Course notes for EE394V Restructured Electricity Markets: Locational Marginal Pricing

Ross Baldick

www.ece.utexas.edu/~baldick/classes/394V/EE394V.html

2

Simultaneous equations

(i) Formulation,
(ii) Linear equations,
(iii) Non-linear equations,
(iv) Examples,
(v) Newton–Raphson algorithm,
(vi) Discussion of Newton–Raphson update,
(vii) Homework Exercises.

2.1 Formulation

- **Simultaneous equations** problems arise whenever there is a collection of conservation equations that must be satisfied:
  - the equations may be **linear** or **non-linear**,
  - in Section 3, we will formulate the solution of power flow as a simultaneous non-linear equations problem,
  - we will also describe an approach to approximating the solution of power flow as a simultaneous *linear* equations problem.
- The equations are specified in terms of a **decision vector** that is chosen from a **domain**.
- The domain will be n-dimensional Euclidean space $\mathbb{R}^n$, where:
  - $\mathbb{R}$ is the set of real numbers, and
  - $\mathbb{R}^n$ is the set of n-tuples of real numbers.

Formulation, continued

- We will usually use a symbol such as x to denote the decision vector:
  - entries of vectors such as x will be indexed by subscripts,
  - the k-th entry of the vector x is x_k,
  - in some problem formulations, such as offer-based economic dispatch in Section 8, it will be convenient to interpret x_k as itself a vector.
- In the discussion of simultaneous equations in this section and of optimization problems in Section 4, the vector x will be a generic decision vector and we will not explicitly specify the entries of x:
  - we will subsequently explicitly define the entries of x when we formulate specific problems such as power flow in Section 3 or economic dispatch in Section 5,
  - the definition of entries in the decision vector x will vary with the problem context and so the number of entries n in the decision vector x will also vary with the problem context.

Formulation, continued

- Consider a vector function g that takes values from a domain $\mathbb{R}^n$ and returns values of the function that lie in a **range** $\mathbb{R}^m$.
- We write $g : \mathbb{R}^n \to \mathbb{R}^m$ to concisely denote the domain and range of the function.
- Similarly to the decision vector, entries of vector functions such as g will be indexed by subscripts:
  - the ℓ-th entry of the vector function g is g_ℓ.
- Vector functions can be:
  - **linear**, of the form $\forall x, g(x) = Ax$, where $A \in \mathbb{R}^{m \times n}$ is a matrix,
  - **affine**, of the form $\forall x, g(x) = Ax - b$, where $A \in \mathbb{R}^{m \times n}$ is a matrix and $b \in \mathbb{R}^m$ is a vector,
  - **polynomial** or with some other specific functional form, or
  - **non-linear**, where there are no restrictions on g.
- As with the decision vector, in this section and in Section 4, the function g will be a generic vector function and we will not explicitly specify the entries of g (except in examples):
  - we will need to assume that we can partially differentiate g.

Formulation, continued

- Suppose we want to find a value $x^\star$ of the argument x that satisfies:

$$g(x) = \mathbf{0}. \tag{2.1}$$

- A value, $x^\star$, that satisfies (2.1) is called a solution of the **simultaneous equations** $g(x) = \mathbf{0}$:
  - we will use superscript $\star$ to indicate a desired or optimal value.
- We will typically assume that the number of equations, m, is the same as the number of entries, n, in the decision vector x.

2.2 Linear equations

2.2.1 Factorization and forwards and backwards substitution

- If g is affine, we usually re-arrange the equations as $Ax = b$:
  - called **linear simultaneous equations** and we will typically assume that $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, so that the number of equations, as specified by the number of entries in b and the number of rows in A, is the same as the number of entries in x.
  - such **square** systems are solved with **factorization** and **forwards** and **backwards** substitution,
  - will assume familiarity with solving linear equations using such **direct algorithms**.

Factorization and forwards and backwards substitution, continued

- Key computational issues with factorization and substitution are:
  - straightforward factorization of $A \in \mathbb{R}^{n \times n}$ requires computational effort on the order of n^3,
  - forwards and backwards substitution requires effort on the order of n^2,
  - although we will often write the solution of linear simultaneous equations as $x = A^{-1}b$, evaluating the inverse of a matrix requires significantly more computational effort than factorization and forwards and backwards substitution.

