Problem Set 3 - ECON 880

Spring 2022 - University of Kansas

Gunawan, Minh Cao

February 22, 2022

Problem 1

In this exercise, we are interested in solving $Ax = b$, where

$$A = \begin{pmatrix} 54 & 14 & -11 & 2 \\ 14 & 50 & -4 & 29 \\ -11 & -4 & 55 & 22 \\ 2 & 29 & 22 & 95 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

using Gauss-Jacobi and Gauss-Seidel method. Both methods yield the same result

$$x = \begin{pmatrix} 0.0189 \\ 0.0168 \\ 0.0234 \\ -0.0004 \end{pmatrix}.$$

Gauss-Jacobi method required 0.0207 seconds with 45 iterations until convergence. The residual is given by

$$10^{-11} \times \begin{pmatrix} 0.0361 \\ -0.1211 \\ -0.0922 \\ 0.2351 \end{pmatrix}$$

Gauss-Seidel method required 0.0193 seconds with 23 iterations until convergence. The residual is given by

$$10^{-12} \times \begin{pmatrix} 0.3191 \\ -0.5796 \\ -0.3766 \\ 0 \end{pmatrix}$$

Problem 2

In this exercise, we are interested in solving $Bq = r$ using fixed-point iteration and extrapolation, where

$$B = \begin{pmatrix} 1 & 0.5 & 0.3 \\ 0.6 & 1 & 0.1 \\ 0.2 & 0.4 & 1 \end{pmatrix}, \quad r = \begin{pmatrix} 5 \\ 7 \\ 4 \end{pmatrix}.$$

2(a)

We use Gauss-Jacobi fixed-point iteration method analogue to Problem 1, and obtain the solution

$$q = \begin{pmatrix} 1.6716 \\ 5.8651 \\ 1.3196 \end{pmatrix}$$

after 98 iterations, with the residual

$$Bq - r = 10^{-13} \times \begin{pmatrix} -0.4174 \\ -0.3908 \\ -0.3375 \end{pmatrix}.$$

2(b)

Following Ken Judd's definition[†], we first define $G = I - B$, and run the following iteration

$$q^{k+1} = \omega G q^k + \omega r + (1-\omega) q^k,$$

where we pick $\omega = 1.05$, tolerance level 10^{-13}, and initial value $q_0 = (0,0,0)'$. The extrapolation converged after $k = 97$ iterations, with the residual

$$Bq - r = 10^{-12} \times \begin{pmatrix} 0.1670 \\ -0.2371 \\ 0.1279 \end{pmatrix}.$$

The solution to the linear equation system is

$$q = \begin{pmatrix} 1.6716 \\ 5.8651 \\ 1.3196 \end{pmatrix}$$

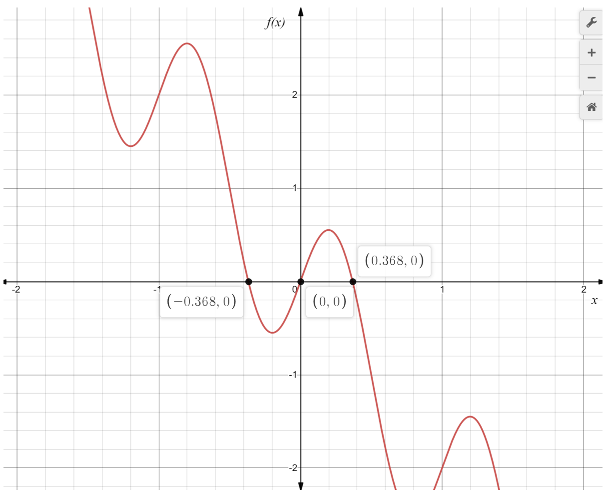
Problem 3

We want to solve the following functions

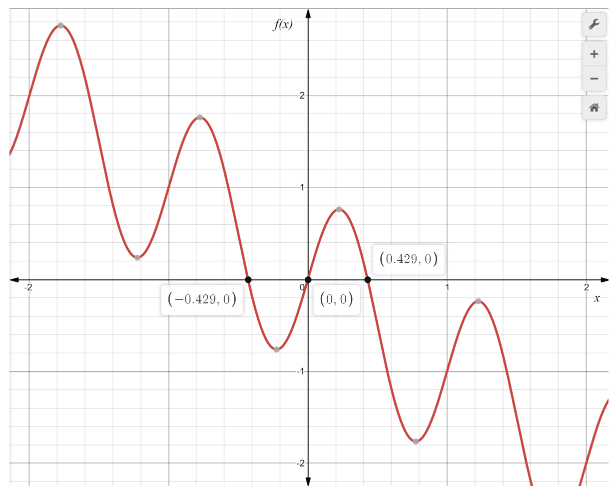
1. $\sin(2\pi x) - 2x = 0$
2. $\sin(2\pi x) - x = 0$
3. $\sin(2\pi x) - 0.5x = 0$

