ADVNA HW4

Francesco Tomba

June 2021

Ex 1 & 2

Newton method was implemented in MATLAB following the requisites of the exercise, namely allowing to solve the linear system at each iteration choosing between a direct solution or an iterative one. The function was tested on the toy problem proposed in the exercise, results coincide with the ones in the slides.

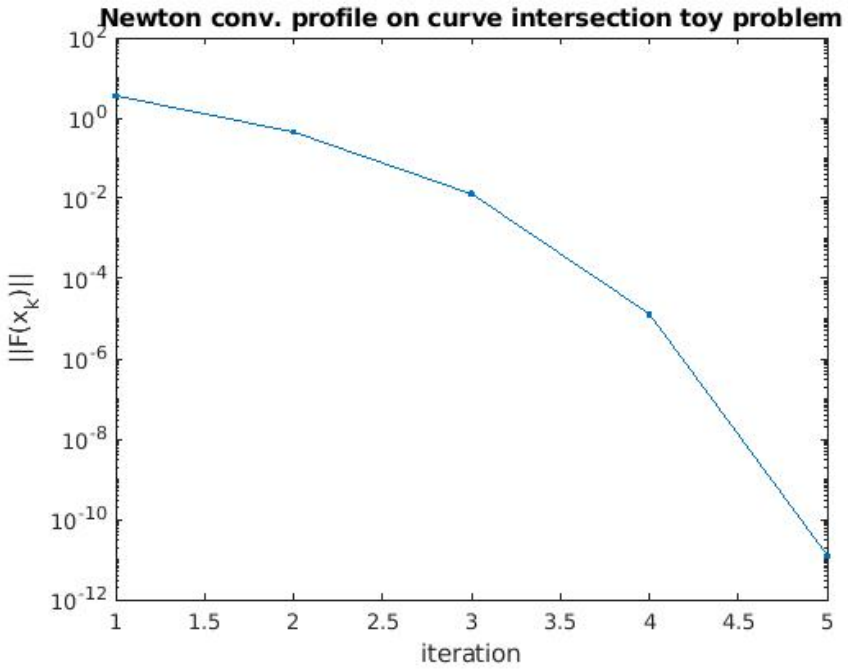

Figure 1: Convergence profile for newton method applied to the example in exercise 2

Ex 3

The newton method function was further tested next problem proposed, as expected, in this particular case, the convergence profiles of the newton method with the direct solver and the newton method with the iterative solver, roughly coincide.

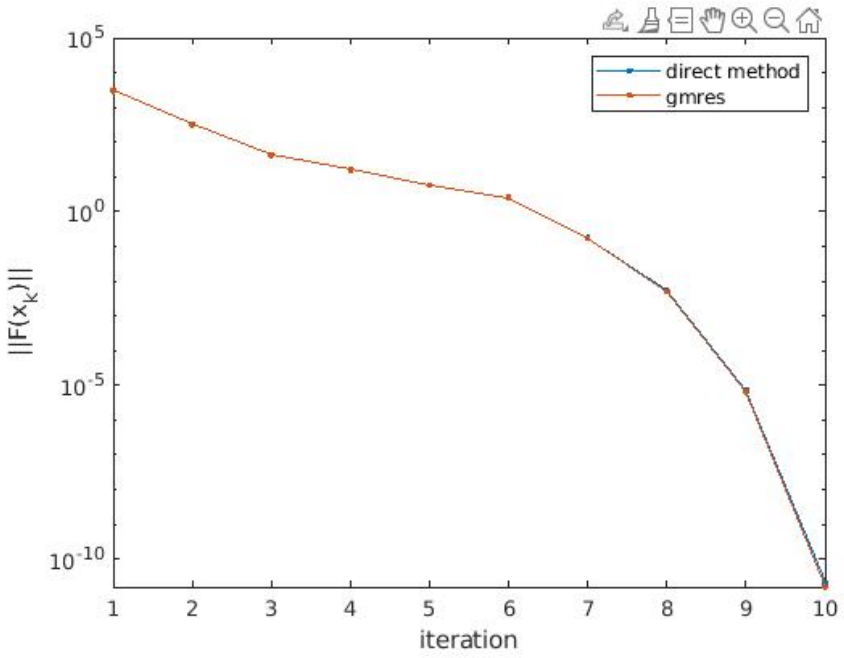

Figure 2: Convergence profile for newton method applied to the example in exercise 3