2.2.2 Modified factorization

- In some cases, we need to consider solutions of simultaneous equations where the coefficient matrix A is **modified**.
- The matrix $A+\gamma uv^\dagger$, where $\gamma\in\mathbb{R}, u,v\in\mathbb{R}^n$ with $\gamma\neq 0$ and $u,v\neq \mathbf{0}$, is called a **rank-one** modification of A.
- If a matrix A has already been factorized, then there are ways to evaluate the factors of $A+\gamma uv^\dagger$ with computational effort that is on the order of n^2.
- This is achieved by **modifying the factorization** of A and is also related to the **Sherman-Morrison** formula:

$$\begin{aligned}\left(A+\gamma uv^\dagger\right)^{-1} &= A^{-1}-\frac{A^{-1}\gamma uv^\dagger A^{-1}}{1+\gamma v^\dagger A^{-1}u},\\ &= A^{-1}\left(\mathbf{I}-\frac{\gamma uv^\dagger A^{-1}}{1+\gamma v^\dagger A^{-1}u}\right). \end{aligned} \tag{2.2}$$

Modified factorization, continued

- For example, to solve $(A+\gamma uv^\dagger)x = b$ for $x' = (A+\gamma uv^\dagger)^{-1}b$, we note by the Sherman-Morrison formula that:

$$
\begin{aligned}
(A+\gamma uv^\dagger)^{-1}b &= A^{-1}\left(\mathbf{I} - \frac{\gamma uv^\dagger A^{-1}}{1+\gamma v^\dagger A^{-1}u}\right)b, \\
&= A^{-1}b',
\end{aligned}
$$

- where:

$$
\begin{aligned}
b' &= \left(\mathbf{I} - \frac{\gamma uv^\dagger A^{-1}}{1+\gamma v^\dagger A^{-1}u}\right)b, \\
&= b + \Delta b',
\end{aligned}
$$

- where:

$$
\Delta b' = \left(-\frac{\gamma uv^\dagger A^{-1}}{1+\gamma v^\dagger A^{-1}u}\right)b.
$$

Modified factorization, continued

- Summarizing, $x' = (A + \gamma u v^{\dagger})^{-1} b$ can be evaluated using the following:
  (i) solve $Ax^{\star} = b$ and $Ax^{\star\star} = u$, so that $x^{\star} = A^{-1}b$ and $x^{\star\star} = A^{-1}u$,
  (ii) define:

$$\begin{aligned}\Delta b' &= -\frac{\gamma u v^{\dagger} A^{-1}}{1 + \gamma v^{\dagger} A^{-1} u} b = -\frac{\gamma u v^{\dagger}}{1 + \gamma v^{\dagger} x^{\star\star}} x^{\star}, \\ b' &= b + \Delta b' = \left(\mathbf{I} - \frac{\gamma u v^{\dagger} A^{-1}}{1 + \gamma v^{\dagger} A^{-1} u}\right) b,\end{aligned}$$

  (iii) either solve $Ax' = b'$ or solve $A\Delta x' = \Delta b'$ and set $x' = x^{\star} + \Delta x'$.
- Note that $x^{\star}$ is the solution of the original **base-case** equations $Ax = b$ and we may have already solved for $x^{\star}$ as part of previous calculations.
- Solving for x' in this way did not require factorization of the matrix $A + \gamma u v^{\dagger}$ and therefore reduced the computational effort from being on the order of n^3 to being on the order of n^2.

2.2.3 Sparsity

- Large-scale linear equations typically exhibit **sparsity**:
  - many of the entries in the matrix are zero, and
  - **sparsity techniques** allow this characteristic to be exploited to reduce computational effort compared to straightforward factorization and substitution.
- This means that factorization and substitution may take effort that is much less than n^3 and n^2, respectively.
- It is still generally computationally faster to factorize and use forwards and backwards substitution on a large sparse system than to invert the matrix.
- If a sparse matrix A has already been factorized, then to obtain a factorization of a modified matrix $A+\gamma uv^{\dagger}$ it is still generally computationally faster to modify the factorization of A than to factorize the modified matrix from scratch.

2.3 Non-linear equations

- If g is not affine, then the equations are non-linear.
- Non-linear equations usually require **iterative** algorithms, and we will briefly develop the Newton–Raphson algorithm:
  - requires an initial guess that is then iteratively improved,
  - we will focus on issues related to linearization that will be important in the context of understanding formulations and approximations used in power flow and electricity markets.