using 1) Bisection, 2) Newton method, 3) Secant method, and 4) fixed-point iteration. We want to evaluate for what value of initial guess $x_0 \in [-2, 2]$ these methods converge. We proceed by creating a grid of 20 points within the range $x_0 \in [-2, 2]$, and use them as initial values for the algorithms. For evaluation purposes, we plot all the three functions on Figure 1 to see how many and where the roots are located. From these graphs, we see that within the interval $[-2, 2]$, function 1 and 2 have both two roots, while function 3 has seven roots.

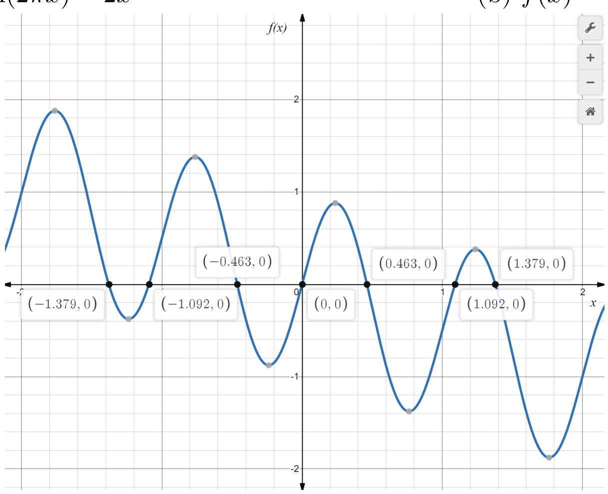
[†]Kenneth L. Judd, 1998. "Numerical Methods in Economics," MIT Press Books, The MIT Press, p.78-79



(a) $f(x) = \sin(2\pi x) - 2x$



(b) $f(x) = \sin(2\pi x) - x$



(c) $f(x) = \sin(2\pi x) - 0.5x$

Figure 1: Function plots for Problem 3

3(a) Bisection

The result for Bisection algorithm is summarized in Table 1. In order for this algorithm to work, we need to pick two values x_{low} and x_{high} so that $f(x_{low}) \cdot f(x_{high}) < 0$. The NaN values are produced because this condition is not satisfied.

		Roots for		
x_{low}	x_{high}	$f_1(x)$	$f_2(x)$	$f_3(x)$
-2	-1.7895	NaN	NaN	NaN
-2	-1.5789	NaN	NaN	NaN
-2	-1.3684	NaN	NaN	-1.3789
-2	-1.1579	NaN	NaN	-1.3789
-2	-0.94737	NaN	NaN	NaN
-2	-0.73684	NaN	NaN	NaN
-2	-0.52632	NaN	NaN	NaN
-2	-0.31579	-0.36824	-0.42937	-1.3789
-2	-0.10526	-0.36824	-0.42937	-0.46283
-2	0.10526	NaN	NaN	NaN
-2	0.31579	NaN	NaN	NaN
-2	0.52632	-0.36824	-0.42937	-0.46283
-2	0.73684	0.36824	0.42937	0.46283
-2	0.94737	0.36824	0.42937	0.46283
-2	1.1579	-0.36824	-0.42937	NaN
-2	1.3684	-0.36824	-0.42937	NaN
-2	1.5789	-0.36824	-0.42937	-1.3789
-2	1.7895	-0.36824	-0.42937	-0.46283
-2	2	0	0	0
1.7895	2	NaN	NaN	NaN
1.5789	2	NaN	NaN	NaN
1.3684	2	NaN	NaN	1.3789
1.1579	2	NaN	NaN	1.3789
0.94737	2	NaN	NaN	NaN
0.73684	2	NaN	NaN	NaN
0.52632	2	NaN	NaN	NaN
0.31579	2	0.36824	0.42937	1.3789
0.10526	2	0.36824	0.42937	0.46283
-0.10526	2	NaN	NaN	NaN
-0.31579	2	NaN	NaN	NaN
-0.52632	2	0.36824	0.42937	0.46283
-0.73684	2	-0.36824	-0.42937	-0.46283
-0.94737	2	-0.36824	-0.42937	-0.46283
-1.1579	2	0.36824	0.42937	NaN
-1.3684	2	0.36824	0.42937	NaN
-1.5789	2	0.36824	0.42937	1.3789
-1.7895	2	0.36824	0.42937	0.46283
-2	2	0	0	0

Table 1: Results for Bisection

3(b) Newton Method

The result for Newton method is summarized in Table 2. In Newton method, we feed an initial value x_0 to the algorithm, let the it run, and obtain a root estimate at the end.

