Introduction to MATLAB for Economics

Introduction to Optimization in MATLAB

Javier Barbero

Universidad Autónoma de Madrid
http://www.javierbarbero.net

October 31, 2014

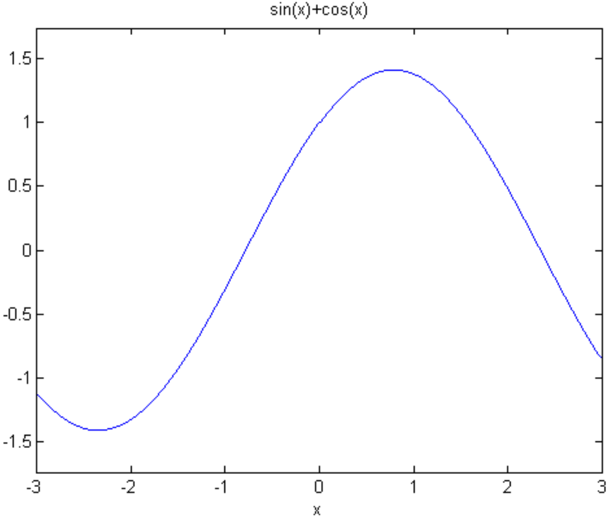
Optimization in MATLAB

- MATLAB can solve two types of optimization problems:
  - **Zero finding**: Find x such $f(x) = 0$.
  - **Minimization**: $\min f(x)$
- A maximization problem can be solved by minimizing the negative of the function: $\max f(x) = \min -f(x)$.
- Zero finding of functions of one variable and minimization can be solved with basic MATLAB.
- For finding the zeros of functions of several variables and minimization with constraints a toolbox is needed. The official *Optimization Toolbox* provides this functionality.

Zero finding

- The function **fzero** is used to find the zero of a 1-D function.
- The objective function can be a function in a separate file or an anonymous function.
- For the solver to work, you must supply an initial guess of where the zero is or an interval in which the zero is located.
- If a function has several zeros, the result is conditioned by the initial guess.
- The function returns the value of x where the zero is located. The value $y = f(x)$ is returned as the second argument.

Zero Finding. Example



Zero Finding. Example

```
% Define the objective function as an anonymous
    function
objFun = @(x) sin(x) + cos(x);

% Plot the objective function
ezplot(objFun, [-3,3]);

% Optimize it with zero as initial guess
[x, val] = fzero(objFun, 0)
```

Zero Finding. Example

```
% Find the zero at the left
>> [x, val] = fzero(objFun, 0)
x =
-0.7854
val =
-1.1102e-16

% Find the zero at the right
>> [x, val] = fzero(objFun, [0 3])
x =
2.3562
val =
-5.5511e-16
```

Minimization

- Minimization in one dimension is performed with the **fminbnd** functions.
- The function searches a minimum between a given interval (bound).
- Minimization of functions of several variables are performed with the **fminsearch** function.
- For the function, each variable is an element of a vector x.

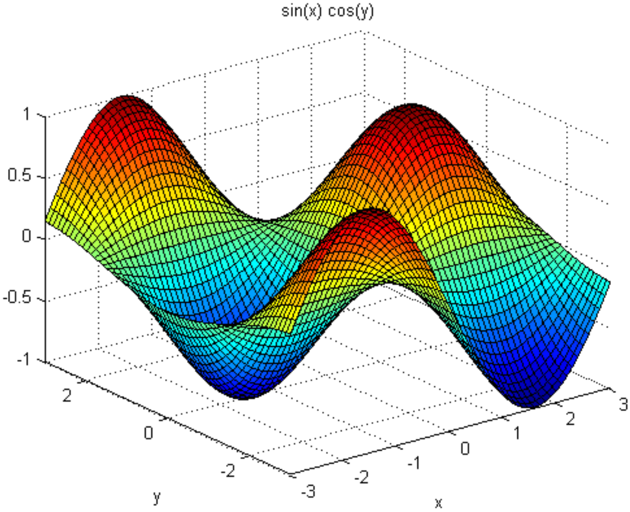
Minimization. Example 1-D

```
% Define the objective function as an anonymous
   function
objFun = @(x) sin(x) + cos(x);

% Plot the objective function
ezplot(objFun, [-3,3]);

% Search for a minimum between -3 and 0
[x, val] = fminbnd(objFun, -3, 0)
```

Minimization. Example 1-D

```
% Search for a minimum between -3 and 0
>> [x, val] = fminbnd(objFun, -3, 0)
x =
-2.3562
val =
-1.4142

% Find for a maximun between -1 and 2
>> [x, val] = fminbnd(@(x) -objFun(x), -1, 2)
x =
0.7854
val =
-1.4142
```

Zero Finding. Example 2-D



Minimization. Example 2-D

```
% Define the objective function as an anonymous
   function
objFun = @(x) sin(x(1))*cos(x(2));
objFunPlot = @(x,y) sin(x).*cos(y);

% Plot the objective function
ezsurf(objFunPlot, [-3,3]);

% Search for a minimum between -3 and 0
[x, val] = fminsearch(objFun, [0,0])
```

Minimization. Example 2-D

```
% Search for a minimum with guess (0,0)
>> [x, val] = fminsearch(objFun, [0,0])
x =
-1.5708    0.0000
val =
-1.0000

% Search for a maximum with guess (0,0)
>> [x, val] = fminsearch(@(x) -objFun(x), [0,0])
x =
1.5708   -0.0000
val =
-1.0000
```

Configuring the Optimizer

- The optimizer functions can be configured by passing them an *options* structure.
- The most important ones are:
  - Display: to display output at each iteration, only the final output, or only if there is a problem.
  - Tolerance both of the X and the Function Value.
  - Maximum number of iterations.
  - etc.
- Options structures are created with the **optimset** command.
- The full list of options is available at: `http://www.mathworks.com/help/matlab/ref/optimset.html` or typing `doc optimset`

Configuring the Optimizer. Example

```
% Define the objective function as an anonymous
   function
objFun = @(x) sin(x(1))*cos(x(2));

% Configrue the optimizer
% Display results at each iteration
options = optimset('Display','iter');

% Search for a minimum between -3 and 0
[x, val] = fminsearch(objFun, [0,0], options)
```

Configuring the Optimizer. Display Options

```
options = optimset('Display',value)
```

Value	Displays
'off'	Nothing
'notify'	A message only if the function fails
'final'	Only the final message
'iter'	Output at each iteration

Steps in optimization

1. Look at your problem: It is a zero finding problem or a minimization one? Does the problem have constraints?
2. Choose the adequate optimizer for the problem you have.
3. Configure the optimizer.
4. Define an initial guess for the solution.
5. Optimize it!.
6. Check if results are correct or if they look strange.

References

Hanselman, D. and Littlefield, B. (2012).
Mastering MATLAB.
Prentice Hall.

The MathWorks, Inc. (2012).
MATLAB — The Language of Technical Computing, Version R2012b (8.0).
Natick, Massachusetts.