ON THE APPLICATION OF A NEWTON RAPHSON’S ITERATIVE METHOD OF THE FIXED POINT THEORY TO THE SOLUTION OF A CHEMICAL EQUILIBRIUM PROBLEM

Okereke, C. Emmanuel, Ph.D. †

ABSTRACT

In this work, I discussed the solution of a chemical equilibrium problem aiming to obtain its fixed point. To do this, the preliminary and basic ideas introducing the fixed point theory was x-rayed and the Newton Raphson’s iterative method for solving the system of non-linear equations discussed, then the problem of the chemical equilibrium involving principal reactions in the production of synthesis gas by partial oxidation of methane with oxygen were stated.

Using a computer program the O reactant ratio that produce an adiabatic equilibrium temperature were obtained by developing a system of seven simultaneous nonlinear equations that has the form which we now solve using the Newton Raphson’s method described in section 2.2 and hence the desired fixed point of the chemical equilibrium problem.

THE NEWTON’S METHOD, A PRELIMINARY TO THE NEWTON RAPHSON’S ITERATIVE METHOD

Let T be an operator mapping a set X into itself, a point \( x \in X \) is called a fixed point of T if
\[
T(x) = x
\]
(1.1)
By (1.1), we achieve a natural construction of the method of successive approximations.
\[
x_{n+1} = T(x_n), \quad n \geq 0 \in X
\]
(1.2)
and if the sequence \( \{x_n\} \), \( n \geq 0 \) converges to some point \( x^* \in X \) for some initial guess \( x_0 \in X \), where T is a continuous operator in a Banach space X, we have
\[
x^* = \lim_{n \to \infty} x_{n+1} = \lim_{n \to \infty} T(x_n) = T\left(\lim_{n \to \infty} x_n\right)
\]
that is \( x^* \) is a fixed point of operations T. Hence we now state without proof the following important results that make easy the understanding of the Newton Raphson’s method used in this work

Theorem (1.1): If T is a continuous operator in a Banach X, \( \{x_n\} \), \( n \geq 0 \) generated by (1.2) converges to some point \( x^* \in X \) for some initial guess \( x_0 \in X \) and we say that \( x^* \) is a fixed point of the operator T [1].

To investigate the uniqueness property, we introduce the concept of contraction mapping as follows. Let \((x, \|\|)\) be a metric space and T a mapping of X into itself. The operator T is said to be a contraction if there exists a real number \( k \), \( 0 \leq k < 1 \) such that

\[
\|T(x) - T(y)\| \leq k \|x - y\|
\]
\[ \| F(x) - F(y) \| \leq k \| x - y \|, \text{ for all } x, y \epsilon x \quad \ldots \ (1.3) \]

Hence, every contraction mapping \( T \) is uniformly continuous. Indeed \( T \) is lipschitz continuous with a lipschitz constant \( k \) which may also be called the contraction constant for \( T \). With the above; we now discuss the Banach fixed point extensively as related to the target of this research.

**Theorem 1.1A** [9] (Banach fixed point theorem (1922). Suppose that we are given an operator \( T: M \subseteq X \rightarrow M \), i.e., \( M \) is mapped into itself by \( T \); \( M \) is a closed nonempty set in a complete metric space \((X, d)\);

\( T \) is \( k \)-contractive, i.e. \( d(Tx, Ty) \leq k \).

Then the following hold:

Existence and uniqueness:- \( T \) has exactly one fixed point on \( M \);

Convergence of the iteration: the sequence \( \{x_n\} \) of successive approximations converges to the solution, \( x \), for an arbitrary choice of initial point \( x_0 \) in \( M \);

Error estimate: for all \( n = 0, 1, 2, \ldots \) we have the a priori error estimate \( d(x_n, x) \leq k^n (1-k)^{-1} d(x_0, x) \), and the a posteriori error estimate \( d(x_{n+1}, x) \leq k(1-k)^{-1} d(x_n, x_{n+1}) \);

Rate of convergence: for all \( n = 0, 1, 2, \ldots \) we have \( d(x_{n+1}, x) \leq kd(x_n, x) \) (1.4)

\( T \) is called Lipschitz continuous and if \( d(Tx, Ty) < d(x, y) \) for all \( x, y \epsilon M \) with \( x \neq y \) … (1.4)

\( T \) is called contractive for \( T \) and we obviously have the implications:

- \( k \)-contractive \rightarrow contractive \rightarrow \text{n- expansive} \rightarrow \text{Lipschitz continuous}.

Every Banach space called the \((x, \| \|)\) also is a complete metric space as \((x, d)\) under \( d(x, y) = \| x-y \| \).

On a B-space, (1.3) therefore becomes

\[ \| Tx - Ty \| \leq k \| x-y \| \]

On the above the following follows

\( \{x_n\} \) is a Cauchy sequence. This follows from

\[ d(x_n, x_{n+1}) = d(Tx_{n-1}, Tx_n) \leq kd(x_{n-1}, x_n) \leq k^2d(x_{n-2}, x_{n-1}) \leq \ldots \leq k^nd(x_0, x_1) \]

\[ \ldots \leq k^n d(x_n, x_{n+1}) \]

(1.5)

Repeated application of the triangle inequality and finally summing the formula for a geometric series yields

\[ d(x_0, x_{n+m}) \leq d(x_0, x_{n+1}) + \ldots + d(x_{n+m-1}, x_{n+m}) \leq (k^n + k^{n+1} + \ldots + k^{n+m-1})d(x_0, x_1) \]

\[ \leq k^n (1-k)^{-1} d(x_0, x_1) \]

(1.6)

Since \( X \) is complete, the Cauchy sequence converges, i.e., \( x_n \rightarrow x \) as \( n \rightarrow \infty \) [3].

Equation (1.5) follows by letting \( m \rightarrow \infty \).

