Solutions for Homework 8 Foundations of Computational Math 1 Fall 2012

Problem 8.1

Textbook, page 330, Problem 6

Solution:

We are given $G : \mathbb{R}^3 \to \mathbb{R}^3$ to define the iteration used to solve a system of nonlinear equations:

$$x^{(k+1)} = G(x^{(k)})$$

$$\begin{pmatrix} x_1^{(k+1)} \\ x_2^{(k+1)} \\ x_3^{(k+1)} \end{pmatrix} = \begin{pmatrix} -\frac{1}{81}\cos x_1^{(k)} + \frac{1}{9}(x_2^{(k)})^2 + \frac{1}{3}\sin x_3^{(k)} \\ \\ \frac{1}{3}\sin x_1^{(k)} + \frac{1}{3}\cos x_3^{(k)} \\ \\ -\frac{1}{9}\cos x_1^{(k)} + \frac{1}{3}x_2^{(k)} + \frac{1}{6}\sin x_3^{(k)} \end{pmatrix}$$

We know there is a fixed point at

$$x^* = \begin{pmatrix} 0 \\ \frac{1}{3} \\ 0 \end{pmatrix}$$

To show that there is a nontrivial region around x^* in which convergence occurs for any $x^{(0)}$ we check the sufficient condition that the spectral radius of the Jacobian of G is strictly less than one at the fixed point. We have

$$J_G(x) = \begin{pmatrix} \frac{1}{81}\sin x_1 & \frac{2}{9}x_2 & \frac{1}{3}\cos x_3 \\ \\ \frac{1}{3}\cos x_1 & 0 & -\frac{1}{3}\sin x_3 \\ \\ \frac{1}{9}\sin x_1 & \frac{1}{3} & \frac{1}{6}\cos x_3 \end{pmatrix}$$

$$J_G(x^*) = \begin{pmatrix} 0 & \frac{2}{27} & \frac{1}{3} \\ \\ \frac{1}{3} & 0 & 0 \\ \\ 0 & \frac{1}{3} & \frac{1}{6} \end{pmatrix}$$

Examining the Gershgorin row disks yields

- Row 1:

$$|\lambda| \leq \frac{11}{27}$$

- Row 2:
$$|\lambda| \leq \frac{1}{3}$$
- Row 3:
$$|\lambda - \frac{1}{6}| \leq \frac{1}{3}$$

Since all the eigenvalues are in disks that are inside the unit circle in the complex plane, we have $|\lambda| < 1$ for all eigenvalues of $J_G(x^*)$. Therefore, $\rho(G(x)) < 1$ and $G(x)$ is a contraction mapping in a nontrivial region around x^*.

An alternative sufficient condition is to examine a matrix norm of the Jacobian. Evaluating $\|J_G(x^*)\|_\infty$ as the maximum of the vector 1-norm of the rows yields

$$\|J_G(x^*)\|_\infty = \max\left(\frac{11}{27}, \ \frac{1}{3}, \ \frac{1}{2}\right) < 1$$

Therefore, since the spectral radius is bounded by all norms we have, $\rho(G(x)) < 1$ and $G(x)$ is a contraction mapping in a nontrivial region around x^*.

Problem 8.2

Consider the system of equations in $\mathbb{R}^2$

$$\xi^2 + \eta^2 = 4$$
$$e^\xi + \eta = 1$$

The system has two solutions in $\mathbb{R}^2$, one with $\xi > 0$ and $\eta < 0$ and one with $\xi < 0$ and $\eta > 0$.

(8.2.a) Derive the iteration for Newton's method to solve the system of two nonlinear equations above.

(8.2.b) Implement it and consider the performance for a variety of initial conditions to solve the system of two nonlinear equations above.

(8.2.c) Derive an iteration using Nonlinear Jacobi-Newton (one-step) to solve the system of two nonlinear equations above.

(8.2.d) Implement it and consider the performance for a variety of initial conditions to solve the system of two nonlinear equations above.

(8.2.e) Derive an iteration using Nonlinear Gauss-Seidel-Newton (one-step) to solve the system of two nonlinear equations above.

(8.2.f) Implement it and consider the performance for a variety of initial conditions to solve the system of two nonlinear equations above.

(8.2.g) Compare to the work required to achieve a given accuracy Nonlinear Jacobi-Newton and Gauss-Seidel-Newton, and Newton's Method when starting at the same initial condition? Also compare to iterations 1 and 2 in the latest programming assignment.

Note you do not have to turn in any implementation. This is not a programming assignment.