2.4 Examples

- Figure 2.1 shows the case of a function $g : \mathbb{R}^2 \to \mathbb{R}^2$.
- There are two sets illustrated by the solid curves.
- These two sets intersect at two points, $x^\star, x^{\star\star}$, illustrated as bullets $\bullet$.
- The points $x^\star$ and $x^{\star\star}$ are the two solutions of the simultaneous equations $g(x) = \mathbf{0}$, so that $\{x \in \mathbb{R}^n | g(x) = \mathbf{0}\} = \{x^\star, x^{\star\star}\}$.
- In general, simultaneous equations problems could have no solutions, one solution, or multiple solutions.

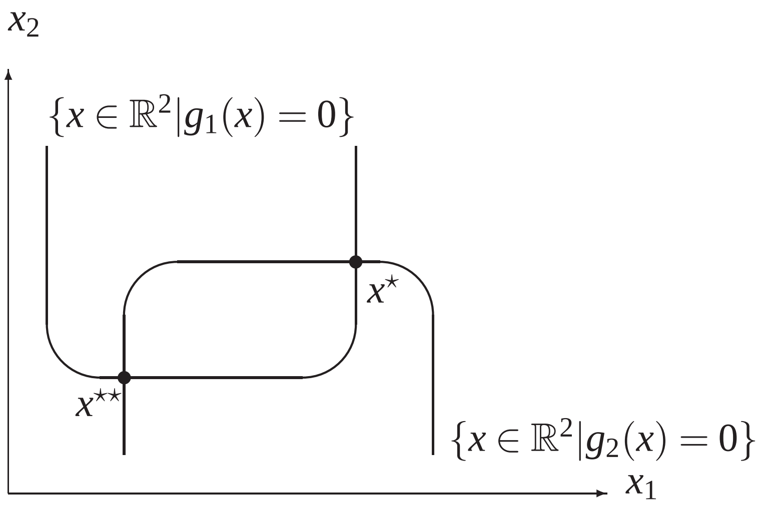

Fig. 2.1. Example of simultaneous equations and their solution.

Examples, continued

- As another example, let: $g : \mathbb{R}^2 \to \mathbb{R}^2$ be defined by:

$$\forall x \in \mathbb{R}^2, g(x) = \begin{bmatrix} (x_1)^2 + (x_2)^2 + 2x_2 - 3 \\ x_1 - x_2 \end{bmatrix}. \tag{2.3}$$

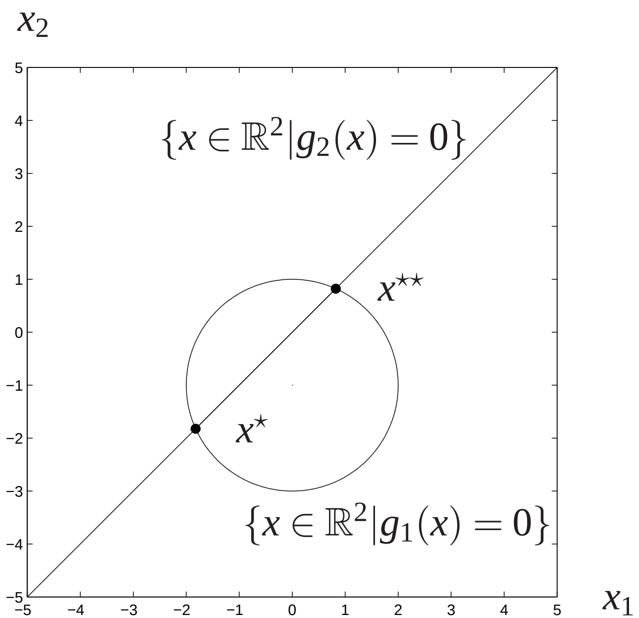

Fig. 2.2. Solution of non-linear simultaneous equations $g(x) = \mathbf{0}$ with g defined as in (2.3).

Examples, continued

- As a third example, let $g : \mathbb{R} \to \mathbb{R}$ be defined by:

$$\forall x \in \mathbb{R}, g(x) = (x-2)^3 + 1. \tag{2.4}$$

- By inspection, $x^\star = 1$ is the unique solution to $g(x) = 0$.

2.5 Newton–Raphson algorithm

- We now consider a general approach to solving simultaneous non-linear equations:

$$g(x) = \mathbf{0}, \tag{2.5}$$

- where $g : \mathbb{R}^n \to \mathbb{R}^n$ so that the number of entries in the decision vector is the same as the number of entries in the vector function:
  - there are the same number of variables as equations.