(II) The error estimate (1.6) follows by letting \( m \rightarrow \infty \) in

\[ d(x_{n+1}, x_{n+m+1}) \leq d(x_0, x_{n+1}) + \ldots + d(x_{n+m-1}, x_{n+m+1}) \leq (k^n + k^{n+1} + \ldots + k^m)d(x_0, x_{n+1}) \]
\[ \leq k(1-k)^n d(x_n, x_{n+m}) \ldots \]

(1.7)
The point \( x \) is a solution of (1.1) for \( T \) is continuous by (1.4). Since \( T(M) \subseteq M \) and \( x_0 \in M \), we have \( x_n \in M \) also, for all \( n \). since \( M \) is closed and \( x_n \rightarrow x \) as \( n \rightarrow \infty \), we get \( x \in M \). Equation (1.2) implies that \( Tx = x \) for \( n \rightarrow \infty \).

Equation (1.6) follows \( d(x_{n+1}, x) = d(Tx_n, Tx) \leq kd(x_n, x) \).

Uniqueness of solution. Suppose \( x = Tx \) and \( y = Ty \); then \( d(x, y) = d(Tx, Ty) \leq kd(x, y) \), which forces \( d(x, y) = 0 \), i.e. \( x = y \).

Continuous Dependence On A Parameter
It is important to note that in many applications, \( T \) depends on an additional parameter \( p \). then, (1.1) is replaced by the equation,

\[ x_p = T_p x_p, \quad x_p \in M, \quad p \in P \] (1.8)

where \( p \in P \).

**Proposition 1.2** (Corollary to Theorem 1.1 A.) let the following conditions be

P is a metric space, called the parameter space.

For each \( p \), the operator \( T_p \) satisfies the hypotheses of Theorem (1.1 A) but with \( k \) in (1.3) independent of \( p \).

For a fixed \( p_0 \in P \), and for all \( x \in M \), \( \lim_{p \to p_0} T_p x = T_{p_0} x \).

Then, for each \( p \in P \), (1.8) has exactly one solution \( x_p \in M \), and \( \lim_{p \to p_0} x_p = x_{p_0} \).

**Proof:** later \( x_p \) be the solution of (1.8) given by theorem 1.1 A, then

\[ d(x_p, x_{p_0}) = d(T_p x_p, T_{p_0} x_{p_0}) \]

\[ \leq d(T_p x_p, T_p x_{p_0}) + d(T_p x_{p_0}, T_{p_0} x_{p_0}) \]

\[ \leq kd(x_p, x_{p_0}) + d(T_p x_{p_0}, T_{p_0} x_{p_0}), \]

And therefore,

\[ d(x_p, x_{p_0}) < (1-k)^{-1}d(T_p x_{p_0}, T_{p_0} x_{p_0}) \rightarrow 0 \quad \text{as} \quad p \rightarrow p_0 \quad \text{by (iii).} \] [9]

1.5. Accelerated Convergence and Newton’s method [5]

We begin with the insight which underlines the acceleration of iterative methods. Let \( x \) be a solution of the real equation \( X = F(x) \), and suppose the sequence of iterations (x, where

\[ x_{n+1} = f(x_n) \ldots \]

(1.9)

And \( x_n \in [a, b] \) for all \( n \), converges to \( x \) as \( n \rightarrow \infty \).

Now for the key: Suppose further that \( f \) is \( m \) – times differentiable on \([a,b] \), with

\[ f'(x) = f''(x) = \ldots = f^{(m-1)} = 0 \]

(1.10)

![Figure 1.4](attachment:image.png)
Since \( x_{n+1} = f(x_n) \) and \( x = f(x) \), we have

\[
|x_{n+1} - x| \leq \sup_{a < \xi < b} f^{(m)}(\xi) \| x_n - x \| / m!
\]

(1.11)

If the supremum in (1.11) is finite, we obtain the convergence of order \( m \), as opposed to the linear convergence (\( m = 1 \)) of (1.9)

Example 1.1[10]. The trick to Newton’s method consists of rewriting the equation \( f(x) = 0 \) in the equivalent form \( X = F(x) \), where \( f(x) = x - f(x) / f'(x) \)

Then the iterative method becomes \( x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \)

We assume that \( f'(x_n) \neq 0 \) for all \( n \). Then, \( f(x) = f(x) f'(x) / f'(x)^2 \)

So that if \( x \) is a solution of \( f'(x) = 0 \) with \( f(x) \neq 0 \), then \( f(x) = 0 \). Thus we have a method with \( m = 2 \) in (1.1.9) i.e. we have quadratic convergence.

We apply this to the equation \( x = T(x) \) in (1.1.1) i.e. \( f(x) = T(x) \).

Computing, we obtain the iterative values \( x_n \) with linear convergences.

The geometric interpretation of Newton’s methods is seen in figure 1.4(a). To find a zero \( x \) of \( f \), take the initial value \( x_0 \) and determine the corresponding functional value, \( f(x_0) \). The next iterative value, \( x_1 \) is the intersection of the tangent line at \( (x_0, f(x_0)) \) and the x-axis. Keep repeating the process, it is typical of Newton’s method that it converges very rapidly if the initial value \( x_0 \) is already in the vicinity of the zero, but figure 1.4(b) shows a better of this.

![Figure 1.4](image-url)

However, we know that the above discussed fixed point method is just the traditional fixed point method that is restricted to the solution of only linear systems and for the purpose of this research we advance onto the modified Newton’s method which is the Newton Raphson’s iterative method here below generated for use in section three

2. **NEWTON’S RAPHRSON’S METHOD**

Sections 2 and 3 are concerned with finding the solution, or solutions, of the system.
Involving n real functions of the n real variables \( x_1, x_2, \ldots, x_n \). Following the previous notation, \( x = (x_1, x_2, \ldots, x_n)^T \), we shall write \( f_i(x) = f_i(x_1, x_2, \ldots, x_n) \) here, and in the subsequent development, \( 1 \leq i \leq n \). Then let \( \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T \) be a solution of (2.1), that is, let \( f_i(\alpha) = 0 \).

