Command *fsolve* in Matlab and Octave

fsolve solves a system of non-linear equations:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0}$$

⇨ vector **F**(**x**) of non-linear equations
⇨ vector **x** of arguments

e. g.: 2a) vector of 2 functions **F**(**x**) and 2 variables **x**=[x, y]

2b) vector of 3 functions **F**(**x**) and 3 variables $\mathbf{x}=[x_1, x_2, x_3]$

! number of equations = number of variables

How the command works

fsolve is an iterative method and solves for roots of functions, starting at a given initial point $\mathbf{x_0}$: solves for **x** that fulfills **F**(**x**) = **0** in several iteration steps

It could be that the (existing) solution is not found: change starting values

Syntax of *fsolve*:

```
x = fsolve(fcn, x0)

x = fsolve(fcn, x0, options)

[x, fval] = fsolve(fcn, x0)

...
```

Elements of *fsolve*:

output input (must be supplied)

[x, fval] = fsolve(fcn, x0, options)

x: vector with solution

fval: vector with function values at solution

- ⇨ fcn evaluated at x
- ⇨ fval = fcn(x)

fsolve: command for solving system of non-linear equations

fcn: vector with functions to solve

x0: vector of initial guess/ starting values for the arguments, used for iterating for the solution

options: to set optionally (more information follows)

e. g.: 2b) [x, fx] = fsolve(@(x) nonlinequB(x), x_0)

varying elements we want to solve for x_0=[1, 1, 1]

Command *options*:

define options before command fsolve

options = optimset( )

different parameters possible, e. g.:

display: default = off

choose iter, e. g.:

displays intermediate results for each loop iteration

positive integer

MaxFunEvals: maximal number of function evaluations

MaxIter: maximal number of algorithm iterations

in Example_4_5.m, e. g.:

options = optimset('MaxIter', 10000, 'MaxFunEvals', 90000);

(9 functions for 9 variables)

positive scalar

TolFun: termination criterion for function output

Difference between calculated function in iteration n+1 and calculated function in iteration n < TolFun

⇨ stop optimization

TolX: termination criterion for function input

Difference between x in iteration n+1 and x in iteration n < TolX

⇨ stop optimization

… (see links on problem set)