun with your ODEs, and enjoy the ODE Architect.

APPENDIX A: CONTENTS OF THE ODE LIBRARY

FILENAME	ANI.	DESCRIPTION
FOLDER FIRST ORDER EQUATIONS		
sing1.odx		A Linear ODE with a Singularity I
sing2.odx		A Linear ODE with a Singularity II
compress.odx		A First-Order ODE With “Data Compression”
aircon.odx		It’s Too Hot! Modeling the Automatic Control of an Air Conditioner
radioact.odx		Radioactive Decay
newton1.odx		Newton’s Law of Cooling I: Constant External Temperature
newton2.odx		Newton’s Law of Cooling II: Variable External Temperature
drag1.odx		Falling Bodies I: Linear Drag
drag2.odx		Falling Bodies II: Newtonian Drag
message.odx		Modeling a Digital Signal: A Coaxial Cable
logistic.odx		The Logistic Equation for Population Growth
harvest1.odx		Harvesting Populations I: Proportional Rate Harvesting
harvest2.odx		Harvesting Populations II: Constant Rate Harvesting
seasonal.odx		A Model for Seasonal Harvesting of a Limited Resource
FOLDER PHYSICAL MODELS		
radioact.odx		Radioactive Decay
newton1.odx		Newton’s Law of Cooling I: Constant External Temperature
newton2.odx		Newton’s Law of Cooling II: Variable External Temperature
drag1.odx		Falling Bodies I: Linear Drag
drag2.odx		Falling Bodies II: Newtonian Drag
hooke.odx	Y	The Linear Unforced Spring: Hooke’s Law
drivhook.odx	Y	The Forced, Damped Linear Spring: The Phenomenon of Beats
sq-beats.odx	Y	Beats II: Driving an Undamped Linear Spring with a Square Wave
resonan.odx	Y	Resonance I: The Undamped Linear Spring
sq-reson.odx	Y	Resonance II: Driving an Undamped Linear Spring with a Square Wave
tri-beat.odx	Y	The Beat Goes On: Linear, Undamped Spring Driven by Sawtooth Wave
twforce.odx	Y	A Damped System with a Triangular Wave Forcing Function
dcforce1.odx	Y	A Damped System with a Discontinuous Forcing Function
dcforce2.odx	Y	A Damped System with a Square Wave Forcing Function
impulse.odx	Y	A Damped System with an Impulse Forcing Function
agespr.odx	Y	Modeling an Aging Spring
hardspr.odx	Y	Modeling a “Hard” Spring
softspr.odx	Y	Modeling a “Soft” Spring
two-spr.odx	Y	Two Linear Springs: Lissajous Curves
two-spr2.odx	Y	Two Linear Springs: An Animation
threespr.odx	Y	Three Linear Springs: An Animation
message.odx		Modeling a Digital Signal: A Coaxial Cable
paralrc.odx		A Parallel L-R-C Electric Circuit
scroll.odx		The Scroll Circuit: Organized Chaos
nlpend.odx	Y	The Undamped Nonlinear Pendulum
nlpend2.odx	Y	The Unforced, Damped Nonlinear Pendulum
drivpend.odx	Y	The Driven Nonlinear Pendulum
nlpendsq.odx	Y	The Nonlinear Pendulum Driven by a Square Wave Forcing Function
varypend.odx	Y	A Model of a Pendulum with a Variable Length Rod
doubpen1.odx	Y	The Linear Double Pendulum
doublpen.odx	Y	The Nonlinear Double Pendulum
spin1.odx		Spinning Bodies and Integral Surfaces I: The Momentum Ellipsoid
spin2.odx		Spinning Bodies and Integral Surfaces II: The Cylinder
vander.odx		The van der Pol Oscillator
duffing.odx	Y	Duffing’s System
well.odx		A Double-Well Potential