Solution:

Newton's Method: We have

$$F(\xi,\eta) = \begin{pmatrix} f_1(\xi,\eta) \\ f_2(\xi,\eta) \end{pmatrix} = \begin{pmatrix} \xi^2+\eta^2-4 \\ e^\xi+\eta-1 \end{pmatrix}$$

The Jacobian requires the partial derivatives

$$\frac{\partial f_1}{\partial \xi} = 2\xi \qquad \frac{\partial f_1}{\partial \eta} = 2\eta$$

$$\frac{\partial f_2}{\partial \xi} = e^\xi \qquad \frac{\partial f_2}{\partial \eta} = 1$$

These partial derivatives are also useful for the Nonlinear-Jacobi-Newton and Nonlinear-Gauss-Seidel-Newton iterations.

Newton's method is therefore

$$\begin{pmatrix} \xi_{k+1} \\ \eta_{k+1} \end{pmatrix} = \begin{pmatrix} \xi_k \\ \eta_k \end{pmatrix} - \begin{pmatrix} 2\xi & 2\eta \\ e^\xi & 1 \end{pmatrix}^{-1} \begin{pmatrix} f_1(\xi_k,\eta_k) \\ f_2(\xi_k,\eta_k) \end{pmatrix}$$

which of course is computed by solving a linear system and not explictly inverting the Jacobian. Note that the Jacobian is singular when $\xi e^{-\xi} = \eta$, e.g., $(\xi,\eta) = (0,0)$.

Nonlinear Jacobi-Newton: There are two choices here. The first is

$$f_1(\tau,\eta_k) = 0 \to \xi_{k+1} = \tau$$
$$f_2(\xi_k,\tau) = 0 \to \eta_{k+1} = \tau$$

which is then "solved" with one step of scalar Newton on each equation using $\partial f_1/\partial \xi$ and $\partial f_2/\partial \eta$ respectively yielding

$$\xi_{k+1} = \xi_k - \frac{f_1(\xi_k, \eta_k)}{\partial f_1/\partial \xi(\xi_k, \eta_k)} = \xi_k - \frac{\xi_k^2 + \eta_k^2 - 4}{2\xi_k}$$

$$\eta_{k+1} = \eta_k - \frac{f_2(\xi_k, \eta_k)}{\partial f_2/\partial \eta(\xi_k, \eta_k)} = \eta_k - (e^{\xi_k} + \eta_k - 1)$$

The second possibility is

$$f_2(\tau, \eta_k) = 0 \to \xi_{k+1} = \tau$$
$$f_1(\xi_k, \tau) = 0 \to \eta_{k+1} = \tau$$

which is then "solved" with one step of scalar Newton on each equation using $\partial f_2/\partial \xi$ and $\partial f_1/\partial \eta$. respectively yielding

$$\xi_{k+1} = \xi_k - \frac{f_2(\xi_k, \eta_k)}{\partial f_2/\partial \xi(\xi_k, \eta_k)} = \xi_k - \frac{e^{\xi_k} + \eta_k - 1}{e^{\xi_k}}$$

$$\eta_{k+1} = \eta_k - \frac{f_1(\xi_k, \eta_k)}{\partial f_1/\partial \eta(\xi_k, \eta_k)} = \eta_k - \frac{\xi_k^2 + \eta_k^2 - 4}{2\eta_k}$$

Nonlinear Gauss-Seidel-Newton: There are two choices here. The first is

$$f_1(\tau, \eta_k) = 0 \to \xi_{k+1} = \tau$$
$$f_2(\xi_{k+1}, \tau) = 0 \to \eta_{k+1} = \tau$$

which is then "solved" with one step of scalar Newton on each equation using $\partial f_1/\partial \xi$ and $\partial f_2/\partial \eta$ respectively yielding

$$\xi_{k+1} = \xi_k - \frac{\xi_k^2 + \eta_k^2 - 4}{2\xi_k}$$

$$\eta_{k+1} = \eta_k - (e^{\xi_{k+1}} + \eta_k - 1)$$

The second possibility is

$$f_2(\tau, \eta_k) = 0 \to \xi_{k+1} = \tau$$
$$f_1(\xi_{k+1}, \tau) = 0 \to \eta_{k+1} = \tau$$

which is then "solved" with one step of scalar Newton on each equation using $\partial f_2/\partial \xi$ and $\partial f_1/\partial \eta$. respectively yielding

$$\xi_{k+1} = \xi_k - \frac{e^{\xi_k} + \eta_k - 1}{e^{\xi_k}}$$
$$\eta_{k+1} = \eta_k - \frac{\xi_{k+1}^2 + \eta_k^2 - 4}{2\eta_k}$$

The complexity per step of these iterations (including those from the programming assignment for comparison) is approximately as follows:

- $G_1(x)$: 3 basic operations, 1 square root, 1 logarithm
- $G_2(x)$: 3 basic operations, 1 square root, 1 exponential
- Nonlinear Jacobi-Newton
  - version 1: 10 basic operations, 1 exponential
  - version 2: 11 basic operations, 1 exponential
- Nonlinear Gauss-Seidel-Newton
  - version 1: 10 basic operations, 1 exponential
  - version 2: 11 basic operations, 1 exponential
- Newton: 19 basic operations, 2 exponential
  - evaluate f_1 and f_2: 6 basic operations, 1 exponential
  - evaluate J: 2 basic operations, 1 exponential
  - compute LU: 3 basic operations
  - solve with L: 2 basic operations
  - solve with U: 4 basic operations
  - update x:2 basic operations