Ex 4

The eigenpair problem can be formulated as a non linear system of equations, as suggested by the exercise text, by using the eigenvalue definition and imposing the condition that the norm of the eigenvector associated to it is equal to 1.

This becomes the following system of equations:

$$F(\mathbf{x}) = \begin{cases} A\mathbf{u} - \lambda\mathbf{u} \\ \mathbf{u}^T\mathbf{u} - 1 \end{cases} \qquad \text{with } \mathbf{x} = \begin{pmatrix} u_1 \\ \vdots \\ u_n \\ \lambda \end{pmatrix} \tag{1}$$

Solving $F(\mathbf{x}) = 0$ leads to find the eigenpair $(\mathbf{u}, \lambda)$. The Jacobian of F, $F'(\mathbf{x})$

so it is:

$$F'(\mathbf{x}) = \begin{pmatrix} \frac{\partial F_1}{\partial \mathbf{u}} & \frac{\partial F_1}{\partial \lambda} \\ \frac{\partial F_2}{\partial \mathbf{u}} & \frac{\partial F_2}{\partial \lambda} \end{pmatrix} = \begin{pmatrix} A - \lambda \mathbf{I} & -\mathbf{u} \\ 2\mathbf{u}^T & 0 \end{pmatrix} \tag{2}$$

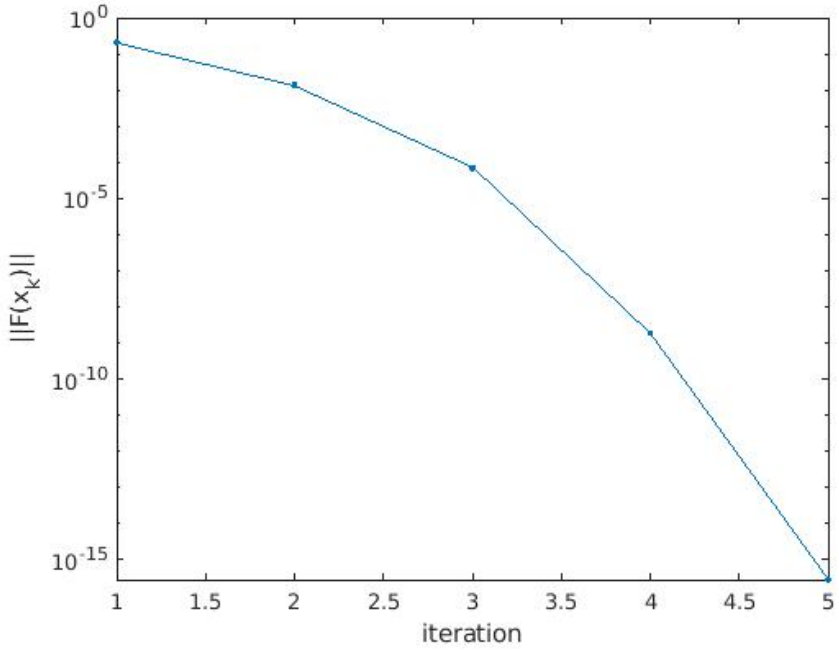

Figure 3: Convergence profile for newton method applied to the problem in exercise 4

The convergence in this case is quadratic at, each iteration the norm of the relative error halves it's order of magnitude, and since $||e_k|| \in O(||F(x_k)||)$ the convergence is quadratic. The solution obtained for the eigenvalue and eigenvalue is accurate up to machine precision, namely the norm of the distance between the eigenvalue obtained by using MATLAB's `eigs` and the one calculated by the method proposed is of the order of 10^{-15}, the distance between the true eigenvalue and the calculated one is 10^{-17}

To solve the next request of the exercise, the goal is to find the conditions on $(\mathbf{u}, \lambda)$ for which the Jacobian of the non linear system is non singular. Remind that a matrix M is singular whenever exists a vector $\mathbf{x}$ different from the null vector such that $M\mathbf{x} = 0$.

