# AMSC 460 Homework 7 Part 1

Jeffrey Zhang

May 8, 2022

10.1.4 The nonlinear system

$$5x_1^2 - x_2^2 = 0, \quad x_2 - 0.25\left(\sin x_1 + \cos x_2\right) = 0$$

has a solution near $\left(\frac{1}{4}, \frac{1}{4}\right)^t$.

a. Find a function $\mathbf{G}$ and a set $D$ in $\mathbb{R}^2$ such that $\mathbf{G} : D \to \mathbb{R}^2$ and $\mathbf{G}$ has a unique fixed point in $D$.

**Theorem 10.4:** Let $f$ be a function from $D \subset \mathbb{R}^n$ into $\mathbb{R}$ and $\mathbf{x}_0 \in D$. Suppose that all the partial derivatives of $f$ exist and constants $\delta > 0$ and $K > 0$ exist so that whenever $\|\mathbf{x} - \mathbf{x}_0\| < \delta$ and $\mathbf{x} \in D$, we have

$$\left|\frac{\partial f(\mathbf{x})}{\partial x_j}\right| \leq K, \quad \text{for each } j = 1, 2, \ldots, n \tag{1}$$

Then $f$ is continuous at $\mathbf{x}_0$.

**Definition 10.5:** A function $\mathbf{G}$ from $D \subset \mathbb{R}^n$ into $\mathbb{R}^n$ has a fixed point at $\mathbf{p} \in D$ if $\mathbf{G}(\mathbf{p}) = \mathbf{p}$.

The following theorem extends the Fixed-Point Theorem 2.4 on page 61 to the $n$ dimensional case. This theorem is a special case of the Contraction Mapping Theorem, and its proof can be found in [Or2], p. 153 .

Solving the $i$th equation for $x_i$ gives the fixed-point problem

$$x_1 = \frac{x_2}{\sqrt{5}}$$
$$x_2 = \frac{1}{4}\left(\sin(x_1) + \cos(x_2)\right)$$

Let $\mathbf{G} : \mathbb{R}^2 \to \mathbb{R}^2$ be defined by $\mathbf{G}(\mathbf{x}) = \left(g_1(\mathbf{x}), g_2(\mathbf{x})\right)^t$, where

$$g_1\left(x_1, x_2\right) = \frac{x_2}{\sqrt{5}}$$
$$g_2\left(x_1, x_2\right) = \frac{1}{4}\left(\sin(x_1) + \cos(x_2)\right)$$

Solving the partial derivatives we get:

$$\frac{\partial g_1}{\partial x_1} = 0 \quad \frac{\partial g_2}{\partial x_1} = \frac{\cos(x_1)}{4} \quad \frac{\partial g_1}{\partial x_2} = \frac{1}{\sqrt{5}} \quad \frac{\partial g_2}{\partial x_2} = \frac{-\sin(x_2)}{4}$$

If we consider the domain:

$$D = \left\{ (x_1, x_2)^t \mid 0 \le x_1, x_2 \le 1 \right\}$$

this implies:

$$\left|\frac{\partial g_1}{\partial x_1}\right| = 0 \quad \left|\frac{\partial g_2}{\partial x_1}\right| \le \frac{1}{4} \quad \left|\frac{\partial g_1}{\partial x_2}\right| = \frac{1}{\sqrt{5}} \quad \left|\frac{\partial g_2}{\partial x_2}\right| \le \frac{1}{4}$$

b. Apply fixed-point iteration to approximate the solution to within $10^{-5}$ in the $l_\infty$ norm.

**Theorem 10.6:** Let $D = \left\{ (x_1, x_2, \ldots, x_n)^t \mid a_i \le x_i \le b_i \right.$, for each $i = 1, 2, \ldots, n\}$ for some collection of constants $a_1, a_2, \ldots, a_n$ and $b_1, b_2, \ldots, b_n$. Suppose $\mathbf{G}$ is a continuous function from $D \subset \mathbb{R}^n$ into $\mathbb{R}^n$ with the property that $\mathbf{G}(\mathbf{x}) \in D$ whenever $\mathbf{x} \in D$. Then $\mathbf{G}$ has a fixed point in $D$.

Moreover, suppose that all the component functions of G have continuous partial derivatives and a constant $K < 1$ exists with

$$\left|\frac{\partial g_i(\mathbf{x})}{\partial x_j}\right| \le \frac{K}{n}, \qquad \text{whenever } \mathbf{x} \in D,$$

for each $j = 1, 2, \ldots, n$ and each component function $g_i$. Then the fixed-point sequence $\left\{\mathbf{x}^{(k)}\right\}_{k=0}^{\infty}$ defined by an arbitrarily selected $\mathbf{x}^{(0)}$ in $D$ and generated by

$$\mathbf{x}^{(k)} = G\left(\mathbf{x}^{(k-1)}\right), \qquad \text{for each } k \ge 1$$

converges to the unique fixed point $\mathbf{p} \in D$ and

$$\left\|\mathbf{x}^{(k)} - \mathbf{p}\right\|_\infty \le \frac{K^k}{1-K} \left\|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\right\|_\infty. \tag{2}$$