FILENAME	ANI.	DESCRIPTION
FOLDER CHEMICAL MODELS		
autocat.odx brussel.odx oregon.odx		The Autocatalator Reaction The Brusselator Reaction System The Oregonator Model for Chemical Oscillations
FOLDER POPULATION MODELS		
logistic.odx harvest1.odx harvest2.odx seasonal.odx predprey.odx pred-log.odx satiatate.odx sat-mov.odx compete.odx mutualis.odx budworm.odx sir.odx possum.odx	Y	The Logistic Equation for Population Growth Harvesting Populations I: Proportional Rate Harvesting Harvesting Populations II: Constant Rate Harvesting A Model for Seasonal Harvesting of a Limited Resource The Lotka-Volterra Equations for Predator-Prey Systems A Predator-Prey System with Resource Limitation Predator-Prey Model with Satiation Predator-Prey Model with Satiation: An Animation of the Bifurcations A Model for Species Competition Mutualism: Symbiotic Interactions An Insect Outbreak Model: The Spruce Budworm The SIR Model for an Epidemic The Possum Plague: Disease Dynamics Down Under
FOLDER GOLDEN ODES		
sing1.odx sing2.odx compress.odx euler.odx airy.odx bessel.odx painleve.odx painlev2.odx parrot.odx teddy.odx nonaut1.odx cyclgraf.odx lazy8.odx sadnodsp.odx nlcenter.odx twocycle.odx twonodsp.odx twosp-cn.odx duffing.odx vander.odx	Y	A Linear ODE with a Singularity I A Linear ODE with a Singularity II A First-Order ODE With "Data Compression" Euler Equations Airy's Equation Bessel's Equation Painleve Transcendents I Painleve Transcendents II An Interesting Phase Portrait: The Parrot An Interesting Phase Portrait: The Teddy Bears A Non-Autonomous, Undriven Linear System A System with Cycle Graphs The Lazy-Eight Cycle-Graph A System with a Node, A Saddle, and a Spiral Point An Interesting Phase Portrait: A Non-Linear Center A Quadratic System with Two Limit Cycles System with Two Nodes and Two Saddle Points A System with Two Saddles and Two Centers Duffing's System The van der Pol Oscillator
FOLDER HIGHER DIMENSIONAL SYSTEMS		
conserv1.odx lorenz.odx roessler.odx autocat.odx oregon.odx lead.odx two-spr.odx two-spr2.odx threespr.odx scroll.odx spin1.odx spin2.odx sir.odx	Y Y Y	A Three-Dimensional Conservative System The Lorenz Equations The Roessler Attractor The Autocatalator Reaction The Oregonator Model for Chemical Oscillations A Model for Lead in the Body Two Linear Springs: Lissajous Curves Two Linear Springs: An Animation Three Linear Springs: An Animation The Scroll Circuit: Organized Chaos Spinning Bodies and Integral Surfaces I: The Momentum Ellipsoid Spinning Bodies and Integral Surfaces II: The Cylinder The SIR Model for an Epidemic

FILENAME	ANI.	DESCRIPTION
FOLDER MISCELLANEOUS MODELS		
aircon.odx		It's Too Hot! Modeling the Automatic Control of an Air Conditioner
cold1.odx		Cold Pills I: A Model for the Flow of a Dose of Medication in the Body
cold2.odx		Cold Pills II: A Model for the Flow of Medication with Periodic Dosage
lead.odx		A Model for Lead in the Body
combat.odx		Lanchester's Combat Model
fitzhugh.odx		The Fitzhugh-Nagumo Equations: A Model for Neural Activity
twoosci.odx	Y	Coupled Oscillators: The Tortoise and the Hare
twoosci2.odx	Y	Coupled Oscillators: Entrainment
FOLDER BIFURCATIONS		
saddnode.odx		A Saddle-Node Bifurcation
trancrit.odx		A Transcritical Bifurcation
pitchfrk.odx		A Pitchfork Bifurcation
infinper.odx		An Infinite-Period Bifurcation
hopf.odx		Hopf Bifurcation I
hopf2.odx		Hopf Bifurcation II: Subcritical Hopf Bifurcation
hopf2mov.odx	Y	Subcritical Hopf Bifurcation: An Animation
hopf3.odx		Hopf Bifurcation III: Moving Equilibrium Point
hopf3mov.odx	Y	Moving Equilibrium Point: An Animation
homohopf.odx		Hopf Bifurcation: Bifurcation with Homoclinic Orbit
homo-mov.odx	Y	Bifurcation with Homoclinic Orbit: An Animation
homobif.odx		A Homoclinic Bifurcation
homobmov.odx	Y	A Homoclinic Bifurcation: An Animation
FOLDER MODEL-BASED ANIMATIONS		
compart1.odx	Y	A Linear Single Compartment Model
compart2.odx	Y	A Linear Two Compartment Model
compart3.odx	Y	A Linear Three Compartment Model
compart4.odx	Y	A Linear Four Compartment Model
circrc.odx	Y	A Simple RC Electric Circuit
cirrc1.odx	Y	A Simple RLC Electric Circuit
cirrc2.odx	Y	A Sample Electric Circuit
project1.odx	Y	Projectile Motion
risefall.odx	Y	Vertical Motion
penddbl.odx	Y	Double Pendulum Model (Opens in Expert Mode)
pendsimp.odx	Y	The Driven Nonlinear Pendulum
pendvar.odx	Y	The Driven Nonlinear Pendulum with Adjustable Length
sprcpl2.odx	Y	Coupled Springs with Two Masses of the Same Size
sprcpl1.odx	Y	Coupled Springs with Two Masses of Different Size
sprsmpl1.odx	Y	A Simple Linear Spring
sprsmpl2.odx	Y	Mass with Two Springs
sprvert1.odx	Y	A Simple Vertical Linear Spring
sprvert2.odx	Y	Two Vertical Linear Springs