2.5.1 *Initial guess*

- We will distinguish successive iterates by superscript in parentheses.
- Let $x^{(0)}$ be the initial guess of a solution to (2.5).
- In general, we expect that $g(x^{(0)}) \neq \mathbf{0}$.
- We seek an updated value of the vector $x^{(1)} = x^{(0)} + \Delta x^{(0)}$ such that:

$$g(x^{(1)}) = g(x^{(0)} + \Delta x^{(0)}) = \mathbf{0}. \tag{2.6}$$

2.5.2 Taylor approximation

2.5.2.1 Scalar function

$$\begin{aligned} g_1(x^{(1)}) &= g_1(x^{(0)}+\Delta x^{(0)}), \text{ since } x^{(1)} = x^{(0)}+\Delta x^{(0)}, \\ &\approx g_1(x^{(0)}) + \frac{\partial g_1}{\partial x_1}(x^{(0)})\Delta x_1^{(0)} + \cdots + \frac{\partial g_1}{\partial x_n}(x^{(0)})\Delta x_n^{(0)}, \\ &= g_1(x^{(0)}) + \sum_{k=1}^{n} \frac{\partial g_1}{\partial x_k}(x^{(0)})\Delta x_k^{(0)}, \\ &= g_1(x^{(0)}) + \frac{\partial g_1}{\partial x}(x^{(0)})\Delta x^{(0)}. \end{aligned} \tag{2.7}$$

- In (2.7), the symbol "$\approx$" should be interpreted to mean that the difference between the expressions to the left and to the right of the $\approx$ is small compared to $\left\|\Delta x^{(0)}\right\|$.

Scalar function, continued

- The expression to the right of the $\approx$ in (2.7) is called a **first-order Taylor approximation** of g about $x^{(0)}$:

$$g_1(x^{(0)}) + \frac{\partial g_1}{\partial x}(x^{(0)})\Delta x^{(0)}.$$

- For a partially differentiable function g_1 with continuous partial derivatives, the first-order Taylor approximation about $x = x^{(0)}$ approximates the behavior of g_1 in the vicinity of $x = x^{(0)}$.
- The first-order Taylor approximation represents a plane that is **tangential** to the graph of the function at the point $x^{(0)}$.

- For example, suppose that $g_1 : \mathbb{R}^2 \to \mathbb{R}$ is defined by:

$$\forall x \in \mathbb{R}^2, g_1(x) = (x_1)^2 + (x_2)^2 + 2x_2 - 3.$$

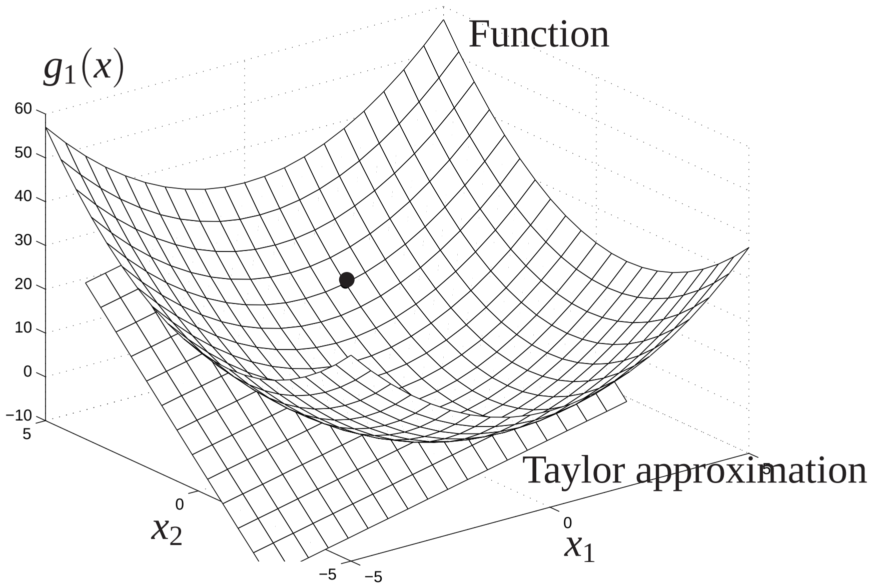

Fig. 2.3. Graph of function and its Taylor approximation about $x^{(0)} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$.