Now coming back to the eigenpair problem, lets suppose that the considered matrix A has 2 eigenvectors $\mathbf{u}$ and $\mathbf{v}$ which are orthogonal and associated to the same eigenvalue λ. Let consider now the following matrix vector multiplication $F'([u; \lambda])\mathbf{x}$ where $\mathbf{x} = (\mathbf{v}; 0)$.

$$F'([\mathbf{u};\lambda])\begin{pmatrix}\mathbf{v}\\0\end{pmatrix}=\begin{pmatrix}A-\lambda\mathbf{I} & -\mathbf{u}\\ 2\mathbf{u}^T & 0\end{pmatrix}\begin{pmatrix}\mathbf{v}\\0\end{pmatrix}=\begin{pmatrix}(A-\lambda\mathbf{I})\mathbf{v}\\ 2\mathbf{u}^T\mathbf{v}\end{pmatrix}=\begin{pmatrix}0\\0\end{pmatrix} \quad (3)$$

This is true since by hypotesis vector $\mathbf{v}$ is an eigenvalue associated to the same eigenvalue λ (first component) and vectors $\mathbf{u}$ and $\mathbf{v}$ are orthogonal (second component). Since the vector $(\mathbf{v};0)$ is different from the null vector we have found that the matrix F' in this case is singular.

So if $(\mathbf{u},\lambda)$ is an eigenpair, F' is non singular if the geometrical multiplicity of the eigenvalue λ considered is greater than 1, since if there are more than 1 eigenvectors which are linearly independent we can always find an orthogonal base to the eigenspace, and so we can always find a vector $\mathbf{v}$ which leads to the situation described before.

Ex 5

Exercise 5 request was about implementing Broyden quasi newton method as a MATLAB function, then to test it on the toy example proposed previously. As expected the method converges in 8 iterations.

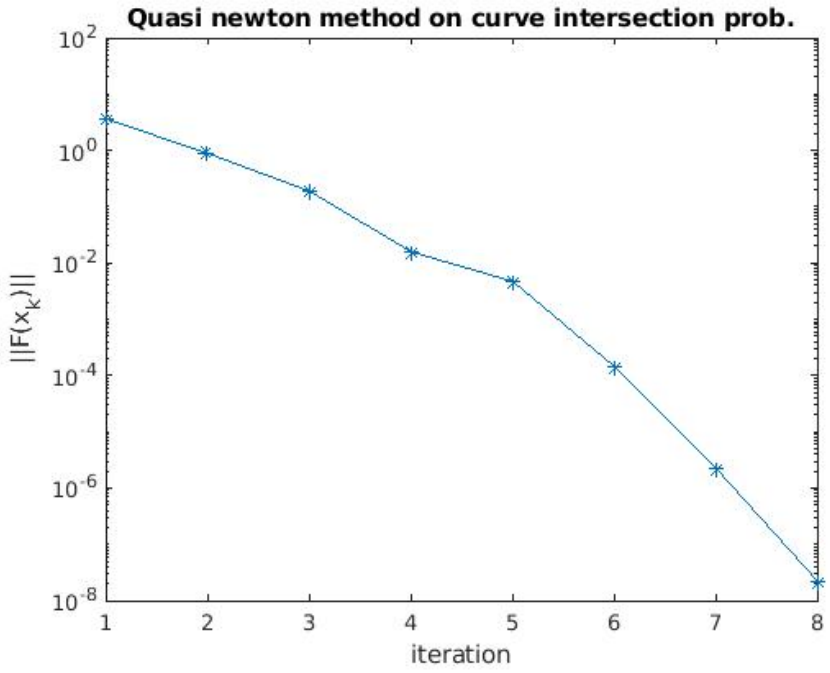

Figure 4: Convergence profile for newton method applied to the problem in exercise 5