Notes: A "Y" in the second column denotes a Library file that incorporates an animation. Some Library files are present in more than one folder.

Building a model of a simple pendulum: Detailed instructions. To supplement the on-line Model Builder help, we present below detailed instructions on how to build a model of a simple pendulum using the Model Builder¹. You should have the Model Builder program open, and then follow along below:

1. In the ODE Architect Tool, go to the menu bar, choose **Tools|Model Builder**. Now you should have the Model Builder Window open, displaying the Property Inspector Box.
2. Now, create a new file within the model builder by choosing **File|New Model**.
3. First, define the size of the model you want to build. In the Property Inspector Box, click on the **Display** tab. Locate the two settings for Window Height and Window Width. The default values for these two are 300 pixels and 200 pixels, respectively. You can change the default values by left-clicking on the values and inserting a new number; remember to press Enter on your keyboard after you change any setting.
4. The other two settings, World Height and World Width define the proportions of the model displayed. In other words, by increasing these numbers, you will shrink the whole model by a certain proportion. For now, leave these two numbers as default, and if our model appears to be too small or too big as a whole, you can come back and change these two settings.
5. Still within the **Display** tab, you can change the background color by going to the Brush option. Double-click on the Brush cell (with a "+"), two options will come up, left-click on the Color option and choose one from the drop-down menu.
6. Now you are ready to do some actual model building! To assist you in placing and measuring the relative parts of the model, it is recommended that you use the "Snap to Grid" capability of the software. Go to the menu bar, select the **View|Grid** submenu, and select an appropriately scaled spacing. Make sure that both the Show Grid and Snap to Grid options are checked. Now, click on the Elements Tab in the Property Inspector Window to start making the individual elements of your model.
7. Use a line to represent the rigid rod of the simple pendulum. Press the Line Tool Button from the tool bar underneath the menu bar. Now, position the cursor in the middle of the window (right now this window should still be called untitled). Hold the left-button of the mouse and drag down to make a vertical line that will simulate the rod. You can configure this line, such as changing its color, by selecting Line1 in the **Elements** tab and Double-click on Pen (with a "+").
8. Now make another part to simulate the bob. Select the Rectangle Tool Button from the tool bar. Hold the left-button of the mouse and drag diagonally across the screen. If you don't like the bob you just made, you can select Rect1 from the **Elements** Tab and hit the Delete key on the keyboard to delete it. Similar to Line1, the bob's appearance can be configured by accessing the Brush and the Pen options.
9. To position the bob, press the Translation Tool Button on the toolbar, and drag the bob so that it hangs from Line1 (You can also use the arrow keys on the keyboard for more sensitive positioning).

