Lab 14: Built in Matlab Routines for ODEs

Math 3341: Introduction to Scientific Computing Lab — Spring 2018

Using MATLAB's ODE solvers

When using most of the ODE solvers in MATLAB you usually will use the syntax:

```
[t,y] = ode45(odefun,tspan,y0).
```

For most ODE solvers in MATLAB:

- calling a solver without any output arguments automatically plots the solution as it is computed.
- using a single output argument stores the *structure* of the solution. Different components of the solution can be called using dot notation.

Table 1

Command	**Description**
`ode45(odefun,tspan,y0)`	where `tspan = [t0 tf]`, integrates the system of differential equations $y' = f(t,y)$ from `t0` to `tf` with initial conditions `y0`. Each row in the solution array `y` corresponds to a value returned in column vector `t`.
`ode23(odefun,tspan,y0)`	works similarly to `ode45` using a different method. See below for details
`quiver(u,v,x,y,scale,options)`	A quiver plot displays velocity vectors as arrows with components `(u,v)` at the points `(x,y)`. Arrows are automatically scaled to fit within the grid and then stretches them by the factor `scale`. A list of `options` can be specified like line color, thickness, etc as with traditional plot commands.

Choosing an ODE Solver

There are several choices when it comes to ode solvers in MATLAB. `ode45` is a versatile ODE solver and is the first solver you should try for most problems. However, if the problem is stiff or requires high accuracy, then there are other ODE solvers that might be better suited to the problem.

`ode45` is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair. It is a single-step solver which means in computing $y(t_n)$, it needs only the solution at the immediately preceding time point, $y(t_{n-1})$.

`ode23` is an implementation of an explicit Runge-Kutta (2,3) pair of Bogacki and Shampine. It may be more efficient than `ode45` at crude tolerances and in the presence of moderate stiffness. `ode23` is a single-step solver.

For more information visit Choose an ODE Solver.

Naming Conventions

The functions for solving odes follow a naming convention

- begin with `ode`
- followed by digits denoting the orders of the underlying integration formulas with
- `s`, `t` `tb` denote functions intended for stiff problems and `i` for fully implicit systems.

Lab Exercises

I. Direction Fields and Solution Curves

1. Download the files in `lab14files.zip`. This contains the script files needed to complete this lab.
2. Open the script file `lab14_part01.m`. Run the script file. You'll see that this plots the direction field and a single solution curve for the ODE

$$\frac{dy}{dt} = -y(t) - 5e^{-t}\sin(5t), \quad y(0) = 1$$

3. Add two additional solution curves to this plot by using two different initial conditions. Then plot the resulting solution curves. Be sure to add these to the `legend` command

II. Systems of ODEs

For this portion you will solve the system for problem 4 of homework 7 using the built in MATLAB function `ode45`.

$$\begin{cases} y_1'(t) = y_3; \\ y_2'(t) = y_4; \\ y_3'(t) = -2y_1 + \frac{3}{2}y_2; \\ y_4'(t) = \frac{4}{3}y_1 - 3y_2 \end{cases} ; \qquad \mathbf{y}(0) = \begin{bmatrix} -1 \\ 4 \\ 1 \\ 1 \end{bmatrix}; \qquad t \in [0, 15]$$

1. Define an anonymous function that defines the system.
2. Use `ode45` to solve the system with the given initial condition using a step size of $\Delta t = 0.01$.
3. Generate the following plots
   (a) The solutions $y_1(t)$ and $y_2(t)$ versus t.
   (b) The solutions $y_3(t)$ and $y_4(t)$ versus t.
   (c) The solutions of $y_1(t)$ versus $y_3(t)$.

Submit the plots and associated script files in the provided lab report template.