	Roots for		
x_0	$f_1(x)$	$f_2(x)$	$f_3(x)$
-2	0.36824	0	NaN
-1.7895	-0.36824	0.42937	NaN
-1.5789	0.36824	-0.42937	NaN
-1.3684	-0.36824	0.42937	NaN
-1.1579	-0.36824	0.42937	-1.0919
-0.94737	0.36824	-0.42937	-1.0919
-0.73684	0.36824	0.42937	0.46283
-0.52632	-0.36824	-0.42937	-0.46283
-0.31579	-0.36824	-0.42937	-0.46283
-0.10526	0	0	0
0.10526	0	0	0
0.31579	0.36824	0.42937	0.46283
0.52632	0.36824	0.42937	0.46283
0.73684	-0.36824	-0.42937	-0.46283
0.94737	-0.36824	0.42937	1.0919
1.1579	0.36824	-0.42937	1.0919
1.3684	0.36824	-0.42937	NaN
1.5789	-0.36824	0.42937	NaN
1.7895	0.36824	-0.42937	NaN
2	-0.36824	0	NaN

Table 2: Results for Newton method

3(c) Secant Method

The result for Secant method is summarized in Table 3. For this method, we pick x_{k-h} and x_{k+h} to estimate $f'(x) \approx \frac{f(x_{k+h})-f(x_{k-h})}{x_{k+h}-x_{k-h}}$. This method is analogous to Newton method, except we use the approximated $f'(x)$, instead of using the analytical one.

3(d) Fixed Point Iteration Method

The result for fixed point iteration method is summarized in Table 4. For this method, we rewrite the functions to obtain the iteration function for x:

1. $x^{k+1} = 0.5\sin(2\pi x^k)$
2. $x^{k+1} = \sin(2\pi x^k)$
3. $x^{k+1} = 2\sin(2\pi x^k)$

Comment: We limit the iteration number to 10^6 for this method, and we observe that the calculated roots are mostly incorrect. We may need to define x differently to obtain better results.

x_{k-h}	x_{k+h}	Roots for $f_1(x)$	$f_2(x)$	$f_3(x)$
-2	-1.7895	0.36824	0.42937	1.0919
-1.7895	-1.5789	0.36824	-0.42937	-1.0919
-1.5789	-1.3684	-0.36824	-4.5918e-41	NaN
-1.3684	-1.1579	0.36824	0.42937	NaN
-1.1579	-0.94737	0	0	-1.0919
-0.94737	-0.73684	0	-0.42937	-1.0919
-0.73684	-0.52632	-0.36824	-0.42937	-0.46283
-0.52632	-0.31579	-0.36824	-0.42937	-0.46283
-0.31579	-0.10526	-0.36824	0.42937	0.46283
-0.10526	0.10526	0	0	0
0.10526	0.31579	0.36824	-0.42937	-0.46283
0.31579	0.52632	0.36824	0.42937	0.46283
0.52632	0.73684	0.36824	0.42937	0.46283
0.73684	0.94737	0	0.42937	1.0919
0.94737	1.1579	-3.7616e-37	-9.6296e-35	1.0919
1.1579	1.3684	-6.163e-33	0.42937	NaN
1.3684	1.5789	1.8808e-37	0.42937	NaN
1.5789	1.7895	0.36824	0.42937	1.0919
1.7895	2	0.36824	-3.7616e-37	-1.0919

Table 3: Results for Secant method

x_0	Roots for $f_1(x)$	$f_2(x)$	$f_3(x)$
-2	0.074224	-0.050641	-1.995
-1.7895	0.38421	-0.17081	1.8144
-1.5789	0.0017095	-0.64403	1.868
-1.3684	-0.11799	0.82832	-1.3745
-1.1579	-0.45458	-0.27138	0.80653
-0.94737	0.37138	-0.0022826	-0.85803
-0.73684	0.46711	0.52722	0.68693
-0.52632	0.26376	0.24965	-1.0318
-0.31579	-0.012863	0.013494	-0.086307
-0.10526	-0.2114	-0.94452	-0.0064801
0.10526	0.2114	0.94452	0.0064801
0.31579	0.012863	-0.013494	0.086307
0.52632	-0.26376	-0.24965	1.0318
0.73684	-0.46711	-0.52722	-0.68693
0.94737	-0.37138	0.0022826	0.85803
1.1579	0.45458	0.27138	-0.80653
1.3684	0.11799	-0.82832	1.3745
1.5789	-0.0017095	0.64403	-1.868
1.7895	-0.38421	0.17081	-1.8144
2	-0.074224	0.050641	1.995