¹My thanks to Mr. John Lu for providing these detailed instructions—MEM

10. With the two individual parts made and placed at the right location, now combine them into one object, so that they can move together in unity. Click on any one part of the elements you want to group, and, while holding the shift key, click on the rest of the parts in the group individually. Go to the menu bar and select **Model|Make Part**.
11. We define the pivot point for this new combined part. Click on the Rotational Tool Button in the tool bar. The pivot point is represented by a small circle at the lower-left corner of the new part. Now drag it to the upper end of Line1, so that the new part will rotate about this point. With the Rotational Tool Button selected, you can drag the new part and it should simulate the motion of the simple pendulum.
12. We now assign a variable to the rotation, i.e., the angular displacement from the vertical equilibrium position. In the **Elements** tab, select the object Part1, which represents the simple pendulum as a whole. Now click on the **Rotation** option in the Property Inspector Window. Type "theta" for the variable name and press **Enter** to enter. You can check your model now by going to the **Variables** tab and enter a numerical value for theta in degrees. The simple pendulum should swing corresponding to the value of theta entered.

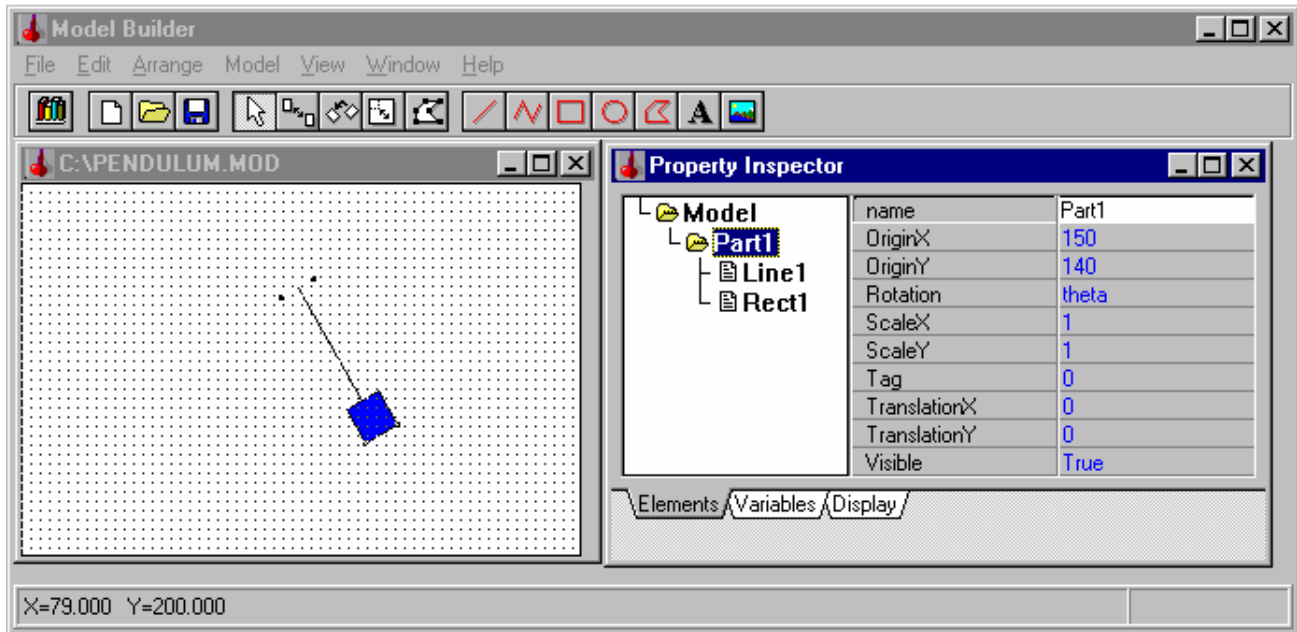


Figure 50: Building a model using the Model Builder Tool.

