Solving ODEs Symbolically

Ordinary Differential Equations

ODE = equation describing a relationship between a function and its derivatives

- show up in dynamic systems -- systems that change with respect to time
- know the rate of change without know the underlying function

initial value problem (IVP) =

ordinary
vs
partial

first-order = only unknown function, 1st derivative of function, & independent variable show up in ODE

Solving an ODE

solving an ODE = finding the underlying function $y(t) = \ldots$

Example: show that $y(t) = 10\,e^{-2t}$ is the solution to $\frac{d\,y(t)}{d\,t} = -2\,y(t)$ with initial condition $y(0) = 10$

Check differential equation:

Check initial condition:

Steady-state of an ODE

steady-state = solution when things stop changing

To find the steady-state:

Example: find the steady-state of $\frac{d\,y(t)}{d\,t} = A \cdot y(t) + B$

Solving first-order ODEs symbolically using Maple

- construct and enter the ODE symbolically
- use `dsolve` to solve the ODE symbolically
- substitute values for symbolic constants
- explicitly plot the solution function

> $restart$

Equation for a linear first-order ordinary differential equation

> $ode := diff(y(t), t) = A \cdot y(t) + B$

Find the steady-state solution

> $steady := solve(rhs(ode) = 0, y(t))$

Solve the ODE symbolically

With no initial condition:

> $soln := dsolve(ode, y(t))$

With a symbolic initial condition:

> $solnWinitCond := dsolve(\{ode, y(0) = y0\}, y(t))$

Note: If you have an initial condition, you should use it in *desolve*

Plot the solution function

First, substitute values for symbolic constants

> $subVals := A = -\frac{2}{5}, B = \frac{1}{5}, y0 = 3$

> $solnSub := subs(subVals, solnWinitCond)$

> $plot(rhs(solnSub), t = 0..15)$

> $steadySub := subs(subVals, steady)$

> $restart$

Equation for a non-linear first-order ordinary differential equation

> $ode := diff(y(t), t) = \text{alpha}\cdot y(t) - \text{gamma}\cdot y(t)^2$

Solve symbolically

> $soln := dsolve(\{ode, y(0) = y0\}, y(t))$

Plot the solution

First, substitute values for symbolic constants

> $subVals := \text{alpha} = \frac{2}{3}, \text{gamma} = \frac{1}{10}, y0 = 5$

> $solnSub := subs(subVals, soln)$

See if Maple can simplify the equation

> $solnSub := simplify(solnSub)$

> $plot(rhs(solnSub), t = 0..10)$

Find the steady-state solution

> $steady := solve(rhs(ode) = 0, y(t))$

> $steadySub := subs(subVals, \{steady\})$

> $evalf(steadySub)$

Plot the solution for a different set of values for the symbolic constants

> $subVals := \text{alpha} = -\frac{2}{3}, \text{gamma} = \frac{1}{10}, y0 = 5$

> $solnSub := simplify(subs(subVals, soln))$

> $plot(rhs(solnSub), t = 0..10)$

Solving Systems of ODEs

> $restart$

Equations for a system of first-order ODEs

> $ode1 := diff(x1(t), t) = x1(t) - 2 \cdot x2(t)$

> $ode2 := diff(x2(t), t) = 3 \cdot x1(t) - 2 \cdot x2(t)$

Solve the system of ODEs symbolically

> $initCond := x1(0) = x1\_0, x2(0) = x2\_0$

> $odeSymSoln := dsolve(\{ode1, ode2, initCond\}, \{x1(t), x2(t)\})$

Solving a system of equations

vs

Solving a system of ODEs

Plot the solutions

> $constants := x1\_0 = 4, x2\_0 = 1$

> $odeSoln := subs(constants, odeSymSoln)$

> $plot([rhs(odeSoln[1]), rhs(odeSoln[2])], t = 0..8, color = [black, red])$

Solving Higher-Order ODEs in Maple

higher-order ODE : equation describing a relationship between 2^{nd} (or 3^{rd} or higher-order) derivative & the function

- may (or may not) have lower-order derivatives in the equation

Example: $\frac{d^2 y(t)}{d t^2} = -\frac{G}{L} y(t)$

> $restart$

To solve one (or more) higher order ODEs, treat it as a *system* of ODEs consisting of:

- one (or more) ODEs
- one initial condition for each order of each ODE

1 1st order ODE ⇒

2 1st order ODEs ⇒

1 2nd order ODE ⇒

2 2nd order ODEs ⇒

1 3rd order ODE ⇒

Higher-order ODE is implicitly a system of ODEs:

Specifying initial conditions:

function:

1^{st} derivative:

2^{nd} derivative:

Equation for a second-order ODE

> $ode2 := diff(y(t), t\$2) = -\frac{G}{L} \cdot y(t)$

Solve the second-order ODE

Second-order ODE requires 2 initial conditions, one for $y(t)$ and one for $\frac{\mathrm{d}\, y(t)}{\mathrm{d}\, t}$

> $initCond := y(0) = y_0, \mathrm{D}(y)(0) = 0$

Solve it symbolically

> $odeSymSoln := dsolve(\{ode2, initCond\}, y(t))$

>