Let the \( n \) functions \( f_i(x) \) be such that \( x_i = F_i(x) \) \( (2.2) \) implies \( f_i(x) = 0, 1 \leq j \leq n \). Basically, the \( n \) equations (2.2) will constitute a suitable rearrangement of the original system (2.1). In particular, let \( \alpha_i = f_i(\alpha) \) \( \ldots \) \( (2.3) \)

Let the starting vector \( x_0 = [x_{10}, x_{20}, \ldots, x_{n0}]^T \) be an approximation to \( \alpha \). Define successive new estimates of the solution vector, \( x_k = [x_{1k}, x_{2k}, \ldots, x_{nk}]^T \), \( k = 1, 2, \ldots \) by computing the individual elements from the recursion relations.

\[
x_{ik} = F_i(x_{1k-1}, x_{2k-1}, \ldots, x_{nk-1}).
\]

\( \text{(2.4)} \)

suppose there is starting \( R \) describable as \( \{x_j - \alpha_j | \leq h, 1 \leq j \leq n \} \) and for \( x \) in \( R \) there is a positive number \( \mu \), less than one, such that

\[
\sum_{j=1}^{n} \left| \frac{\partial F_i(x)}{\partial x_j} \right| \leq \mu \quad \text{(2.5)}
\]

Then, if the starting vector \( x_0 \) lies in \( R \), we show that the iterative method expressed by (2.4) converges to a solution of the system (2.1), that is,

\[
\lim_{k \to \infty} x_k = \alpha \quad \text{(2.6)}
\]

Using the mean-value theorem, the truth of (2.1) is established by first noting from (2.3) and (2.4), that

\[
x_{ik} - \alpha_i = F_i(x_{1k-1}) - F_i(\alpha).
\]

\[
= \sum_{j=1}^{n} \left( x_{j,k-1} - \alpha_j \right) \frac{\partial F_i(\alpha + \xi_{j,k-1}(x_{j,k-1} - \alpha))}{\partial x_j},
\]

\( \text{(2.7)} \)

In which \( 0 < \xi_{j,k-1} < 1 \). that is,

\[
\left| x_{ik} - \alpha \right| \leq h \sum_{j=1}^{n} \frac{\left| \partial F_i \right|}{\partial x_j} \leq \mu h < h,
\]

\( \text{(2.8)} \)
Showing that the points \( x_k \) lie in \( R \). also, by induction, from (2.5) and (2.7),
\[
| x_{ik} - \alpha | \leq \mu \max (| x_{j,k-1} - \alpha_j |) \leq \mu^ht. \tag{2.9}
\]
Therefore, (2.6) is true, and the procedure converges to a solution of (2.1). Note that if the \( F_i(x) \) are linear, we have the Newton’s method, and the sufficient conditions of (2.5) are the same as the second set of sufficient conditions controlling the Newton’s iteration.

For the nonlinear equations, there is also a counterpart to the Newton’s method, previously discussed for the linear case. We proceed as before, except that some replacements are made by
\[
X_{ik} = F_i(x_{ik}, x_{2k}, \ldots, x_{i-1,k}, x_{i,k-1}, \ldots, x_{n,k-1}). \tag{2.10}
\]
That is, the most recently computed elements of the solution vector are always used in evaluating the \( F_i \). The proof of convergence according to (2.10) is much the same as for the jacobi-type iteration. We have
\[
\sum \left[ \alpha_i + \xi_{ik} (x_{1,k-1} - \alpha_1) + \xi_{2k} (x_{2,k-1} - \alpha_2) + \cdots + \xi_{nk} (x_{n,k-1} - \alpha_n) \right].
\]
It will appear inductively that the above is true, because the various points concerned remain in \( R \). If \( e_{k-1} \) is the largest of the numbers \( | x_{j,k-1} - \alpha_j | \), then
\[
x_{ik} - \alpha_{il} \leq \mu e_{k-1} < e_{k-1} < h.
\]
it follows that
\[
x_{2k} - \alpha_{2} = (x_{ik} - \alpha_1) \frac{\partial F_2 (\Sigma 2k)}{\partial x_j} + n \sum_{j=2}^{n} \left( x_{j,k-1} - \alpha_j \right) \frac{\partial F_2 (\Sigma 2k)}{\partial y_j},
\]
where
\[
e_{2k} = [\alpha_1 + \xi_{2k} (x_{1,k-1} - \alpha_1), \alpha_2 + \xi_{2k} (x_{2,k-1} - \alpha_2), \ldots, \alpha_n + \xi_{2k} (x_{n,k-1} - \alpha_n)].
\]
that is, \( l x_{2k} - \alpha_{2} \leq \mu e_{k-1} < e_{k-1} < h \).

Therefore, \( l x_{ik} - \alpha_{il} \leq \mu^t h \), and convergence according to (2.1) is again established.

Observe that the first of the sufficiency conditions of the same (2.10) has been reaffirmed under slightly general circumstance.

2.2 Newton-Raphson’s Iteration for Nonlinear Equations.

The equations to be solved are again those of (2.1), and we retain the nomenclature of the previous section. The Newton-Raphson process, to be described, is once more iterative in character. We first define
\[
f_j (x) = \frac{\partial f_i (x)}{\partial x_j} \quad \ldots \tag{2.11}
\]
Next define the matrix \( \phi (x) \) as
\[
\phi (x) = (f_i (x)), 1 \leq i \leq n, 1 \leq j \leq n. \quad \ldots \tag{2.12}
\]
Thus det \( (\phi (x)) \) is the Jacobian of the system (2.1) for the vector \( x = [x_1, x_2, \ldots, x_n] \). Now define the vector
\[
f (x) = [f_1 (x), f_2 (x), \ldots, f_n (x)] \tag{2.13}
\]

*Journal of Mathematical Sciences & Mathematics Education* Vol. 9 No. 2 6
With these definitions in mind, and with the starting vector
\[ x_0 = [x_{10}, x_{20}, \ldots, x_{n0}]^t, \] let \( x_{k+1} = x_k + \delta_k, \ldots \)                           (2.14)

