ments are smaller than the computer's digital resolution. If this happens Igor will stop the calculation and complain.

Solution Methods

Igor makes four solution methods available, Runge-Kutta-Fehlberg, Bulirsch-Stoers with Richardson extrapolation, Adams-Moulton and Backward Differentiation Formula.

Runge-Kutta-Fehlberg is a robust method capable of surviving solutions or derivatives that aren't smooth, or even have discontinuous derivatives. Bulirsch-Stoers, for well-behaved systems, will take larger steps than Runge-Kutta-Fehlberg, so it may be considerably faster. Step size for a given problem is larger so you get greater accuracy. Of course, if you ask for values closer together than the achievable step size, you get no advantage from this.

Details of these methods can be found in the second edition of *Numerical Recipes*(see **References** on page III-349).

The Adams-Moulton and Backward Differentiation Formula (BDF) methods are adapted from the CVODE package developed at Lawrence Livermore National Laboratory. In our very limited experience, for well-behaved nonstiff systems the Bulirsch-Stoers method is much more efficient than either the Runge-Kutta-Fehlberg method or the Adams-Moulton method, in that it requires significantly fewer steps for a given problem.

As shown above, stiff systems benefit greatly by the use of the BDF method. However, for nonstiff methods, it is not as efficient as the other methods.

See **IntegrateODE** on page V-452 for references on these methods.

Interrupting IntegrateODE

Numerical solutions to differential equations can require considerable computation and, therefore, time. If you find that a solution is taking too long you can abort the operation by clicking the Abort button in the status bar. You may need to press and hold to make sure IntegrateODE notices.

When you abort an integration, IntegrateODE returns whatever results have been calculated. If those results are useful, you can restart the calculation from that point, using the last calculated result row as the initial conditions. Use the /R=(*startX*) flag to specify where you want to start.

For Igor programmers, the V_ODEStepCompleted variable will be set to the last result. It is probably a good idea to restart a step or two before that:

```
IntegrateODE/R=(V_ODEStepCompleted-1) …
```

Stopping and Restarting IntegrateODE

Any result can be used as initial conditions for a new solution. Thus, you can use the /R flag to calculate just a part of the solution, then finish later using the /R flag to pick up where you left off. For instance, using the harmonic oscillator example:

```
Make/D/O/N=(500,2) HarmonicOsc = 0
SetDimLabel 1,0,Velocity,HarmonicOsc
SetDimLabel 1,1,Displacement,HarmonicOsc
HarmonicOsc[0][%Velocity] = 5           // initial velocity
HarmonicOsc[0][%Displacement] = 0       // initial displacement
Make/D/O HarmPW={.01,.5,.1,.45}      // damping, freq, forcing amp and freq
Display HarmonicOsc[][%Displacement]

IntegrateODE/M=1/R=[,300] Harmonic, HarmPW, HarmonicOsc
```