13. Now we save the model by going to the menu and select **File|Save**. A window will appear requesting the desired file name and path to save this model.
14. We have now finished the construction of the model for the simple pendulum. Figure ?? shows the result of this construction process. To animate it in the Architect, we follow these steps:
15. Type in the equations that govern the simple pendulum in the text field of the Architect: (using x to represent theta for easier typing) $x' = v$; $v' = -\sin(x)$; $\text{theta} = 180 * x / \pi$
16. Since angular displacement, x , is measured in degrees in the model builder, we make a unit conversion into radians by adding a line in the text field: $\text{theta} = 180 * x / \pi$

17. Click on Enter.
18. Leave the initial value for x as 0, the default value. Enter a value for the initial velocity, say, $v = 1$, and click Solve.
19. Click on the Model tab in the lower right graph window.
20. Right-click within the window and select Load.
21. Click on the Disk tab, and find the path to the model you just saved, then click OK.
22. Now the model should appear in this graph window (see Figure 51). Right-click on the model and select Connections. Choose theta in the left box and click on Link. Then press OK.
23. Now hit the Animate button or press F9, and sit back and enjoy.

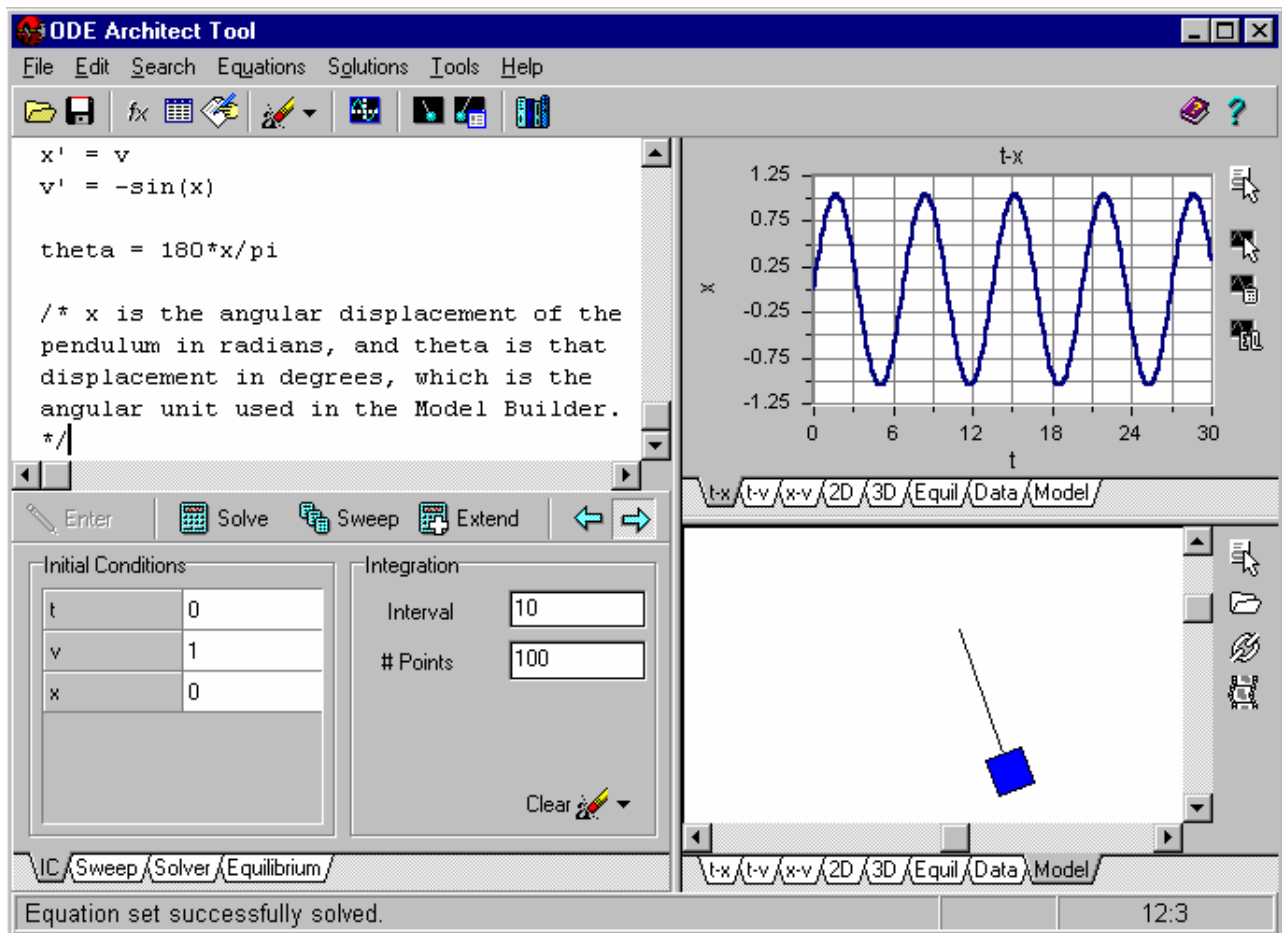


Figure 51: Animating the pendulum model with the Architect Tool.

