Matrix Representations of Nonlinear Equation Iterations—Application to Parallel Computation

By John R. Rice

Abstract. A matrix representation of iterative methods is presented which includes almost all those based on polynomial methods. A simple lemma and corollaries are established which show that the order of convergence of the iteration is the spectral radius of the matrix representation. A number of old and new methods, particularly those adapted to parallel computation, are analyzed using this representation.

1. Introduction. We consider solving the single equation

\[ f(x) = 0. \]

The purpose of this note is threefold. First, we exhibit a matrix representation of iterative methods which includes almost all of those based on polynomial approximation or Taylor series expansion. This representation can be written down by inspection in many cases, including the "standard" iteration methods. We next present a simple lemma and corollaries which show that the order of convergence of the iteration is the spectral radius of the matrix representation. Finally, we consider iteration methods particularly suited for parallel computation and single out two particularly efficient ones.

Recall that a computer with parallel processing can perform many computations simultaneously. In particular, \( f(x) \) can be evaluated for a number of arguments in the same time that it takes to evaluate it for one argument. Note that parallel computation sometimes can be used to speed even one evaluation of \( f(x) \), e.g.

\[ f(x) = x^2 + \cos(3x + 1) \cdot \sin(3x + 2) + \sin(6x - 5) \cdot \cos(3x - 1) \]

can be evaluated at almost the same speed as \( g(x) = 4 \cos(3x + 2) \). We assume that an adequate number of processors is available for any parallel computation specified in this paper.

2. Matrix Representations. We consider iterative methods to compute a sequence of \( n \)-tuples \( \{x_1^k, k = 1, 2, \ldots, n\} \), of values each of which is an estimate of the root \( x^* \) of (1). We denote the \( n \)-tuple by \( X^i = (x_1^i, \ldots, x_n^i) \), the \( x_k^i \) need not be distinct. In order to exploit parallel computation, one considers iterations of the form

\[ x_{k+1}^i = \varphi_k(X^i), \quad k = 1, 2, \ldots, n, \]

where \( \varphi_k \) is some iteration function. These computations usually can be performed simultaneously, and the speed of evaluation of \( X^{i+1} \) from \( X^i \) is independent of \( n \).
We consider polynomial interpolation methods defined as follows:

1. For each \( k \), a subset of the \( n \)-tuple \( X' \) is selected. We indicate this selection by the row vector \( T_k = (t_{k1}, \ldots, t_{kn}) \) which has \( t_{ki} = 1 \) if \( x'_i \) is selected and has \( t_{ki} = 0 \) otherwise.

2. A polynomial \( P_k(y) \) in \( y \) is obtained which interpolates the points \( (x'_i, y_i = f(x'_i)) \) selected. (This is classical inverse interpolation.)

3. The value \( x_{i+1} \) is \( P_k(0) \).

These methods are then represented by the matrix \( T \) with rows \( T_k \). Two simple cases are illustrated below:

**Secant Method.** We have a sequence of pairs \( x'_i, x'_2 \), and \( x'_{i+1} \) is determined by linear interpolation through \( x'_1 \) and \( x'_2 \). We simply take \( x'_{i+1} \) to be \( x'_i \). The matrix representation is

\[
T_s = \begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix}
\]

and one might symbolically visualize that

\[
(x'_1, x'_2+1) = T_s \otimes \begin{bmatrix} x'_1 \\ x'_2 \end{bmatrix},
\]

but the required “operator” \( \otimes \) is undefined. Recall that the error \( e''_i = |x^* - x'_i| \) of the secant method satisfies \( e''_i = K e'_i e'_2 \) and observe that \( e''_i = e'_i \). Note that the spectral radius of \( T_s \) is 1.618 which is the order of convergence of the secant method.

**Newton's Method.** We may interpret Newton's method as the limiting case of chord interpolation as the two points of interpolation coalesce. Thus we have a sequence of pairs of identical points \( x'_1 = x'_2 \). The point \( x'_{i+1} \) is determined by the interpolation of value and derivative (which is the limiting case for linear interpolation at coalescent points) and since \( x'_2+1 = x'_1+1 \), the matrix representation has two identical rows:

\[
T_N = \begin{bmatrix}
1 & 1 \\
1 & 1
\end{bmatrix}.
\]

Recall that the errors \( e''_i = |x^* - x'_i| \) satisfy

\[
e''_i = K (e'_i)^2 = K e'_i e'_2, \quad e''_i = e'_{i+1} = K (e'_i e'_2),
\]

and observe that the spectral radius of \( T_N \) is 2, which is the order of convergence of Newton's method.

We note later some composite methods and such a method is represented by a set of matrices, one for each step in the composition. Thus, if one alternately used Newton's method and the secant method, the matrix representation is \( T_s, T_N \). Observe that, with matrix products,

\[
T_s \ast T_N = \begin{bmatrix}
2 & 2 \\
1 & 1
\end{bmatrix}
\]

which has spectral radius 3. It follows from Corollary 3 that this is the order of convergence of this composite method.
3. Order of Convergence. A discussion of parallel computation leads one to
distinguish between the time that a computation takes and the effort that it takes.
Thus we define two kinds of order for an iteration as follows: Set
\[ e'_k = -\log |x'_k - x^*|, \quad k = 1, 2, \cdots, n; \quad i = 1, 2, \cdots, \]
\[ E' = (e'_1, e'_2, \cdots, e'_n), \]
and the temporal order is
\[
(3) \quad \rho = \lim_{i \to \infty} \frac{|E'^{i+1}|}{|E'^i|},
\]
where \( \cdot \) is a convenient vector norm. The exact choice does not affect the later
analysis so we do not make one here. Let \( p \) be the total number of evaluations of
\( f(x) \) and similar quantities (e.g. \( f'(x) \)) made in (2). The computational order is \( \omega = p^{1/p} \).