The fundamental theorem concerning convergence is much less restrictive than those of the previous sections. We have the result that if the components of \( \phi (x) \) are continuous in a neighborhood of a point \( \alpha \) such that
\[ f (\alpha) = 0, \text{ if } \det (\phi (\alpha)) \neq 0, \text{ and if } x_0 \text{ is "near" } \alpha, \] then
\[ \lim_{k \to \infty} x_k = \alpha. \]                   (2.15)

An outline for a method of proof follows. By (2.13) and (2.14), since \( f_i (\alpha) = 0 \),
\[ \delta_k = \phi^{-1} (x_k) [f(\alpha)]. \] \ldots (2.15)

By the mean-value theorem,
\[ f_i (x_k) - f_i (\alpha) = \sum_{j=1}^n \left[ jf(\alpha + \xi_{ik} (x_k - \alpha))(x_{jk} - \alpha), \right. \]
where \( 0 < \xi_{ik} < 1. \)

For the ith row of a matrix \( \psi \) use \( [f_{11} (\alpha + \xi_{ik} (x_k - \alpha)), \ldots, f_{in} (\alpha + \xi_{ik} (x_k - \alpha))]. \) Then \( x_{i+1} = x_i - \alpha + \delta_k \) \( \phi^{-1} (x_k) \phi (x_k) - \psi \) \( (x_k - \alpha). \)

Since the entries in the matrix \( \phi (x_k) - \psi \) are differences of the type \( f_{ij} (x_k - \alpha)), \text{ they can be kept uniformly small if the starting vector } X_0 \text{ lies in an initially chosen region } R \text{ describable as } |x_i - \alpha_i| \leq h, 1 \leq i \leq n. \text{ concurrent with this is the fact that since } \det (\phi (x_k)) \text{ can be bounded from zero. The net result is that, for } 0 < \mu < 1, |x_k - \alpha_i| \leq h \mu^k, 1 \leq i \leq n. \text{ thus the sequence } [x_k] \text{ converges to } \alpha [6].

**Example (2.1) [7].** To illustrate the procedure, we use the example below namely.
\[ f_1 (x_1, x_2) = \frac{1}{2} \sin(x_1, x_2) - \frac{x_2}{4\pi} - \frac{x_1}{2} = 0 \]
\[ f_2 (x_1, x_2) = \left(1 - \frac{1}{4\pi} \right) (e^{x_1} - e) + \frac{e x_2}{\pi} - 2x_2 = 0. \] \ldots (2.16)

It is readily seen that
\[ \frac{\partial f_1}{\partial x_1} = \frac{1}{2} x_2 \cos(x_1, x_2), \quad \frac{\partial f_1}{\partial x_2} = -\frac{1}{4\pi} + \frac{x_1}{2} \cos(x_1, x_2), \]
\[ \frac{\partial f_2}{\partial x_1} = -2x_1 + \left(2 - \frac{1}{4\pi} \right) e^{2x_1}, \quad \frac{\partial f_2}{\partial x_2} = \frac{e}{\pi}. \]

The increments \( \Delta x_1 \) and \( \Delta x_2 \) in \( x_1 \) and \( x_2 \) are determined by
\[ \frac{\partial f_1}{\partial x_1} \Delta x_1 + \frac{\partial f_1}{\partial x_2} \Delta x_2 = -f_1, \]
\[ \frac{\partial f_2}{\partial x_1} \Delta x_1 + \frac{\partial f_2}{\partial x_2} \Delta x_2 = -f_2. \]

Or, writing the determinant D of the coefficient matrix (the Jacobian),

*Journal of Mathematical Sciences & Mathematics Education Vol. 9 No. 2*
\[ D = \frac{\partial f_1}{\partial x_1} \frac{\partial f_2}{\partial x_2} - \frac{\partial f_1}{\partial x_2} \frac{\partial f_2}{\partial x_1}, \]

then

\[
\Delta x_i \left( \frac{f_1 \frac{\partial f_2}{\partial x_2} - f_2 \frac{\partial f_1}{\partial x_2}}{D} \right) \Delta x_i \left( \frac{f_1 \frac{\partial f_2}{\partial x_1} - f_2 \frac{\partial f_1}{\partial x_1}}{D} \right)
\]

for case in verification, detailed results are tabulated in Table 3.1 and moreover, calculations were carried out using slide rule and the entries 0.000000 showed tiny negative values.

**APPLICATION OF NEWTON-RAPHSON'S METHOD IN SOLVING THE CHEMICAL EQUILIBRIUM PROBLEM**

The principal reactions in the production of synthesis gas by partial oxidation of methane with oxygen are:

\[
\begin{align*}
\text{CH}_4 + \frac{1}{2} \text{O}_2 & \rightarrow \text{CO} + 2\text{H}_2 \\
\text{CH}_4 + \text{H}_2\text{O} & \rightarrow \text{CO} + 3\text{H}_2 \\
\text{H}_2 + \text{CO}_2 & \rightarrow \text{CO} + \text{H}_2\text{O}
\end{align*}
\]

(3.1) (3.2) (3.3)

Write a program that finds the 0 reactant ratio that will produce an adiabatic equilibrium temperature of 2200\(^\circ\)F at an operating pressure of 20 atmospheres, when the reactant gases are preheated to an entering temperature of 1000\(^\circ\)F.