Ex 6

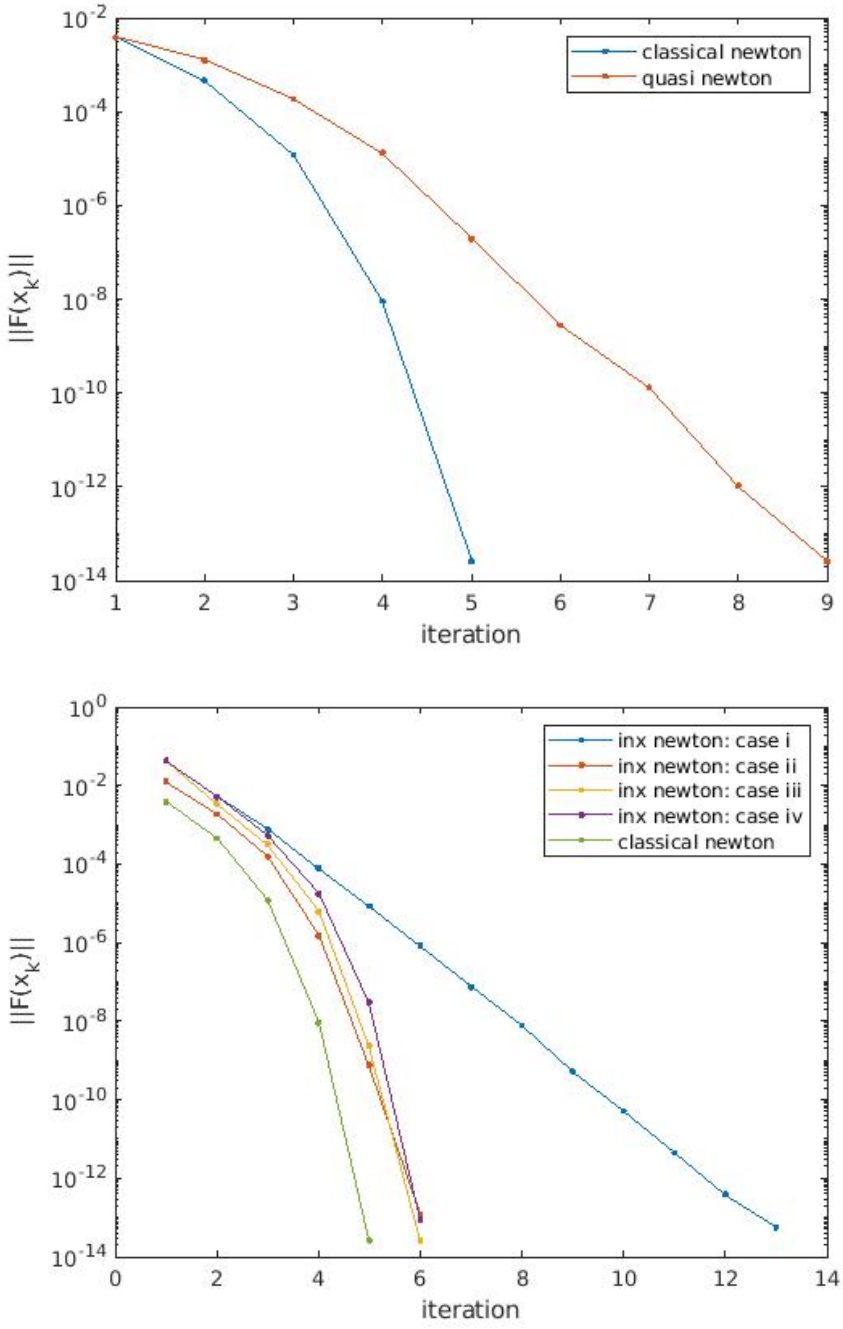

Figure 5: Convergence profile for newton and variants applied to the problem in exercise 6