From the solution above, we can deduce that $K = \frac{2}{\sqrt{5}}$ since $n = 2$. If we set $x_1 = 0.25$ and $x_2 = 0.25$, we see that $k \approx 106$ from the theorem above, which means we need less than or equal to 106 fixed point iterations. Thus:

| $k$ | $x_1^k$ | $x_2^k$ | $\left\|\left\|x^k - x^{k-1}\right\|\right\|_\infty$ |
|---|---|---|---|
| 0 | 0.2500000 | 0.2500000 | |
| 1 | 0.1118034 | 0.3040791 | 0.1381966 |
| 2 | 0.1359883 | 0.2664234 | 0.0376557 |
| 3 | 0.1191482 | 0.2750721 | 0.0168401 |
| 4 | 0.1230160 | 0.270380 | 0.002161 |
| 5 | 0.1208899 | 0.2709848 | 0.0006132 |
| 6 | 0.1214881 | 0.2711679 | 0.0002742 |
| 7 | 0.1212700 | 0.2710876 | 0.0002742 |
| 8 | 0.1212341 | 0.2710876 | 0.0000819 |
| 9 | 0.1214623 | 0.2711133 | 0.0000359 |
| 10 | 0.1212456 | 0.2711027 | 0.000045 |
| 11 | 0.1212408 | 0.2711062 | 0.000048 |

Hence, the solution of the non-linear equation is

$$\boxed{x_1 = 0.12124 \text{ and } x_2 = 0.2711}$$

10.2.1d Use Newton's method with $\mathbf{x}^{(0)} = \mathbf{0}$ to compute $\mathbf{x}^{(2)}$ for each of the following nonlinear systems.

$$5x_1^2 - x_2^2 = 0$$
$$x_2 - 0.25\left(\sin x_1 + \cos x_2\right) = 0$$

The systems of equations would be:

$$f_1(x_1, x_2) = 5x_1^2 - x_2^2$$
$$f_2(x_1, x_2) = x_2 - 0.25\left(\sin x_1 + \cos x_2\right)$$

Thus the Jacobian Matrix is:

$$J = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix}$$
$$J = \begin{pmatrix} 10.0x_1 & -2.0x_2 \\ -0.25\cos\left(x_1\right) & 0.25\sin\left(x_2\right) + 1.0 \end{pmatrix}$$

Let the initial guess for the iteration be:

$$\mathbf{x}^{(0)} = (0, 0)^t$$

So, the function values are:

$$f = \begin{pmatrix} 0 \\ -0.25 \end{pmatrix}$$

The Jacobian Matrix will be:

$$J = \begin{pmatrix} 0 & 0 \\ -0.25 & 1 \end{pmatrix}$$

We can see that the Jacobian Matrix is Singular, hence the solution is not possible

10.4.5a Use the method of Steepest Descent to approximate minima to within 0.005 for the following functions.

$$g\left(x_1, x_2\right) = \cos\left(x_1 + x_2\right) + \sin x_1 + \cos x_2$$

Listing 1: Steepest Descent

```
def Steepest_Descent(x1,x2,TOL,N,g, grad_g):
    k = 1
    while(k <= N):
        g1 = g(x1,x2)
        z = grad_g(x1,x2)
        z0 = (z[0]**2 + z[1]**2)**(1/2)
        if z0 == 0:
            print("Zero gradient")
            return (x1,x2,g1)
        z[0] /= z0
        z[1] /= z0
```

```
        a1 = 0
        a3 = 1
        g3 = g(x1 - a3*z[0], x2 - a3*z[1])
        while(g3 >= g1):
            a3 = a3/2
            g3 = g(x1 - a3*z[0], x2 - a3*z[1])
            if a3 < TOL/2:
                print("No likely improvement")
                return (x1,x2,g1)
        a2 = a3/2
        g2 = g(x1 - a2*z[0], x2 - a2*z[1])
        h1 = (g2 - g1)/a2
        h2 = (g3 - g2)/(a3-a2)
        h3 = (h2 - h1)/a3
        a0 = 0.5*(a2 - h1/h3)
        g0 = g(x1 - a0*z[0], x2 - a0*z[1])
        a = a3 if g3 < g0 else a0
        x1 -= a*z[0]
        x2 -= a*z[1]
        print(x1,x2,g(x1,x2))
        if(abs(g(x1,x2) - g1) < TOL):
            print("Here")
            print(k)
            return (x1,x2,g(x1,x2))
        k += 1
    return(x1,x2, g(x1,x2))
```

Using this algorithm, the end result obtained is $\boxed{(5.215094930996798, -2.6111396030482363, -2.597781020758581)}$. The code is given in the txt file.