- For $x^{(0)} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$, $\Delta x^{(0)} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, and $g_1 : \mathbb{R}^2 \to \mathbb{R}$ defined by:

$$\forall x \in \mathbb{R}^2, g_1(x) = (x_1)^2 + (x_2)^2 + 2x_2 - 3,$$

evaluate:

$g_1(x^{(0)})$

$\frac{\partial g_1}{\partial x}(x^{(0)})$

$g_1(x^{(0)}) + \frac{\partial g_1}{\partial x}(x^{(0)})\Delta x^{(0)}$

$g_1(x^{(0)} + \Delta x^{(0)})$

2.5.2.2 *Vector function*

- We now consider the vector function $g : \mathbb{R}^n \to \mathbb{R}^n$.
- Since g is a vector function and x is a vector, the Taylor approximation of g involves the $n \times n$ matrix of partial derivatives $\frac{\partial g}{\partial x}$ evaluated at $x^{(0)}$.
- A first-order Taylor approximation of g about $x^{(0)}$ yields:

$$g(x^{(0)} + \Delta x^{(0)}) \approx g(x^{(0)}) + \frac{\partial g}{\partial x}(x^{(0)}) \Delta x^{(0)},$$

- where by the $\approx$ we mean that the norm of the difference between the expressions to the left and the right of $\approx$ is small compared to $\left\| \Delta x^{(0)} \right\|$.
- The first-order Taylor approximation again represents a "plane" that is tangential to the graph of the function; however, the situation is much more difficult to visualize for a vector function.

2.5.2.3 *Jacobian*

- The matrix of partial derivatives is called the **Jacobian** and we will usually denote it by $J(\bullet)$:
  - in some later development, we will need to consider particular sub-matrices of the Jacobian and we will also use the symbol J to denote particular sub-matrices.
  - the definition will be clear from the context.
- Using J to stand for the Jacobian, we have:

$$\begin{aligned} g(x^{(1)}) &= g(x^{(0)} + \Delta x^{(0)}), \text{by definition of } \Delta x^{(0)}, \\ &\approx g(x^{(0)}) + J(x^{(0)})\Delta x^{(0)}. \end{aligned} \tag{2.8}$$

- In some of our development, we will approximate the Jacobian when we evaluate the right-hand side of (2.8)
- In this case, the linear approximating function is no longer tangential to f.

2.5.3 Initial update

- Setting the right-hand side of (2.8) to zero to solve for $\Delta x^{(0)}$ yields a set of linear simultaneous equations:

$$J(x^{(0)})\Delta x^{(0)} = -g(x^{(0)}). \tag{2.9}$$

2.5.4 General update

$$\begin{aligned} J(x^{(\nu)})\Delta x^{(\nu)} &= -g(x^{(\nu)}), & (2.10) \\ x^{(\nu+1)} &= x^{(\nu)} + \Delta x^{(\nu)}. & (2.11) \end{aligned}$$

- (2.10)–(2.11) are called the **Newton–Raphson update**.
- $\Delta x^{(\nu)}$ is the **Newton–Raphson step direction**.
- Suppose that $g : \mathbb{R}^n \to \mathbb{R}^n$ is affine and suppose that $x^{(0)} \in \mathbb{R}^n$ is arbitrary. Use the Newton–Raphson update to obtain $x^{(1)}$. What can you say about $g(x^{(1)})$?

2.6 Discussion of Newton–Raphson update

- In principle, the Newton–Raphson update is repeated until a suitable **stopping criterion** is satisfied that is chosen to judge whether the solution is accurate enough.
- Issues:
  - (i) The need to calculate the matrix of partial derivatives and solve a system of linear simultaneous equations at each iteration can require considerable effort.
  - (ii) At some iteration we may find that the linear equation (2.10) does not have a solution, so that the update is not well-defined.
  - (iii) Even if (2.10) does have a solution at every iteration, the sequence of iterates generated may not converge to the solution of (2.5).

Discussion of Newton–Raphson update, continued

- Approximations and variations have been developed due to:
  - the computational effort of performing multiple iterations, and
  - the potential that the iterates fail to form a convergent sequence.
- One variation is to perform just *one* Newton–Raphson update starting from a suitable initial guess to obtain an approximate answer.
- We will develop this variation in the context of power flow because it:
  - is used in many electricity market models, and
  - sheds light on decomposition approaches even when the non-linear equations are being solved more accurately.

