

Optimization-Based Modeling and Design of Electronic Circuits Assignment 3

Prof. J. E. Rayas-Sánchez March 2019

Develop a Matlab function to implement Broyden's method for solving systems of nonlinear equations.

The Matlab function should be terminated when a solution is found, when the relative change in the optimization variables is small enough, or when a maximum number of iterations is reached:

$$\|f(\mathbf{x}_{i+1})\|_{\infty} < \varepsilon_{\mathrm{f}} \lor \|\mathbf{x}_{i+1} - \mathbf{x}_{i}\|_{2} < \varepsilon_{\mathrm{x}} (\|\mathbf{x}_{i}\|_{2} + \varepsilon_{\mathrm{x}}) \lor i > i_{\mathrm{max}}$$

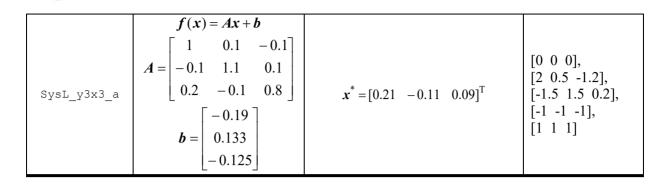
where $x_i \in \mathbb{R}^n$ is the vector containing the *n* optimization variables at the *i*-th iteration, and $f: \mathbb{R}^n \to \mathbb{R}^n$ is the system of nonlinear equations. Scalar limits are defined by ε_f , ε_x and i_{max} .

The implemented algorithms should have the following functionality:

```
% Usage: [x,f,i,FunEval,EF] = Broyden(fun,x0,MaxIter,epsf,epsx)
양
         fun: name of the multidimensional vector function (string). This
              function takes a vector argument of length n and returns a
용
응
              row vector of length n.
응
         x0: starting point (vector of length n).
         MaxIter: maximum number of iterations to find a solution.
응
응
         epsf: maximum acceptable error in the root of the function.
9
         epsx: minimum relative change in the optimization variables x.
응
         x: solution found for the nonlinear system of equations (vector
응
         of length n).
         f: function value at the solution (vector of length n).
ջ
응
         i: number of iterations needed to find the solution.
         FunEval: Number of function evaluations needed.
응
응
         EF: exit flag,
         EF=1: successful optimization (a root was found within epsf).
응
응
         EF=2: algorithm converged (relative change in x is small enough).
         EF=-1: maximum number of iterations exceeded.
```

Using $\varepsilon_f = \text{epsf} = 10^{-5}$, $\varepsilon_x = \text{epsx} = 10^{-6}$ and $i_{\text{max}} = \text{MaxIter} = 1000$, test your algorithm with at least the following systems of nonlinear equations and starting points:

Name	f(x) =	x^*	$oldsymbol{x_0}^{\mathrm{T}}$
SysNL_y2x2_a	$f(x) = \begin{bmatrix} x_1^2 - 4 \\ x_2^3 - 8 \end{bmatrix}$	$\mathbf{x}_a^* = \begin{bmatrix} 2 & 2 \end{bmatrix}^{\mathrm{T}}$ $\mathbf{x}_b^* = \begin{bmatrix} -2 & 2 \end{bmatrix}^{\mathrm{T}}$	[1 1], [-1 -1], [5 10], [-50 100], [1000 5000]
SysNL_y3x3_a	$f(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2^2 + x_2 \\ e^{x_3} - 1 \end{bmatrix}$	$\mathbf{x}_a^* = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^{\mathrm{T}}$ $\mathbf{x}_b^* = \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^{\mathrm{T}}$	[1 1 1], [1 0 -1], [-2 2 -2], [-3 0.1 2], [10 15 20]
SysNL_y2x2_b	$f(x) = \begin{bmatrix} x_1^2 + x_2^2 - 2 \\ e^{(x_1 - 1)} + x_2^3 - 2 \end{bmatrix}$	$\mathbf{x}_{a}^{*} = \begin{bmatrix} 1 & 1 \end{bmatrix}^{\mathrm{T}}$ $\mathbf{x}_{b}^{*} = \begin{bmatrix} -0.7137483 & 1.22088731 \end{bmatrix}^{\mathrm{T}}$	[0 0], [2 0.5], [-0.5 1.5], [1.5 -1.5], [-2 -2]
SysNL_y1x1_b	$f(x) = [(x_1 - 1)^2 + 2]$	$x^* = 1$	0, 1.5, -0.5, 2, 0.9



For each objective function and starting point x_0 , you must report the solution found x, the number of iterations needed (Iter), the number of function evaluations needed (FunEval), the exit flag value (EF), and the Euclidean norm of the error between the solution found and the corresponding exact solution. Summarize your results in one table per system of equations, with the following structure:

$\boldsymbol{x}_0^{\mathrm{T}}$	$\boldsymbol{x}^{\mathrm{T}}$	Iter	FunEval	EF	$\ \boldsymbol{x}^* - \boldsymbol{x}\ _2$

Solve the same problems using the Matlab command fminsearch on the same test cases (with the same starting points), adjusting the objective function accordingly. Make a comparison with your previous results.

Indicate the reference(s) used to write your algorithm.

Include in your report the conclusions.

Optional: Implement a globally convergent Newton method for solving systems of nonlinear equations, and test your implementation using the same cases described above, making a comparison with your previous results.

Submission deadline: March 20, 2019.