Table 4: Results for fixed point iteration method

Problem 4

1. The linear convergence rate is defined by:

$$\lim_{n\to\infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|} \leq \beta < 1$$

for some β.
We can rewrite the equation above:

$$|x_{k+1} - x^*| \leq \beta |x_k - x^*|$$

Also, We can write the above equation for x_k instead of x_{k+1}:

$$|x_k - x^*| \leq \beta |x_{k-1} - x^*|$$

Write recursively:

$$|x_{k+1} - x^*| \leq \beta |x_k - x^*| \cdots \leq \beta^{n+1} |x_0 - x^*|$$

Hence, the above in equality is the necessary condition for the bisection method to be linearly convergence. In fact, we only can prove the necessary condition, but not the sufficient condition.

2. The bisection method create the nested sequence as follow:

$$[a_n, b_n] \in [a_{n-1}, b_{n-1}] \cdots [a_0, b_0]$$

By the construction, we have:

$$a = a_0 \leq a_1 \leq a_2 \leq \cdots \leq a_n \cdots \leq b_n \leq \cdots \leq b_2 \leq b_1$$

3. $x_n = \frac{a_n + b_n}{2}$, also $a_n \leq x^* \leq b_n$ for all n. It implies:

$$|x_n - x^*| \leq \frac{a_n + b_n}{2} - a_n = \frac{b_n - a_n}{2}$$

By induction:

$$|x_n - x^*| \leq \frac{b_n - a_n}{2} = \frac{b_{n-1} - a_{n-1}}{2^2} = \frac{b_{n-2} - a_{n-2}}{2^3} = \frac{b_0 - a_0}{2^{n+1}}$$

let $\beta = \frac{1}{2}$,$|x_0 - x^*| = \frac{b_0 - a_0}{2}$, we can prove the necessary condition for the linearly convergence of the bisection method.

4. We can also create a subsequence from the sequence e_n (the error of the bisection method) so that $e_{n+1} < \beta e_n$ for all n, for some $\beta \in (0, 1)$. Then the bisection method is linearly convergence.

Problem 5

The Newton method is given by:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

If $f(x_k) < 0$,

1. Case 1: $x_k < x^*$
   Assume $f'(x_k) \neq 0$

2. Case 1.1: $f'(x_k) > 0$ Hence, by the Newton method recursive equation, we have:

   $$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

   Since $f(x_k) < 0$ and $f'(x_k) > 0 \implies \frac{f(x_k)}{f'(x_k)} < 0 \implies -\frac{f(x_k)}{f'(x_k)} > 0$
   Let $-\frac{f(x_k)}{f'(x_k)} = \epsilon$
   The Newton equation can be written as: $x_{k+1} = x_k + \epsilon$ for some $\epsilon > 0$, so $x_{k+1} > x_k$ and $f'(x_k) > 0$ the function is strictly increasing, implies $f(x_{k+1}) > f(x_k)$, which is somehow lead us to closer to the root of $f(x)$.

3. Case 1.2 $f'(x_k) < 0$
   Since $f(x_k) < 0$ and $f'(x_k) < 0 \implies \frac{f(x_k)}{f'(x_k)} > 0 \implies -\frac{f(x_k)}{f'(x_k)} < 0$
   Let $-\frac{f(x_k)}{f'(x_k)} = \epsilon$
   The Newton equation can be written as: $x_{k+1} = x_k + \epsilon$ for some $\epsilon < 0$, so $x_{k+1} < x_k$ and $f'(x_k) < 0$ the function is strictly decreasing, implies $f(x_{k+1}) > f(x_k)$, which is somehow lead us to closer to the root of $f(x)$. if the function does not have the local maximum around x_k.
   Symmetric argument for the case $x_k > x^*$

Discussion

In the argument above, we rely on the fact that the function f is piecewise monotonic, but in some case, f may behave so that Newton method cannot lead us to the true answer. However, after each iteration, the Newton method at least in the worst case can guide us to the local maximum (in this question) of the function (i.e near some x' such that $f(x')$ near zero).