Assuming that the gases behave ideally, so that the component activities are identical with component partial pressures, the equilibrium constants at 2200\(^\circ\)F for the three equations are respectively:

\[
K_1 = \frac{P_{\text{CO}} P_{\text{H}_2}^2}{P_{\text{CH}_4} P_{\text{H}_2\text{O}}} = 1.3 \times 10^{11} \rightarrow \ldots \quad (3.4)
\]

\[
K_1 = \frac{P_{\text{CO}} P_{\text{H}_2}^2}{P_{\text{CH}_4} P_{\text{H}_2\text{O}}} = 1.783 \times 10^5 \rightarrow \ldots \quad (3.5)
\]

\[
K_1 = \frac{P_{\text{CO}} P_{\text{H}_2}^2}{P_{\text{CH}_4} P_{\text{H}_2\text{O}}} = 2.608 \rightarrow \quad (3.6)
\]

Here \(P_{\text{CO}}\), \(P_{\text{CO}_2}\), \(P_{\text{H}_2\text{O}}\), \(P_{\text{CH}_4}\) and \(P_{\text{O}_2}\) are the partial pressures of CO (carbon monoxide), CO\(_2\) (carbon dioxide), H\(_2\)O (water vapor), H\(_2\) (hydrogen), CH\(_4\) (methane), and O\(_2\) (oxygen), respectively. Enthalpies of the various components at 1000\(^\circ\)F and 2200\(^\circ\)F are listed in Table (3.1)
Table (3.1) Component Enthalpies in BTU/b mole

<table>
<thead>
<tr>
<th>Component</th>
<th>1000°F</th>
<th>2200°F</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH₄</td>
<td>-13492</td>
<td>8427</td>
</tr>
<tr>
<td>H₂O</td>
<td>-90546</td>
<td>-78213</td>
</tr>
<tr>
<td>CO₂</td>
<td>-154958</td>
<td>-139009</td>
</tr>
<tr>
<td>CO</td>
<td>-38528</td>
<td>-28837</td>
</tr>
<tr>
<td>H₂</td>
<td>10100</td>
<td>18927</td>
</tr>
<tr>
<td>O₂</td>
<td>10690</td>
<td>20831</td>
</tr>
</tbody>
</table>

A fourth reaction may also occur at high temperatures:
C + CO₂ → 2CO (3.1) at 2200°F, any carbon formed would be deposited as a solid; the equilibrium constant is given by

\[ K_4 = \frac{P^2_{co}}{a_c P_{CO}} = 1.7837 \times 10^{-5} \]  

where \( a_c \) is the activity of carbon in the solid state. Do not include reaction (3.7) in the equilibrium analysis. After establishing the equilibrium composition, considering only the homogeneous gaseous reactions given by (3.1), (3.2), and (3.3), determine the thermodynamic likelihood that solid carbon would appear as a result of reaction (3.7). Assume that the activity of solid carbon is unaffected by pressure and equals unity.

Use the Newton-Raphson method to solve the system of simultaneous nonlinear equations developed as the result of the equilibrium analysis.

### 3.2 METHOD OF SOLUTION

Because of the magnitude of \( K \), the equilibrium constant for reactions, the first reaction can be assumed to go to completion at 2200°F, that is virtually no unrelated oxygen will remain in the product gases at equilibrium.

Let the following nomenclature be used.

- \( x_1 \) mole fraction of CO in the equilibrium mixture
- \( x_2 \) mole fraction of CO₂ in the equilibrium mixture
- \( x_3 \) mole fraction of H₂O in the equilibrium mixture
- \( x_4 \) mole fraction of H₂ in the equilibrium mixture
- \( x_5 \) mole fraction of CH₄ in the equilibrium mixture
- \( x_6 \) number of moles of O₂ per mole of CH₄ in the feed gas
- \( x_7 \) number of moles of product gases in the equilibrium mixture per mole of CH₄ in the feed gases.

Then a system of seven simultaneous equations may be generated from three atom balances an energy balance, a mole fraction constraint and two equilibrium relations.

Atom conservation balances: the number of atoms of each element entering equals the number of atoms of each element in the equilibrium mixture.

- Oxygen: \( x_6 = (1/2x_1 + x_2 + 1/2x_3) \rightarrow \ldots \)  (3.9)
- Hydrogen: \( 4 = (2x_3 + 2x_4 + 4x_2) \rightarrow \ldots \)  (3.10)
- Carbon: \( 1 = (x_1 + x_2 + x_5) \rightarrow \ldots \)  (3.11)
Since the reaction is to be conducted adiabatically, that is, no energy is added to or removed from the reacting gases, the enthalpy (H) of the reactants must equal the enthalpy of the products.

\[ [\text{HCH}_4 + x_0\text{H}_2]_1000^\circ F = x_1[\text{CH}_2\text{O}_2] + x_3\text{H}_2\text{O} + x_4\text{H}_2 + x_5\text{HCH}_2]_2200^\circ F \quad \text{... (3.12)} \]

mole fraction constraint.

\[ x_1 + x_2 + x_4 + x_5 = 1 \quad \text{... (3.13)} \]

Equilibrium relations.

\[ K_i = \frac{P_i x_i X_i}{x_{co} X_{co}} = 1.7837 \times 10^{-1} \rightarrow \quad \text{... (3.14)} \]

\[ K_i = \frac{P_i x_i X_i}{x_{co} X_{co}} = 2.6058 \rightarrow \quad \text{... (3.15)} \]

The relationships (3.14) and (3.15) follow directly from (3.5) and (3.6), respectively, where \( P \) is the total pressure and \( P_{co} = P x_i \), etc. in addition, there are five side conditions.

\[ x_i \geq 0, \quad I = 1, 2, 5 \quad \text{... (3.16)} \]

These C ions more that all mole fractions in the equilibrium mixture are nonnegative, that is, any solution of equation (3.9) to 3.15) that contains negative mole fractions is physically meaningless from physical-chemical principle there is one and only one solution of the equation that satisfies conditions (3.16). Any irrelevant solutions may be detected easily.