So assuming that a square root is about 5 basic operations and that the logarithm and exponential require the same number of operations, say, T we have

- $G_1 \approx G_2 : 8 + T$
- $NJN \approx NGN : 11 + T$
- Newton : $19 + 2T$

and therefore $N \approx 2NJN \approx 2.5Basic$. Note however when comparing these it is a bit more complicated than just requiring Newton to converge 2.5 times faster than the basic iterations. This is more or less correct for low to moderate accuracy demands. But recall that in the last few steps of Newton it gains accuracy very rapidly so adding just 1 or 2 steps can double or quadruple the accuracy achieved. As a result for high accuracy it is reasonable to consider a hybrid algorithm that uses a cheap iteration early and then when near convergence switch to Newton or Newton-like method to justify the extra work with rapid increase in accuracy. The difficulty of course is knowing when to switch.

Newton's method will converge to either root if started close enough to the desired root. The Jacobi-Newton and Gauss-Seidel-Newton iterations above each converge to one of the roots, i.e., for each method one choice converges to the left root and one choice converges to the right root.

An example of the convergence of Jacobi–Newton version 1 is given in the following table:

k	ξ_k	η_k	$f(x)$
0	3.0000000000000	-3.0000000000000	21.3247391099454
1	0.6666666666667	-19.0855369231877	361.1579032406471
2	-269.8599564014366	-0.9477340410547	72821.2942948456184
3	-134.9357252560221	1.0000000000000	18204.6499503686791
4	-67.4789790317444	1.0000000000000	4550.4126111666028
5	-33.7617186607337	1.0000000000000	1136.8536469265316
6	-16.9252883481865	1.0000000000000	283.4653856692574
7	-8.5512689566755	0.9999999553891	70.1242006804467
8	-4.4510470460175	0.9998067003450	16.8114371588341
9	-2.5625663054064	0.9883336546272	3.5441536877801
10	-1.8711599064779	0.9228933927861	0.3612382808661
11	-1.7768409739172	0.8460550035810	0.1279364425072
12	-1.8125861773961	0.8308282780812	0.0249725235539
13	-1.8192770928835	0.8367685550131	0.0100101155071
14	-1.8165422826151	0.8378570771506	0.0018834373674
15	-1.8160384830952	0.8374130401185	0.0007481281086
16	-1.8162432221918	0.8373311082493	0.0001411573832
17	-1.8162809850307	0.8373644095221	0.0000561080469
18	-1.8162656319772	0.8373705509877	0.0000105842904
19	-1.8162628004632	0.8373680541099	0.0000042068890

An example of the convergence of GS–Newton version 1 is given in the following table:

k	ξ_k	η_k	$f(x)$
0	3.0000000000000	-3.0000000000000	21.3247391099454
1	0.6666666666667	-0.9477340410547	2.6573557429817
2	2.6596834739030	-13.2917646661754	179.7449241223404
3	-31.1309773107379	1.0000000000000	966.1377483216759
4	-15.6136721731887	0.9999998343969	240.7867584006025
5	-7.9029057458625	0.9996303321824	59.4551800290062
6	-4.1413032454067	0.9840978863624	14.1188412203592
7	-2.4366657938969	0.9125480512757	2.7700841370341
8	-1.8682488727146	0.8456062115913	0.2054037154812
9	-1.8132766154096	0.8368812173101	0.0116577441223
10	-1.8164911674595	0.8374047292483	0.0008868420260
11	-1.8162470589161	0.8373650335088	0.0000664216353
12	-1.8162653443237	0.8373680073282	0.0000049806881

Examples of the quadratic convergence for Newton are seen in the following tables:

k	ξ_k	η_k	$f(x)$
0	3.0000000000000	-3.0000000000000	21.3247391099454
1	2.1264689951402	-1.5401976715265	6.5222557659637
2	1.4317499497180	-1.5598437751721	1.6963937582217
3	1.0828001964636	-1.7253081217009	0.2721375678425
4	1.0071547398278	-1.7295605361423	0.0100420515012
5	1.0041731878787	-1.7296372753518	0.0000150639414
6	1.0041687384846	-1.7296372870259	0.0000000000335

k	ξ_k	η_k	$f(x)$
0	-3.0000000000000	3.0000000000000	14.1492624198454
1	-2.7299011642368	0.9367655024299	4.3298904309113
2	-1.9548822406432	0.8842230843214	0.6039665272625
3	-1.8208047280871	0.8394360036047	0.0200269683066
4	-1.8162697132257	0.8373703851691	0.0000248890490
5	-1.8162640688339	0.8373677998953	0.0000000000386