Note: If for any reason you are not happy with the size and scales, you can always go back to the model builder and modify this file. You can change the Window Height and Window Width values to enlarge or contract the window size, or change the World View and World Height values to change the scale.

APPENDIX C: GRAPHING A FUNCTION

Since the Architect is all about graphical solutions to differential equations, you must have an ODE in the ODE editor window before you can do any graphing. Here's an example: Suppose you want to graph the nullcline of the ODE $x' = -x + \sin t$; the nullcline is the curve $x = \sin t$ where x' is zero. Enter the following into the ODE Editor:

$$\begin{aligned}x' &= -x + \sin(t) \\z &= \sin(t)\end{aligned}$$

Enter the IC's $t = 0, x = 1$, for example, in the input control window on the IC panel. On a graph window to the right of the screen, click on the 2D tab and select the variables x and z on the vertical axis (using different colors to distinguish them). Now solve. You will see the graph of the nullcline ($z = \sin(t)$) and the graph of the solution curve of the initial value problem. Wherever the solution curve cuts the nullcline it does so horizontally, as expected since $x' = 0$ at those points.

Things are more complicated if the equation of the nullcline is implicit. Let's work this out for a planar autonomous system:

$$x' = f(x, y), \quad y' = g(x, y), \tag{1}$$

where we want to plot the x -nullcline, $f(x, y) = 0$ and the y -nullcline $g(x, y) = 0$. First note that $f(x, y) = 0$ is an implicit solution of the differential equation

$$\frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy = 0 \tag{2}$$

since using the total differential we have

$$df(x, y) = \frac{\partial f}{\partial x} dx + \frac{\partial f}{\partial y} dy = d(0) = 0.$$

But the solutions of the differential equation (1) can be parameterized by t to obtain the system of ODEs that the Architect can solve:

$$\frac{dx}{dt} = \frac{\partial f}{\partial y}, \quad \frac{dy}{dt} = -\frac{\partial f}{\partial x}. \tag{3}$$

To see this, note that

$$\frac{dy/dt}{dx/dt} = \frac{dy}{dx} = \frac{\partial f/\partial y}{-\partial f/\partial x}$$

but $dy/dx = -(\partial f/\partial y)/(\partial f/\partial x)$ is just another way of writing (2).

So, choose as initial conditions for your system (3), $x = x_0, y = y_0$, where $f(x_0, y_0) = 0$ and use the Architect to solve. The resulting graph in the x - y plane is an arc of the x -nullcline $f(x, y) = 0$.

Now, if you want to overlay orbits of the original system (1) on the graph of the x -nullclines, delete the system (3) from the editor window (after you have drawn the nullcline!), insert the ODEs from the original system (1) and use the Architect to plot solution curves.

Here is an example: Let's plot the x -nullcline and some orbits of the system

$$x' = -x^3 + x^2y^2 - 1, \quad y' = -y. \tag{4}$$

The x -nullcline is defined by the implicit equation

$$f(x, y) = -x^3 + x^2y^2 - 1 = 0.$$

Note that $\partial f/\partial x = -3x^2 + 2xy^2$ and $\partial f/\partial y = 2x^2y$, and that $x = -1, y = 0$ is a point on the nullcline. So the initial value problem for the nullcline is

$$x' = 2xy^2, \quad y' = 3x^2 - 2xy^2, \quad x(0) = -1, \quad y(0) = 0. \quad (5)$$

Solve and plot in the x - y plane; be sure to solve backward and forward in time to get a significant part of the nullcline in the graphics window $-2 < x < 2, -2 < y < 2$. Then replace the ODEs in system (5) by the ODEs in system (4) and plot several orbits. Note that the orbits cross the nullcline horizontally.