2.7 Summary

- In this section we considered solution of simultaneous linear and non-linear equations problems.
- We introduced the Sherman-Morrison formula.
- We considered linearization of a function.
- We developed the Newton–Raphson algorithm.

This chapter is based on Sections 2.1, 2.2, and 9.2 of *Applied Optimization: Formulation and Algorithms for Engineering Systems*, Cambridge University Press 2006.

Homework exercises

2.1 Consider the matrix $A = \begin{bmatrix} 2 & 3 & 4 \\ 7 & 6 & 5 \\ 8 & 9 & 11 \end{bmatrix}$ and the vector $b = \begin{bmatrix} 9 \\ 18 \\ 28 \end{bmatrix}$.

(i) Factorize this matrix into L and U factors. For example, you can use the MATLAB function `lu`. (Note that MATLAB will provide a factorization of the form $PA = LU$, where P is a permutation matrix.) If you have not studied LU factorization before, you should read through slides 37 to 61 of www.ece.utexas.edu/~baldick/classes/380N/Linear.pdf.

(ii) Solve $Ax = b$ (or $PAx = Pb$ using forwards and backwards substitution.)

Homework exercises, continued

2.2 This exercise concerns Taylor's theorem. Let $g : \mathbb{R}^2 \to \mathbb{R}^2$ be defined by:

$$\forall x \in \mathbb{R}^2, g(x) = \begin{bmatrix} \exp(x_1) - x_2 \\ x_1 + \exp(x_2) \end{bmatrix}.$$

(i) Use Taylor's theorem to linearly approximate $g(x+\Delta x)$ in terms of:

- $g(x)$,
- the Jacobian $J(x)$, and
- Δx.

Write out the linear approximation explicitly for the given g. That is, you must explicitly differentiate g to find the entries in J.

(ii) Calculate the difference between the exact expression for $g(x+\Delta x)$ and the linear approximation to it. Let us call this difference $e : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ defined by:

$$\forall x \in \mathbb{R}^2, \forall \Delta x \in \mathbb{R}^2, e(x, \Delta x) = g(x+\Delta x) - (\text{the linear approximation}).$$

(iii) Show that:

$$\frac{\|e(x,\Delta x)\|^2}{\|\Delta x\|^2} \leq \frac{\exp(2x_1)(\exp(\Delta x_1) - 1 - \Delta x_1)^2}{(\Delta x_1)^2} + \frac{\exp(2x_2)(\exp(\Delta x_2) - 1 - \Delta x_2)^2}{(\Delta x_2)^2}.$$

Use the norm given by: $\forall x \in \mathbb{R}^2, \|x\| = \sqrt{(x_1)^2 + (x_2)^2}$.

(iv) Show that $\|e(x,\Delta x)\| / \|\Delta x\| \to 0$ as $\|\Delta x\| \to 0$. Use the norm given by: $\forall x \in \mathbb{R}^2, \|x\| = \sqrt{(x_1)^2 + (x_2)^2}$. Be careful when proving this limit. (Hint: Consider $\|e(x,\Delta x)\|^2 / \|\Delta x\|^2$ and use the previous part together with l'Hôpital's rule to evaluate the limit of the ratio.)

2.3 In this exercise we will apply the Newton–Raphson update to solve $g(x) = \mathbf{0}$ where $g : \mathbb{R}^2 \to \mathbb{R}^2$ was specified by (2.3):

$$\forall x \in \mathbb{R}^2, g(x) = \begin{bmatrix} (x_1)^2 + (x_2)^2 + 2x_2 - 3 \\ x_1 - x_2 \end{bmatrix}.$$

(i) Calculate the Jacobian explicitly.
(ii) Calculate $\Delta x^{(\nu)}$ according to (2.10) in terms of the current iterate $x^{(\nu)}$.
(iii) Starting with the initial guess $x^{(0)} = \mathbf{0}$, calculate $x^{(1)}$ according to (2.10)–(2.11).
(iv) Calculate $x^{(2)}$ according to (2.10)–(2.11).
(v) Sketch g_1, $x^{(0)}$, $x^{(1)}$, and the first-order Taylor approximation to g_1 about $x^{(0)}$.
(vi) Sketch g_1, $x^{(1)}$, $x^{(2)}$, and the first-order Taylor approximation to g_1 about $x^{(1)}$.
(vii) Sketch, on a single graph, the points and functions in Parts (v) and (vi) versus x_1 along the "slice" where $x_1 = x_2$. Discuss the progress of the iterates.