Ordinary Differential Equations

3D-Filmstrip knows how to calculate and display solutions of the initial value problem for first and second order systems of ordinary differential equations (ODE) in one, two, or three dependent variables.

Let us recall briefly what this means. We will be dealing with vector-valued functions x of a single real variable t called the "time". Here x can take values in $\mathbb{R}$, $\mathbb{R}^2$, or $\mathbb{R}^3$. The problem is to find x from a knowledge of how x' depends on x and the t (in the first order case) or a knowledge of how its x'' depends on x, x', and t (in the second order case). Thus in the first order case the ODE we are trying to solve has the form $x' = f(x, t)$ and in the second order case it is $x'' = f(x, x', t)$.

In the first order case, the so-called Local Existence Theorem for First Order ODE tells us that provided the function f is differentiable, given an "initial time" t_0, and an "initial position" x_0, then in some sufficiently small interval around t_0, there will be a unique solution $x(t)$ to the ODE with $x(t_0) = x_0$.

There is a similar local existece theorem for second order ODE (which in fact is an easy consequence of the first order theorem). It says that given an initial time t_0, an initial position x_0, **and** an initial velocity v_0 then, in some sufficiently small interval around t_0, there will be a unique solution $x(t)$ of the ODE with $x(t_0) = x_0$ and $x'(t_0) = v_0$.

Theory provides not only an abstract existence theorem, but also many explicit numerical algorithms for finding approximating solutions, in terms of the function f and the initial data. One of the all-time favorites for general purposes is the so-called Fourth Order Runge-Kutta Method, and this is the one that 3D-Filmstrip uses.

Although the overall approach to solving such ODE is quite similar for first and second order ODE and for the various dimensions, the details for giving initial conditions and for displaying solutions are different for each case, and for that reason instead of having a single ODE category, it turns out to be convenient to have six. The naming of these categories is fairly self-evident. For example, the ODE(1D) 1stOrder Category deals with the case that x take values in $\mathbb{R}$, and the equation is first order, while the ODE(3D) 2ndOrder Category deals with the case that x take values in $\mathbb{R}^3$, and the equation is second order.

For the ODE(1D) 2ndOrder order category, the usual reduction is made to a first order system in two variables (x, u) where u represents x'.

In the case of the ODE(2D) 1stOrder category, the orbit is drawn dotted, with a constant time interval between dots. This gives a valuable visual clue concerning the velocity at which the orbit is traced out, but if you wish to turn this feature off, just set Dot Spacing to zero using the ODE Settings... dialog (see below).

In all the ODE categories, when you have chosen either a particular pre-programmed example (or set up your own, using the User Defined… feature) then as with the other categories you will first see a visualization of a default solution. This display will usually stop quickly on its own, but you can also click the mouse button (or type Command period) to stop it. You may then choose the item "ODE Settings…" in the Settings menu and this will allow you to set the various data the program needs to compute and display an orbit, namely:

a) the initial time,

b) the time-span,

c) the step-size (used in the Runge-Kutta method),

d) the initial value of x, and (in a second order case)

e) the initial value of x'.

Choosing Create from the Main menu will then display the solution for these newly selected settings.

There is an ODE control panel that opens by default just below the main display window. This has buttons to do more easily things you can also do with the menus (Create, Erase, Continue, double or half the scale, and bring up the dialog to set initial conditions, step-size, time-span, and dottedness). In addition there are buttons for single-stepping the ODE forward or backward, and there is a read-out of the current time, position, and velocity. This control panel can be hidden using the Hide ODE Controls command of the Action menu (and then may be reopened with the Show ODE Controls command).

The main display shows the evolution of an orbit in the phase space. By default, the program also shows projections of the orbit on the coordinate axes (using different colors to distinguish the projections). This display occurs in a second pane of the graphics window that opens automatically below the main pane.

This pane can be hidden by choosing "Hide Direction Fields" from the Action menu. There is a rectangular button at the right edge of the screen where the two panes meet. If you press on this button, the button itself will disappear and be replaced by a horizontal line. Drag the horizontal line to where you would like the new boundary between panes and release the mouse. (At least twenty percent of the total screen height must be devoted to each pane.)

For first order ODE in one and two dimensions, the program by default displays the direction field defined by the current ODE. (Since second order ODEs in one variable are reduced to first order ODEs in two variables, the direction field is also shown in this case.) The direction field of a time dependent ODE is updated every few integration steps. (As far as I know, 3D-Filmstrip is the only publicly available program that shows direction fields for time-dependent ODE.) For the ODE(3D) 2ndOrder category there is also a direction field shown (when the display is in stereo) for the special case of a charged particle in a magnetic field—but be careful, the field shown is the magnetic field, **not** the direction of the Lorentz force acting on the particle.

It is fairly easy to do a rough “phase space analysis” by keeping the other data fixed and varying the initial values. To make this easier, in certain categories it is possible to choose the initial conditions using the mouse. For example, in the ODE(1D) 1stOrder category, each time you click the mouse in the window, the forward and backward orbits are drawn through that initial value. Similarly, clicking the mouse in the ODE(1D) 2ndOrder category and ODE(2D) 1st Order category also creates an orbit with the mouse point as initial condition. Surprisingly, something similar even works for the ODE(2D) 2ndOrder category. Here to choose an initial condition AND velocity, either select IC By Mouse [Drag] from the Action menu or type Control I. You may then click and drag the mouse to choose the initial position (click) and velocity (drag).