The seven equations may be rewritten in the form.

\[ f_1(x) = 0, \quad I = 1, 2, ..., 7 \quad \text{(5.5.17)} \]

where \( x = [x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T \), as follows:

\[ f_1(x) = \frac{1}{2} x_1 + x_2 + \frac{1}{2} x_3 - \frac{x_4}{x_7} = 0 \rightarrow \quad \text{(3.19 a)} \]

\[ f_2(x) = x_1 + x_2 + 2 x_5 - \frac{2}{x_7} = 0 \rightarrow \quad \text{(3.19 b)} \]

\[ f_3(x) = x_1 + x_2 + x_3 - \frac{1}{x_7} = 0 \rightarrow \quad \text{(3.19 c)} \]

\[ f_4(x) = -28837 x_1 - 139009 x_2 - 78213 x_3 + 18927 x_4 + 8427 x_5 + 13492 x_6 - 10690 x_7 = 0 \rightarrow \quad \text{(3.19 d)} \]

\[ f_5(x) = x_1 x_2 x_4 x_5 x_7 - 1 = 0 \rightarrow \quad \text{(3.19 e)} \]

\[ f_6(x) = P_1^2 x_1 x_4 - 1.7837 \times 10^{-1} x_1 x_5 = 0 \rightarrow \quad \text{(3.19 f)} \]

\[ f_7(x) = x_1 x_3 - 2.6058 x_2 x_4 = 0 \rightarrow \quad \text{(3.19 g)} \]

The system of simultaneous nonlinear equations has the form (2.1), and will be solved using the Newton-Raphson method, described in section 2.2. The partial derivatives of above may be found by partial differentiation of the seven functions, \( f_1(x) \), with respect to each of the seven variables. For example,
The Newton-Raphson method may be summarized as follows:

Choose a starting vector \( \mathbf{x}_0 = [x_{10}, x_{20}, \ldots, x_{70}] \), where \( x_0 \) is hopefully near a solution.

1. Solve the system of linear equations (2.14),
\[
\phi_i(x_k) \delta_k = -f(x_k),
\]
where
\[
\phi_{ij}(x_k) = \frac{\partial f_i}{\partial x_j}(x_k), \quad i=1,2,\ldots,7, \quad j=1,2,\ldots,7,
\]
for the increment vector
\[
\delta_k = \begin{bmatrix} \delta_{1k} & \delta_{2k} & \ldots & \delta_{7k} \end{bmatrix}^T
\]
update the approximation to the root for the next iteration. \( x_{k+1} = x_k + \delta_k \).

2. Check for possible convergence to a root \( \alpha \). One such test might be
\[
|\delta_{ik}| < \varepsilon_2, \quad i=1,2,\ldots,7.
\]
if (3.23) is true for all \( i \), then \( x_k + \delta_k \) is taken to be the root. If test (3.23) is failed for any \( i \), then the process is repeated starting with step 2. The iterative process is continued until test (3.23) is passed for some \( k \), or when \( k \) exceeds some specified upper limit. In the programs that follow, the elements of the augments matrix \( A = [\phi(x_k) : -f(x_k)] \) (3.24) are evaluated by a subroutine named CALCN. The system of linear equations (3.24) is solved by calling on the function SIMUL, described in detail in example (2.1). The main program is a general one, in that it is not specifically written to solve only the seven equations of interest. By properly defining the subroutine CALCN, the main program could be used to solve any system of \( n \) simultaneous nonlinear equations. The main program reads data values for \( \text{itmax} \), \( \text{iprint} \), \( n \), \( \Sigma_1 \), \( \Sigma_2 \), and \( x_1, x_2, \ldots, x_n \), here, \( \text{itmax} \) is the maximum number of Newton-Raphson’s iterations, \( \text{iprint} \) is a variable that controls printing of intermediate output, \( n \) is the number of nonlinear equations, \( \Sigma_1 \) is the minimum pivot magnitude allowed in the Gauss-Jordan reduction algorithm, \( \Sigma_2 \) is a small positive number used in test (3.23), and \( x_1, x_2, \ldots, x_n \) that is, the elements of \( x_0 \).
Journal of Mathematical Sciences & Mathematics Education Vol. 9 No. 2

FLOW DIAGRAM

Main Program

- Begin
- Itmax, iprint
- n, Σ1, Σ2
- x10, ..., xn0

- Evaluate elements
  a_k of augmented
  matrix A (see (5.5.24)).

- “Matrix ill-
  conditioned”
- Solve system of n
  linear equations
  (5.44) for the
  increments
  δ_1, δ_2, ..., δ_n
  and determinant, d.
  (function SIMUL)

- “Convergent”
- Calculate elements
  a_k, i = 1, 2, ..., n
  j = 1, 2, ..., n+1 of
  matrix A

- Itcon =
  9
  “No
  Convergence”

- “Convergent”
- End

Subroutine CALC (Arguments: x_k, A, N)

- Calculate elements
  a_k, i = 1, 2, ..., n
  j = 1, 2, ..., n+1 of
  matrix A

- Retur

- End

End
**Fortran Implementation**

List of Principle various program Symbol

(Main)

- \( A \) - Augmented matrix of coefficients, \( A \) (see (3.22).
- DETER - \( d \), determinant of the matrix \( \phi \) (the jacobian).
- EPS1 - \( \Sigma_1 \), minimum pivot magnitude permitted in subroutine SIMUL.
- EPS2 - \( \Sigma_2 \), small positive number, used in convergence test (3.23).
- IPRINT - Print control variable, if iprint = 1, intermediate solutions are printed after each iteration.
- ITCON - used in convergence test (3.23). ITCON 1 if (3.23) is passed for all \( i \), \( i = 1, 2, \ldots, n \); otherwise ITCON = 0.
- ITER - Iteration counter, \( k \).
- ITMAX - maximum number of iterations permitted, itmax.
- N - number of nonlinear equations, \( n \).
- XINC - vector of increments, \( x_{ik} \), \( i = 1, 2, \ldots, n \).
- XOLD - vector of approximations to the solution, \( x_{ik} \).
- SIMUL - function developed in Example (2.1) solves the system of \( n \) linear equations (2.15) for the increments, \( x_{ik} \).