Among the 4 stragegies suggested for choosing the sequence of tolerances $\{\eta_k\}$ case ii and case iv are the most performant, they achieve infact comparable numbers of linear iterations per cycle of the newton method and similar CPU times to get the solution. By looking at the following table it is usefull to point out that case iii is outperformed by case iv. This is due to the fact the choice of the tolerance sequence of case iii leads to, expecially in the last 2 newton iterations, to do more linear solver's iterations. To arrive to the same precision in the newton solution infact in case iv is sufficient a tolerance of 1^{-6}, but the rule used in case iii pushes η to be as low as 1^{-9}

```
/************************************
 * Classical Newton                 *
 ************************************/
iteration        ||F(x_k)||
   1             3.816426e-03
   2             4.582700e-04
   3             1.202953e-05
   4             9.160613e-09
   5             2.593566e-14
CPU TIME: 0.32
---------------------------------------

/************************************
 * Quasi Newton                     *
 ************************************/
iteration        ||F(x_k)||
   1             3.816426e-03
   2             1.233500e-03
   3             1.871887e-04
   4             1.321427e-05
   5             1.929930e-07
   6             2.781220e-09
   7             1.314042e-10
   8             1.045817e-12
   9             2.581881e-14
CPU TIME: 31.81
---------------------------------------

/************************************
 * Inexact Newton: case i           *
 ************************************/
iteration        ||F(x_k)||             eta                Lin IT
   1             4.299113e-02          1.000000e-01          57
   2             5.070946e-03          1.000000e-01          75
   3             7.789167e-04          1.000000e-01          81
   4             7.834967e-05          1.000000e-01          91
```

```
   5             8.447385e-06          1.000000e-01          108
   6             8.277306e-07          1.000000e-01          123
   7             7.399353e-08          1.000000e-01          135
   8             7.729037e-09          1.000000e-01          147
   9             5.143953e-10          1.000000e-01          172
   10             5.196642e-11          1.000000e-01          192
   11             4.459544e-12          1.000000e-01          204
   12             3.948264e-13          1.000000e-01          218
   13             5.633118e-14          1.000000e-01          248
CPU TIME: 12.74
---------------------------------------

/*************************************
 * Inexact Newton: case ii           *
 *************************************/
iteration        ||F(x_k)||             eta               Lin IT
   1             1.278180e-02          3.333333e-02          70
   2             1.890620e-03          1.111111e-02          88
   3             1.542341e-04          3.703704e-03          106
   4             1.487156e-06          1.234568e-03          129
   5             7.669397e-10          4.115226e-04          166
   6             1.168785e-13          1.371742e-04          228
CPU TIME: 4.23
---------------------------------------

/*************************************
 * Inexact Newton: case iii          *
 *************************************/
iteration        ||F(x_k)||             eta               Lin IT
   1             4.299113e-02          1.000000e-01          57
   2             3.385208e-03          4.084157e-02          80
   3             3.223568e-04          3.215947e-03          103
   4             6.114200e-06          3.062390e-04          131
   5             2.368724e-09          5.808490e-06          193
   6             2.541551e-14          2.250288e-09          317
CPU TIME: 5.30
---------------------------------------

/*************************************
 * Inexact Newton: case  iv          *
 *************************************/
iteration        ||F(x_k)||             eta               Lin IT
   1             4.299113e-02          1.000000e-01          57
   2             5.070946e-03          1.000000e-01          75
   3             5.327198e-04          1.321733e-02          93
   4             1.777660e-05          1.048440e-02          108
```

```
   5             2.996232e-08          1.057848e-03          141
   6             8.743039e-14          2.698840e-06          235
CPU TIME: 3.89
---------------------------------------
```