QUICK REFERENCE

TO ENTER ODES

1. Type your ODEs into the ODE Editor with the derivative term all alone on the left: $x' = r*x*(1 - x) - h$ is fine, but $x' + h = r*x*(1 - x)$ will give an error.
2. Warning: Don't use both the upper and lower case versions of a letter in the same set of equations. For example, $x' = Y$, $Y' = -x$ is OK, but $x' = X$, $X' = -x$ is **not** OK.
3. Assign values to all your parameters and functions: $r = 1.0$; $h = -1 + \sin(t)$. Warning: List parameters and functions **after** listing the ODEs. Use semicolon to separate different equations on the same line.
4. Click on the **Enter** Button at the bottom right of the ODE Editor.

TO SOLVE YOUR ODES

1. Set the **Solve direction** to forward (\Rightarrow) or backward (\Leftarrow).
2. Enter your ICs (initial conditions).
3. Set the **Solve Interval**.
4. Set the number of points (**#Points**).
5. Click on the **Solve** button.

TO EXTEND YOUR SOLUTION

1. Set the direction in which you want to extend the solution. You can extend forward or backward.
2. Set the **Solve Interval**; you may change the **Interval** at any time.
3. Click on the **Extend** button.

SWEEPING A PARAMETER OR IC

1. Activate the **Sweep** panel by clicking on the **Sweep** tab at the bottom left.
2. Select the parameter(s) or variable(s) you want to sweep on. If a variable is selected, the initial values of that variable will be swept.
3. Select the kind of sweep you want.
4. Set the beginning and ending values, and the number of points in the sweep.
5. Return to the IC panel and make sure the **Integration** controls are set to what you want.
6. Click on the **Sweep** button at the top of the input controls panel.

EDITING GRAPHS

1. Your graph should tell a story that is instantly understandable, so keep in mind the following points when you do the final editing:
 - Are there enough solution curves to illustrate solution behavior, but not too many that they confuse the viewer?

- Are the axis scales and the tick marks appropriate?
- Are the curves wide enough to be clearly visible?
- Is the time interval right?
- Is the number of plotted points large enough to avoid “corners” in the plotted curve?
- Would informative titles or text placed on the graph help the viewer understand what you want the graph to “say”? (Use the titles and markers features to do this)

SOME SHORTCUTS AND COMMENTS

1. Right-clicking on an output window, such as a graph or equilibrium window, will bring up the editing menus.
2. When a graph window is in **Solve** mode, clicking in the window sets initial conditions.
3. When a graph window is in **Selection** mode, clicking on a curve will make that curve the current run. You can also activate the **Data** panel, and click a numbered tab. The run corresponding to that tab will be the new current run.
4. Re-click the animate button of the Toolbar to stop an animation that is running.
5. You can re-edit your equations in the ODE Editor at any time. If you do, **don't forget to Enter them!** Be careful not to confuse data and graphs from the previous equations with solutions to your new equations. For example, recalculate your direction fields.

WHAT TO DO IF THE SOLVER "HANGS UP"

1. Sometimes the Solver will not be able to complete your calculation, and will apparently get “stuck” or “hang up”. There is no single explanation for why the solver will hang; it depends on the nature of the ODEs and the settings that you have used on the various control panels. Here are a few things that you can try when the solver gets stuck:
2. Change the **Maximum time step** on the Solver control panel. If your graph just looks “wrong,” change the maximum time step to 0.1, say, and try again. Sometimes an adaptive solver will overlook points where the rate functions or solutions change suddenly. Lowering the maximum time step and resolving may eliminate the problem.
3. Change to a different solver from the one you were using. There are four different solvers to choose from on the Solver Control Panel.
4. Reduce the size of the integration **Interval** on the IC control panel.

REPORT WRITING

1. To prepare a graph for pasting into a report, click **Copy** on the **Plot Control** menu. Then set the desired size and click **Include Background**.
2. To prepare an entire screen for transfer to a report, hit the **Print Screen** button on the keyboard.
3. Hit **Tools** on the **Menu Bar** on the Architect Tool screen, then hit **Report Writer** to go to the Microsoft Word Pad application.
4. Click at the point you want to paste your graph in the Word Pad, and then hit **Paste** from the **Edit** menu of the Word Pad.
5. If you have a different word-processing program open on your Windows desktop, such as Microsoft Word, you can paste your graph into documents there by following the same instructions.

