Numerical Solution of Ordinary Differential Equations

Kurt Bryan, SIMIODE

Let us first load the Maple "plots" package (this is where the ***odeplot*** command lives)

```
> with(plots) :
```

Example 1: Many ODEs cannot be solved in an analytical or closed-form. For example, consider the ODE
u'(t) = sin(u(t)) - t*u(t)

```
> de := u'(t) = sin(u(t)) − t·u(t)
```

Maple can't provide an analytical solution, say with u(0) = 1:

```
> dsolve({de, u(0) = 1}, u(t))
```

The ***dsolve*** command comes up empty-handed. Yet the existence-uniqueness theorem applies and guarantees that a solution exists. In this case we can obtain a numerical approximation to the solution by using the ***dsolve*** command with the ***numeric*** option, as

```
> usol := dsolve({de, u(0) = 1}, u(t), numeric)
```

To evaluate the solution at a given time t, say t = 2.0, execute

```
> usol(2)
```

To plot the solution use the ***odeplot*** command

```
> odeplot(usol, t = 0 ..5)
```

Example 2: The various classical methods---Euler's method, the improved Euler method, or the RK4 method---can be implemented using the ***dsolve*** command. In each case below the step size is 0.1

```
> eulersol := dsolve({de, u(0) = 1}, u(t), numeric, method = classical[foreuler], stepsize = 0.1)
      #Euler's method
> impeuler := dsolve({de, u(0) = 1}, u(t), numeric, method = classical[heunform], stepsize
      = 0.1) #improved Euler's method
> rk4 := dsolve({de, u(0) = 1}, u(t), numeric, method = classical[rk4], stepsize = 0.1)
      #Runge−Kutta 4th order  method
```

A quick comparison. The numerical solution "usol" is an adaptive step size method and probably the most accurate.

```
> usol(2.0); eulersol(2.0); impeuler(2.0); rk4(2.0)
>
```