(subroutine CALCN)

- DXOLD - same as XOLD. Used to avoid an excessive number of reference to subroutine arguments in CALCN.
- I, J - i and j, row and column subscript, respectively.
- NRC - \( N \), dimension of the matrix \( A \) in the calling program. \( A \) is assumed to have the same number of rows and columns.
- P - pressure, \( P \), atm.

**Program Listing (Chemical Equilibrium by Newton-Raphson Method)**

**Main Program**

```c
APPLIED NUMERICAL METHODS, EXAMPLES 5.5
C CHEMICAL EQUILIBRIUM- NEWTON-RAPSON METHOD
C
C THIS PROGRAM SOLVES N SIMULTANEOUS NON LINEAR EQUATIONS
C IN N UNKNOWNS BY THE NEWTON-RAPSON ITERATIVE PROCEDURE
C INITIAL GUESSES FOR VALUES OF THE UNKNOWNS ARE READ INTO
C XOLD (1)……XOLD (N). THE PROGRAM FIRST CALLS ON THE SUBROUTINE
C CALCN TO COMPUTE THE ELEMENTS OF A. THE AUGMENTED MATRIX OF
```
C PARTIAL DERIVATIVES, THEN ON FUNCTION SIMUL (SEE
PROBLEM 5.2)
C TO SOLVE THE GENERATED SET OF LINEAR EQUATIONS
FOR THE CHANGES
C IN THE SOLUTION VALUES XINC(1)…..XINC(N). DETER
IS THE
C JOCABIAN COMPUTED BY SIMUL. THE SOLUTIONS ARE
UPDATED AND THE
C PROCESS CONTINUED UNTIL ITER, THE NUMBER OF
ITERATIONS,
C EXCEEDS ITMAX OR UNTIL THE CHANGE IN EACH OF
THE N VARIABLES
C IS SMALLER IN MAGNITUDE THAN EPS2 (ITCON =1
UNDER THESE
C CONDITIONS). EPS1 IS THE MINIMUM PIVOT MAGNITUDE
PERMITTED
C IN SIMUL. WHEN IPRINT =1, INTERMEDIATE SOLUTION
VALUES ARE
C PRINTED AFTER EACH ITERATION.
C
DIME imension XOLD (21), XINC (21), A (21,21)
C
C ….READ AND PRINT DATA...
1 READ (5,100) ITMAX, IPRINT, EPS1, EPS2, (XOLD(1), 1=1, N)
WRITE(6,200) ITMAX, IPRINT, N, EPS1, EPS2, N, (XOLD)(1),
I=1,N)
C
C …. CALL ON CALCN TO SET UP THE A MATRIX...
CALL CALCN (XOLD, A, 21)
C
C …. CALL SIMUL TO COMPUTE JACOBIAN AND CORRECTIONS
IN XINC...
DETER = SIMUL (N, A, XINC, EPS1, 1, 21)
IF ( DETER .NE. D. ) GO TO 3
WRITE (6 GO TO
C
C ….CHECK FOR CONVERGENCE AND UPDATE XOLD VALUES...
3 ITCON = 1
DO 5 1 – 1, N
IF (ABS(XINC(1)) .GT. EPS 2 ) ITCON = 0
5 XOLD (1) = XOLD(1) + XINC(1)
IF ( IPRINT. EQ. 1 ) WRITE (6, 202) ITER, DETER, N, (XOLD (1), 1 =
1,N)
IF ( ITCON. EQ. 0 ) WRITE (6,203) ITER, N, (XOLD(1), 1 = 1, N)
GO TO 1
9 CONTINUE
C WRITE 96, 204)
PROGRAM Listing (continued)
Subroutine CALCN

    SUBROUTINE CALCN (XOLD, A, NRC)
    C               THIS SUBROUTINE SETS UP THE AUGMENTED MATRIX OF
    C               PARTIAL
    C               DERIVATIVES REQUIRED FOR THE SOLUTION OF THE
    C               NON-LINEAR
    C               EQUATIONS WHICH DESCRIBE THE EQUILIBRIUM
    C               CONCENTRATIONS
    C               OF CHEMICAL CONSTITUENTS RESULTS FROM PARTIAL
    C               OXIDATION
    C               OF METHANE WITH OXYGEN TO PRODUCE SYNTHESIS
    C               GAS. THE PRESSURE
    C               IS 20 ATMOSPHERES. SEE TEXT FOR MENINGS OF
    C               XOLD(1)…XOLD(N)
    C               AND A LISTING OF THE EQUATIONS. DXOLD HAS BEEN
    C               USED AS THE
    C               DUMMY ARGUMENT FOR XOLD TO AVOID AN
    C               EXCESSIVE NUMBER OF
    C               REFERENCES TO ELEMENTS IN THE ARGUMENT LIST.
    C               DIMENSION XOLD(20), DXOLD(NRC), A(NRC , NRC)
    C               DATA P / 20.
    C               …SHIFT ELEMENTS OF DXOLD TO XOLD AND CLEAR A
    C               ARRAY…
      DO 1 I = 1, 7
      XOLD(1) = DXOLD(1)
      DO 1 J = 1, 8
        A(I, J) = 0.
      1
    C               …COMPUTE NON-ZERO ELEMENTS OF A…
      A (1,1) = 0.5
      A (1,2) = 1.0
      A (1,3) = 0.9
      A (1,6) = 1.0 / XOLD (7)
      A (1,7) = XOLD (6) / XOLD (7)**2
      A (1,8) = - XOLD(1) / 2. - XOLD(2) - XOLD(3) /2, * XOLD (6) / XOLD(7)
      A (2,3) = 1.0
      A (2,4) = 1.0
      A (2,5) = 2.0
      A (2,7) = 2.0 / XOLD (7)**2
      A (2,8) = XOLD(1) - XOLD(2) - XOLD(5) + 1.0 / XOLD(7)
Program listing (continued)

