EENG 428 Introduction to Robotics Laboratory

EXPERIMENT 3

Solving Algebraic and Differential Equations

Objectives:

This experiment introduces using MATLAB for solving a system of algebraic equations, and solving differential equations with initial variables, with the ability to simplify the obtained results.

1. Solving a Set of Algebraic Equations

This section explains solving a single or a set of algebraic equations. The command (***solve***) is to be used.

This command has the following syntax:

```
solve(eq)

solve(eq,var)

solve(eq1,eq2,...,eqn)

g = solve(eq1,eq2,...,eqn,var1,var2,...,varn)
```

The latter syntax is explained as follows:

a. Single Equation/Expression

The equation (*eq*) to be solved can be either a symbolic expression or a string. Besides, if (*eq*) is a symbolic expression, or a string that does not contain the equality sign, then ***solve (eq)*** will solve the equation *eq=0* for its default variable.

Alternatively, the variable to solve for can be specified with:

solve(eq,var)

, which solves the equation *eq=0* for the specified variable *var*.

b. System of Equations

As an extension to the first case, the formulation ***solve(eq1,eq2,...,eqn)*** solves the set of equations *eq1, ea2 ,… eqn* if they are symbolic equations, and solves the equations *eq1=0, eq2=0, … eqn=0* if *eq1, eq2, … eqn* are strings. By default, these equations are solved for their default variables unless otherwise specified.

The following syntax:

g = solve(eq1,eq2,...,eqn,var1,var2,...,varn) finds the zeros for the system of equations for the variables specified as inputs (*var1*, *var2*, ,,,, *varn*, respectively).

Example 1.1: **Solving a single symbolic equation**

Solve the following symbolic quadratic equation:

$$ax^2 + bx + c = 0$$

The code:

```
solve('a*x^2 + b*x + c')
```

, returns:

```
-(b + (b^2 - 4*a*c)^(1/2))/(2*a)

 -(b - (b^2 - 4*a*c)^(1/2))/(2*a)
```

, being the two roots of the quadratic equation.

Example 1.2: Solving a set of algebraic equations

Solve the following set of algebraic equations:

$$x + y + z = 4$$

$$2x + 4y + 6z = 18$$

$$2x - y + z = 3$$

The code:

S=solve('x+y+z=4','2*x+4*y+6*z=18','2*x-y+z=3')

Returns the following three variables:

```
    x: [1x1 sym]

    y: [1x1 sym]

    z: [1x1 sym]
```

, where their values can be extracted from the string S, as follows:

```
>> S.x
 ans =
 1
```

```
>> S.y
 ans =

1
 >> S.z
 ans =
2
```

Example 1.3: Solving a set of symbolic equations

Solve the following symbolic set of equations:

$$a * u^2 + v^2 = 0$$

$$u - v = 1$$

$$a^2 - 5 * a + 6 = 0$$

The code:

```
A = solve('a*u^2 + v^2', 'u - v = 1', 'a^2 - 5*a + 6')
```

Returns the symbolic solution vector A:

```
A =

    a: [4x1 sym]
    u: [4x1 sym]
    v: [4x1 sym]
```

, where the variables a, u and v can be simplified as follows:

```
A.a =
 3
 2
 2
 3

A.u =
 (3^(1/2)*i)/4 + 1/4
 (2^(1/2)*i)/3 + 1/3
 1/3 - (2^(1/2)*i)/3
 1/4 - (3^(1/2)*i)/4

A.v =
    (3^(1/2)*i)/4 - 3/4
    (2^(1/2)*i)/3 - 2/3
 - (2^(1/2)*i)/3 - 2/3
 - (3^(1/2)*i)/4 - 3/4
```

2. Solving Differential Equations with Initial Conditions

The command ***dsolve*** is used to solve differential equations with several choices.

Syntax:

dsolve('eq1','eq2',...,'cond1','cond2',...,'v')

, where, *eq1, eq2, ...* is the set of equations to be solved, and *cond1, cond2,...* are their respective initial conditions. Besides, *v*, is the independent variable to solve for, which represents time in most practical cases. It should be emphasized that the symbol ***D*** denotes differentiation with respect to v. The primary default is *d/dx*. The letter *D* followed by a digit denotes repeated differentiation. For example, *D2* is *d2/dx2*. Any character immediately following a differentiation operator is a dependent variable. For example, *D3y* denotes the third derivative of $y(x)$ or $y(t)$.

Example 2.1: Solving a single initial value ODE

Solve the following symbolic ODE, and plot y(x) for $x \in$ [0,5]

$$\frac{d_2y(x)}{dx^2} + 8\frac{dy(x)}{dx} + 2y(x) = \cos(x), \quad y(0) = 0, \quad y'(0) = 1$$

The code:

```
y=dsolve(' D2y + 8*Dy + 2*y = cos(x)', 'y(0)=0, Dy(0)=1','x')
```

, returns the solution as:

```
y =

(14^(1/2)*exp(4*x - 14^(1/2)*x)*exp(x*(14^(1/2) - 4))*(sin(x) - cos(x)*(14^(1/2) - 4)))/(28*((14^(1/2) - 4)^2 + 1)) - (98*14^(1/2) + 378)/(exp(x*(14^(1/2) + 4))*(868*14^(1/2) + 3136)) - (14^(1/2)*exp(4*x + 14^(1/2)*x)*(sin(x) + cos(x)*(14^(1/2) + 4)))/(28*exp(x*(14^(1/2) + 4))*((14^(1/2) + 4)^2 + 1)) - (exp(x*(14^(1/2) - 4))*(98*14^(1/2) - 378))/(868*14^(1/2) - 3136)
```

If we want to plot y, it is done as:

```
>>x=linspace(0,5,100);

>>z = eval(vectorize(y));

>>plot(x,z)
```

Example 2.2 :Solving a set of ODEs

Solve the following set of ODEs

$$\frac{dx}{dy} = y$$

$$\frac{dy}{dx} = -x$$

The code:

```
z = dsolve('Dx = y', 'Dy = -x')
```

, returns the solution vector z

```
z =
    y: [1x1 sym]
    x: [1x1 sym]
```

, where the solutions can be extracted out from z by:

```
z.x

ans =
(C20*i)/exp(i*t) - C19*i*exp(i*t)

z.y

ans =
C19*exp(i*t) + C20/exp(i*t)
```

3. Homework

Considering the following questions, provide your answers in a report including your codes and associated outputs and/or plots by copying them and pasting on the report pages. To be submitted as a hard copy on next lab session. You should work individually and group work will be penalized.

1. Write a MATLAB code that solves the following set of algebraic equations:

$$x + y + z = 6$$
$$2x - y - z = 0$$
$$4x + 2y + z = 14$$

2. Write a MATLAB code to solve the following symbolic ODE

$$\frac{d_2 y}{dx^2} = xy$$