Index

- .odx files, 4
- 2D tab, 18
- 3D Plots, 19
- 3D tab, 19

- animate tool, 20
- animation controls, 20
- animation, model-based, 21
- animation, stopping, 21
- auto-scale, 17
- auxiliary function, example, 3
- auxiliary functions, 1

- bug report URL, 25
- building a model, details, 29
- built-in functions, 2

- clearing solutions, 10
- comments, adding to equation files, 1
- control panel for three-dimensional plots, 19
- controls, graph mode, 14
- current run, 9, 13
- customizing graphs, 18

- data markers, 15
- data, cutting and pasting, 25
- data, output, 13
- deleting solutions, 10
- direction field, 15
- direction field, editing, 17
- discrete dynamical systems, 6
- discrete graphs, 16
- discrete tool, 6

- edit custom data panel, 18
- edit menu, 4
- edit, direction field, 15
- editing 3D plots, 19
- editing graphs, the edit control panel, 15
- eigenvalues and eigenvectors), 12
- engineering functions, 4
- entering equations, 3
- equation files, 4
- equations menu, 5
- equations, entering, 1
- equations, example, 2

- equations, higher order, 3
- equations, independent variable, 2
- equations, syntax, 2
- equilibrium control panel, 11
- equilibrium mode, 14
- equilibrium points, editing markers, 17
- equilibrium, output, 12
- examples, 1–3, 5, 10
- expert mode, 1, 5, 24
- explore, 20
- extending solutions, 9

- file menu, 4
- functions of dependent variables, 3
- functions, built-in, 2

- graph mode, 14
- graph windows, frames, 12, 13
- graph windows, tabs, 13
- graphing a function, 32
- graphs, adding titles to, 17
- graphs, animating, 20
- graphs, annotating, 17
- graphs, changing colors and styles, 15
- graphs, copying, 15
- graphs, copying to reports, 24
- graphs, exploring, 20
- graphs, maximizing size, 14
- graphs, point plots, 16
- graphs, printing, 15
- graphs, restoring size, 14
- graphs, zooming and unzooming, 15

- IC (initial conditions), 8
- initial conditions control panel, 8
- initial conditions, graphical entry, 9
- initial conditions, setting, 8
- input control panels, 8
- input controls, 1
- integration interval control, 8

- library of ODEs, 4
- line style, 18
- linking a model, 22
- loading a model, 21
- lookup tables, 5

- marker tool, 17
- menus, for ODE Architect, 4
- Model Builder tool, 6
- model connections, 22
- model gallery, 22
- models, 21
- models menu, 21
- movies of models, 23

- nullclines, 32
- numerical data, 5

- ODE Editor, 1
- ODE library, 23
- ODE library, contents, 26
- ODE output, 12
- on-off functions, 4
- output windows, 1
- output windows, tabs, 12

- parameters, 2
- Plot Control Mode, 14
- Plot Control Mode menu, 14
- plot, selecting, 14
- plots, discrete, 18
- Poincaré sections, 18
- printing graphs, 15
- projections in 3D plots, 20

- report writer, 6, 23
- rotating axes in 3D plots, 19
- run, 9
- runs, deleting, 10

- scales, changing graph scales, 17
- search menu, 4
- selection mode, 14
- solutions menu, 8
- solve arrows, 9
- solve direction arrows, 9
- solve interval, 8
- solve, graph mode, 9
- solver control panel, 10
- solving ODEs backward, 9
- solving ODEs forward, 9
- square wave, 4
- standard functions, 2
- standard mode, 1
- step-functions, 4

- sweep control panel, 10
- sweep parameters, setting, 10
- sweeping two variables, 11
- systems of equations, 3

- three-dimensional plot, viewpoint, 19
- three-dimensional plots, 19
- three-dimensional plots, axis variables, 19
- three-dimensional plots, control panel, 19
- titles, 17
- toolbar, animate buttons, 21
- toolbar, for ODE Architect, 8
- tools menu, 6

- unzoom, 15
- user-defined functions, 5

- variable names, 4

- zoom, 15