Data

\[
\begin{align*}
\text{ITMAX} & = 50 & \text{IPRINT} & = 1 & \text{N} & = 7 \\
\text{EPS1} & = 1.0 \times 10^{-10} & \text{EPS2} & = 1.0 \times 10^{-05} \\
\text{XOLD}(1) & \ldots \text{XOLD}(5) & = 0.500 & 0.000 & 0.000 \\
\text{XOLD}(6) & \ldots \text{XOLD}(7) & = 0.500 & 2.000 \\
\text{ITMAX} & = 50 & \text{IPRINT} & = 0 & \text{N} & = 7 \\
\text{EPS1} & = 1.0 \times 10^{-10} & \text{EPS2} & = 1.0 \times 10^{-05} \\
\text{XOLD}(1) & \ldots \text{XOLD}(5) & = 0.200 & 0.200 & 0.200 \\
\text{XOLD}(6) & \ldots \text{XOLD}(7) & = 0.500 & 2.000 \\
\text{ITMAX} & = 50 & \text{IPRINT} & = 0 & \text{N} & = 7
\end{align*}
\]

Chemical Equilibrium (Newton-Raphson Method)
Computer Output
Results for the 1st Data Set

\begin{align*}
\text{EPS1} &= 1.0 \times 10^{-10} \quad \text{EPS2} = 1.0 \times 10^{-5} \\
\text{XOLD(1)} \ldots \text{XOLD(5)} &= 0.220 \quad 0.075 \\
\text{XOLD(6)} \ldots \text{XOLD(7)} &= 0.436 \quad 2.350
\end{align*}

\begin{tabular}{c c c c c c c c c}
& & & & & & & & & \\
\hline
\multicolumn{7}{c}{Computer Output} \\
\multicolumn{7}{c}{Results for the 1st Data Set} \\
\hline
\text{ITMAX} & = & 50 & & & & & & & \\
\text{IPRINT} & = & 1 & & & & & & & \\
\text{N} & = & 7 & & & & & & & \\
\text{EPS1} & = & 1.0 \times 10^{-10} & & & & & & & \\
\text{EPS2} & = & 1.0 \times 10^{-5} & & & & & & & \\
\text{XOLD(1)} \ldots \text{XOLD(7)} & & & & & & & & & \\
5.000000E -01 & 0.0 & 0.0 & & & & & & & \\
4.263276E -01 & 2.591652E -01 & 3.3432350E -01 & 1.975559E 00 & & & & & & \\
5.791981E-01 & 4.812878E -02 & 4.681466E -01 & 2.524948E 00 & & & & & & \\
6.129664E -01 & 1.048106E-02 & 5.533223E -01 & 2.880228E 00 & & & & & & \\
6.180951E-01 & 3.811378E -03 & 5.758237E -01 & 2.974139E 00 & & & & & & \\
6.181716E-01 & & & & & & & & & \\
\hline
\end{tabular}
3.716873E -03                        5.767141E -01                   2.977859E  00
ITER                  =                     6
DETER              =                    -0.20134E       09
XOLD(1)…XOLD(    7)
3.228708E -01                        9.223547E -03                   4.601710E -02
6.181716E -01                        5.767153E -01                   2.977863E  00

Computer Output
SUCCESSFUL CONVERGENCE
ITER                 =                 6
XOLD(1)…XOLD(    7)
3.228708E -01                        9.223547E -03                   4.601710E -02
6.181716E -01                        5.767153E -01                   2.977863E  00

Results for the 3rd Data Set
ITMAX                =                 50
IPRINT               =                   1
N                        =                   7
EPS1                  =                   1. 0E -10
EPS2                  =                   1.  0E -05
XOLD( 1)...XOLD (      7)
2.200000E -01                              7.499999e -02                         9.999999e -04                         5.800000E-01                              1.250000e -01                              4.360000e -01                  2.349999e
00
ITER                   =                1
DETER               =               -0.61808E  08
XOLD(1)…XOLD(    7)
6.9514955E -01                           -8.022028E -02                       1.272939E -02                        1.217132E  00
-8.447912E -01                                1.314754E  00                  5.969404E
00
ITER                    =                2
DETER                =                0.12576E   09
XOLD(1)…XOLD(    7)
4.958702E -01                                -1.698154E -02                       5.952045E
-3.65007E -01                               2.379797E  00
1.043425E  01
ITER                   =                3
DETER               =                0.77199E  07
XOLD(1)…XOLD(    7)
Tables (3.2) Equilibrium Gas mixture

| x_1 | Mole fraction CO   | 0.322871 |
| x_2 | Mole fraction CO_2 | 0.009224 |
| x_3 | Mole fraction H_2O | 0.046017 |
| x_4 | Mole fraction H_2  | 0.618172 |
| x_5 | Mole fraction CH_4 | 0.003717 |
| x_6 | Mole fraction O_2 / CH_4 | 0.576715 |
| x_7 | Total moles of product | 2.977863 |

In the feed gases, and total number of moles of product per mole of HC_4 in the feed are tabulated in Table (3.2). Thus the required feed ratio is 0.5767 moles of oxygen per moles of methane in the feed gases.

To establish if carbon is likely to be formed according to reaction (5.5.7) at 2200°F for a gas of the computed composition, it is necessary to calculate the magnitude of

\[
\bar{K} = \frac{P_{CO}^2}{a_{CO} P_{CO_2}} = \frac{P_{x_2}^2}{a_{x_2}} \quad \ldots
\]  

(3.25)

If \( \bar{K} \) is larger than \( k_4 \) from (3.25), then there will be a tendency for reaction (3.24) to shift toward the left; carbon will be formed. Assuming that \( a_c = 1 \),

\[
\bar{K} = \frac{P_{CO}^2}{a_{CO} P_{CO_2}} = \frac{P_{x_2}^2}{a_{x_2}} \quad \ldots
\]  

(3.25)
\[
\frac{-20t(0.32287)}{10.009224} = 22603 < k_4 = 13295.
\]

Therefore there will be no tendency for carbon to form.

† Okereke, C. Emmanuel, Ph.D., Department of Mathematics, Michael Okpara University of Agriculture, Nigeria

REFERENCES