Lab 3: Plotting solutions in implicit form

This lab will teach you to numerically solve and plot implicit solutions to differential equations. In the process you will learn how to:

- Define and use inline functions of one and two variables,
- Use a fzero to find the root of an equation.

You should format your solutions to the 7 exercises with MATLAB's cell mode, using the template. Use publish to HTML format your results, and submit a print out of this.

MAT292, Fall 2010, Homayouni & Simpson

Contents

- Inline functions of one variable
  - Exercise 1
- Inline functions of two variables
  - Exercise 2
- Numerically solving an implicit equation
  - Exercise 3
- Plotting the solution
  - Exercise 4
  - Exercise 5
  - Exercise 6
  - Exercise 7

Inline functions of one variable

Suppose we wish to construct the function:

\[ f(x) = x \cdot \sin(x) \]

In a way that MATLAB can understand and manipulate. This will let us compute values of \( f(x) \), plot it, and create other functions from it.

This can be accomplished by defining an inline function of one variable, with the command:

\[
\text{function name} = @(\text{argument name}) \text{mathematical expression};
\]

For this example, the function name is \( f \), the argument name is \( x \), and the mathematical expression is \( x \cdot \sin(x) \).

Consider the following examples, and take note of:

1. The use of vectorization in the definitions, 2. The different ways that the output can be displayed, 3. The different variable names.

```matlab
f = @(x) x .* sin(x);

x = 1;
fprintf(' f(%g) = %g
', x, f(x));
fprintf(' f(%g) = %g
', 1, f(1));
disp(f(1));
f(1)
x = 2;
fprintf(' f(%g) = %g
', 2, f(2));
 fprintf(' f(%g) = %g
', 2, f(2));
disp(f(2));
```
disp(f([1,2]))

g = @(t) t.^2 - t +1;
g(1)
g(2)

% Construct an array of 100 grid points from -2 to 2, and plot g(t).
t = linspace(-2,2);
plot(t, g(t));
% Label axes
xlabel('t');
ylabel('g');

f(1) = 0.841471
f(1) = 0.841471
0.8415

ans =

0.8415

f(2) = 1.81859
f(2) = 1.81859
1.8186

ans =

1.8186

0.8415 1.8186

ans =

1

ans =

3
Exercise 1

Objective: Write and use an inline function of one variable.

Details: Define the inline function for

\[ f(x) = x^2 \cdot \exp(-x^2) \]

Use this function to compute \( f(0) \), \( f(1) \), and \( f(-1) \) and plot the function with 100 grid points from -3 to 3. Make sure to use vectorization in your definition, and label your axes.

Your submission should show the definition of the function, and the computation of these values.

**Inline functions of two variables**

Suppose we wish to solve the equation

\[ x(y) = y^3 + y \]

for \( y \) as a function of \( x \). If we do this by root finding, then, given a value of \( x_0 \), we need to find the root of the function

\[ g(y) = f(x_0, y) = y^3 + y - x_0 \]

We will accomplish this using inline functions.

Inline functions of two variables are defined as:

\[ \text{func} = @(x,y) f(x,y); \]

where \( \text{func} \) is the name you wish to assign your function, and \( f(x, y) \) is the relevant expression.

Some examples of defining and using inline functions of two variables are given below.
Some examples of defining and using inline functions of two variables are given below.

```matlab
f = @(x,y) y.^3 + y - x;
x = 0; y = 0;
fprintf(' f(x = %g, y = %g) = %g
', x, y, f(x, y));
x = 0; y = 1;
fprintf(' f(x = %g, y = %g) = %g
', x, y, f(x, y));
disp(f(x,y))
f(0,1)
```

```matlab
h = @(t,u) t.^2 + sin(u);
h(1, pi/2)
h(2, 2)
```

```
f(x = 0, y = 0) = 0
f(x = 0, y = 1) = 2

ans =
    2

ans =
    2

ans =
    4.9093
```

**Exercise 2**

Objective: Write and use an inline function of two variables.

Details: Define the inline function for

\[ f(x,y) = x^3 + y^3 - 3y \]

\[ f(x,y) = c \]

is the implicit solution to Example 1 of section 2.2

Use this function to compute \( f(0, 0) \), \( f(1,1) \), and \( f(-1,-1) \). Make sure to use vectorization in your definition.

Your submission should show the definition of the function, and the computation of these values.

**Numerically solving an implicit equation**

Given \( x_0 \), we will now solve the equation

\[ f(x_0, y) = y^3 + y - x_0 = 0 \]

for \( y \). This will be accomplished using fzero, which employs a more advanced for of Newton’s method.
The minimal arguments needed for using fzero are:

fzero(function_handle, guess)

function_handle is an abstract data structure which directs MATLAB towards a function of a single variable that it will try to zero.

