SHANGHAI JIAO TONG UNIVERSITY, UM-SJTU JOINT INSTITUTE

Introduction to Numerical Analysis Project2

Yu Cang
018370210001

August 1, 2018

1 TASK 1

Note the symmetry of x_1, for any solution (a,b), $(-a,b)$ is also valid.
If no special claim, the norm function for a vector is taken as the 2-norm.

(a) For fixed-point iteration, the iteration function G is taken as

$$G \triangleq \begin{bmatrix} \sqrt{x_2} \\ \sqrt{1-x_1^2} \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = x \tag{1.1}$$

The iteration converges to $(0.7862, 0.6180)$ with intial guess $x_{init} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$ and tolerance set to 10^{-10}.

(b) The newton iteration is implemented as mentioned in the project sheet, each time the linear system

$$J_F(x_k)\,w_k = y_k \tag{1.2}$$

is solved with the built-in function 'linsolve' inside matlab.
The iteration converges to $(0.7862, 0.6180)$ with intial guess $x_{init} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$ and tolerance set to 10^{-10}.

(c) The input of the Broyden's method is a little bit different from that in newton or fixed-point. The initial Jacobi matrix should be provided, denoted as A_0. In my implementation, it is given as

$$A_0 = J_F(x_0) \tag{1.3}$$

An extra calculation on x_1 should be done before the iteration loop as the iteration process requires both x_0 and x_1. It is approximated through newton's method in my implementation as

$$x_1 \approxeq x_0 - A_0^{-1} F(x_0) \tag{1.4}$$

After all the preparation work, the iteration loop can be carried out as mentioned in the project sheet, and the key part is the application of Sherman-Morrison's formula to calculated the inverse of A_k iteratively as

$$A_k^{-1} = A_{k-1}^{-1} + \frac{(s_k^T A_{k-1}^{-1} y_k) s_k^T A_{k-1}^{-1}}{s_k^T A_{k-1}^{-1} y_k} \tag{1.5}$$

where $s_k = x_k - x_{k-1}$ and $y_k = F(x_k) - F(x_{k-1})$. And the iteration of x_k is given as

$$x_{k+1} = x_k - A_k^{-1} F(x_k) \tag{1.6}$$

The iteration converges to $(0.7862, 0.6180)$ with intial guess $x_{init} = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$ and tolerance set to 10^{-10}.

(d) With numerical tests, newton's method out-performs the other two in terms of time cost.

2 Task 2

With trival inequality analysis, the exact solution is given as $x = [0.5, 0, -\frac{\pi}{6}]$, which can be used for varification.
In general, the initial value for such a non-linear system is hard to know, even the range for each component is hard to estimate. In this case, the downhill simplex method can be applied to produce a fine inital guess for further calculation like newton or quasi-newton.
Numerical experiments were done to identify the difference in terms of time cost.

	Without initialization	With initialization
fixed-point	$4.033ms$	$3.970ms$
neweon	$4.661ms$	$0.679ms$
broyden	$4.447ms$	$0.799ms$

It can be observed from the table that the downhill simplex initialization process does help a lot to reduce time cost for further newton or broyden calculation as it provide a better initial guess.
However, the time cost is almost equivalent for the fixed-point iteration, which indicates that it does not benefit from the downhill simplex initialization.

3 Task 3

(a)

(b)

(c)

(d)

(e)