We need to tell MATLAB that it is trying to solve for the y argument. This is done by creating an inline function of one variable from the existing function, i.e.

g = @(y) f(x0, y)

And then having fzero try to solve

g(y) = 0

You can either define the new inline function and use it:

g = @(y) f(x0, y); y = fzero(g, guess);

or you can put the inline function argument directly in the function handle argument:

y = fzero(@(y) f(x0,y), guess);

The guess, is an initial guess for what we anticipate y to be. Finding a good initial guess may require some experimentation. For this example, the guess may be quite poor, and the algorithm will still converge. For others, more care must be taken.

Note the various ways of constructing the inline function of one variable, and the different guesses.

```matlab
f = @(x,y) y.^3 + y - x;

x0 = 1;
guess = 1;
g = @(y) f(x0, y);
y = fzero(g, guess);
fprintf(' y(%g) = %g
', x0, y);

guess = 50;
y = fzero(g, guess);
fprintf(' y(%g) = %g
', x0, y);

x = 2;
g = @(y) f(x, y);
y = fzero(g, 2);
fprintf(' y(%g) = %g
', x, y);

guess = 2;
y = fzero(@(y) f(2, y), guess);
fprintf(' y(%g) = %g
', 2, y);
```

y(1) = 0.682328
y(1) = 0.682328
y(2) = 1
y(2) = 1

**Exercise 3**

Objective: Use fzero to solve an equation \( f(x,y) = 0 \) for y at different values of x.
Details: Using the m-file definition of
\[ f(x,y) = x^3 + y^3 - 3y \]
Define the appropriate inline function and compute the single positive solution at \( x = 0 \) and the two positive solutions at \( x = 1 \). The positive solution at \( f(0,y) = 0 \) is larger than 1.

You will need to experiment with different guesses to get these three solutions.

Your submission should show the definition of the function, and the computation of these three roots.

**Plotting the solution**

Now we are ready to plot \( y \) as a function of \( x \). We will construct an array of \( x \) values, solve for \( y \) at each value of \( x \), and then plot the results. This will be accomplished with a for loop.

```matlab
f = @(x,y) y.^3 + y - x;
% 50 points equally spaced from -5 to 2, inclusive
xvals = linspace(-5, 2, 50);
% Preallocate storage space for the solutions
yvals = zeros(size(xvals));
% Loop through all the values
for j = 1:length(xvals)
    yvals(j) = fzero(@(y) f(xvals(j), y), 0);
end
% Plot the results
plot(xvals, yvals);
xlabel('x');
ylabel('y');
```
**Exercise 4**

Objective: Plot a portion of an implicit equation.

Details: Plot the portion of the solution to
\[ f(x, y) = x^3 + y^3 - 3y \]
for \( x = 0 \) and \( y = 0 \) for \( x \) from -1 to 1. Use 50 grid points.

You will need to be careful with the initial guess you use in solving the equation.

Your submission should show the definition of the function, the construction of the arrays, the for loop, and the resultant figure.

Label your axes.

**Exercise 5**

Objective: Solve a differential equation and plot a portion of it.

Details: Find the general implicit solution of the differential equation
\[ \frac{dy}{dx} = \frac{x y^3}{(1+x^2)^{1/2}} \], exercise 14 of Section 2.2

Find the particular solution passing through \( y(0) = 1 \), and plot the solution with 50 grid points between -1 and 1.

Your submission should show the general and particular solutions, in implicit form, in the comments, the definition of the appropriate inline functions, arrays, the for loop, and the figure.

Label your axes.

**Exercise 6**
Objective: Solve a differential equation and plot the largest interval of existence for this solution.

Details: Find the particular implicit solution of the differential equation
\[ \frac{dy}{dx} = -\frac{x}{y}, \quad y(0) = -2 \]

Determine the largest interval of validity and plot the solution with 50 grid points in this interval.

In the comments before your code, explain why fzero has an error.

Your submission should show the particular solution, in implicit form, in the comments, the definition of the appropriate inline functions, arrays, loops, and the figure.

Label your axes.

Exercise 7

Objective: Solve a differential equation and plot the solution on the specified interval

Details: Find the particular implicit solution of the differential equation
\[ \frac{du}{dt} = -(u^4 - 1), \quad u(0) = 0.1 \]

and plot it for \( t \) from 0 to 1.

Determine a good starting guess that works throughout the interval and plot the solution with 50 grid points in this interval.

Your submission should show the particular solution, in implicit form, in the comments, the definition of the appropriate inline functions, arrays, loops, and the figure.

Label your axes.