The definition (3) is standard for the order, but it leads to certain technical com-
plications since the limit need not exist in all cases. Thus, we alternatively define
the temporal order by
\[
(3a) \quad \rho = \lim_{i \to \infty} \left( \frac{|E'|}{|E^i|} \right)^{1/i}.
\]
If the limit in (3) exists, then (3a) gives the same value. This paper shows that the
limit in (3a) does exist and shows how to compute the order \( \rho \).

Note that we say a method converges if \( |E'| \) tends to infinity. This does not
imply that \( x'_k - x^* \) converges to zero for all \( k \). We have chosen to select the "best"
component \( x'_k \) as the one which measures the convergence and its order.

We first establish a lemma which relates the order of convergence to zero of a
sequence of recursively defined quantities to a matrix \( T \) used in the recursion. This
lemma has application to other areas of numerical analysis besides the specific one
in this paper. For simplicity of notation, we introduce the index set
\[ I_k = \{ j \mid t_{jk} = 1 \}, \quad k = 1, 2, \cdots, n, \]
and consider the following recursion
\[
(4) \quad v_k^{i+1} = a \prod_{j \in I_k} (1 + \delta(v'_j)) v'_j, \quad k = 1, 2, \cdots, n,
\]
which defines the sequence of vectors \( V' \).

**Lemma.** Suppose there are constants \( \delta^* \) and a number \( M \) so that
\[ |\delta(v'_j)| \leq \delta^* |v'_j| \quad \text{for} \quad |v'_j| \leq M. \]

Let \( \rho_T \) be the spectral radius (largest eigenvalue) of \( T \). Further suppose that either
\( |a| < 1 \) or there is a \( k^* \), \( 1 \leq k^* \leq n \), so that \( I_k \) has at least two nonzero components.
Then there is an \( \epsilon > 0 \) so that \( |V^0| < \epsilon \) implies that there is convergence and the order
\( \rho \) of the convergence satisfies
\[
(5) \quad \rho = \rho_T.
\]

**Proof.** It follows from the second supposition that there is an \( \epsilon_1 > 0 \) so that
\( |V^0| < \epsilon_1 \) implies that some \( v'_k \) converges to zero. We assume that \( \epsilon < M \), \( \epsilon < \epsilon_1 \)
and define
\[ e'_k = -\log |v'_k|, \quad E' = (e'_1, \cdots, e'_n). \]
We have from (4) that
\[ e_{k}^{i+1} = \log a + \sum_{j \in I_{k}} e_{j}^{i} + \sum_{j \in I_{k}} \log (1 + \delta(v_{j}^{i})) \]
or
\[ \frac{E_{i}^{i+1}}{||E_{i}^{i}||} = \frac{A}{||E_{i}^{i}||} + \frac{TE_{i}^{i}}{||E_{i}^{i}||} + \frac{TW}{||E_{i}^{i}||}, \]
where
\[ W = (\log(1 + \delta(v_{i}^{j})))_{n-1}, \quad A = (\log a, \cdots, \log a). \]
The first assumption of the lemma implies that
\[ ||TW||/||E_{i}^{i}|| \leq \max \delta \]
and hence \( ||TW||/||E_{i}^{i}|| \) converges to zero since \( ||E_{i}^{i}|| \) converges to \( +\infty \). Similarly, we have that \( ||A||/||E_{i}^{i}|| \) converges to zero. We have now established that \( ||E_{i}^{i+1}||/||E_{i}^{i}|| \) behaves like \( ||TE_{i}^{i}||/||E_{i}^{i}|| \) as \( i \) tends to infinity.

The vector \( W \) depends on \( i \) and we denote \( (||A|| + ||TW||)/||E_{i}^{i}|| \) by \( \epsilon_{i} \) and we know that \( \epsilon_{i} \) tends to zero. Even though \( ||E_{i}^{i}|| \) tends to infinity, we might have \( ||TE_{i}^{i}|| = 0 \) if \( E' \) does not contain any component of the subspace associated with the eigenvalue \( \rho_{T} \). For the moment, we assume that \( E' \) has such a component for all \( i \) sufficiently large. Then we have
\[ ||E_{i}^{i}|| \geq ||TE_{i}^{i-1}|| (1 - \epsilon_{i-1}/\rho_{T}) \geq ||T||^{i-1} ||TE_{i}^{i}|| (1 - \epsilon_{i}/\rho_{T})^{i-1} \]
\[ \geq (\rho_{T})^{i-1} ||TE_{i}^{i}|| (1 - \epsilon_{i}/\rho_{T})^{i-1}. \]
Fix \( j \) and we have
\[ \lim_{i \to \infty} ||E_{i}^{i}||^{1/i} = \lim_{i \to \infty} \left[ (\rho_{T})^{i-1/j} ||TE_{i}^{i}||^{1/i} (1 - \epsilon_{i}/\rho_{T})^{1-(i+1)/j} \right] \]
\[ = \rho_{T}(1 - \epsilon_{i}/\rho_{T}). \]
Now \( j \) is arbitrary and since \( \epsilon_{i} \to 0 \) as \( j \to \infty \), we have
\[ \lim_{i \to \infty} ||E_{i}^{i}||^{1/i} \geq \rho_{T}. \]
We likewise note that
\[ ||E_{i}^{i}|| \leq ||TE_{i}^{i-1}|| (1 + \epsilon_{i-1}/\rho_{T}) \]
and, since \( E' \) contains components in the subspace associated with \( \rho_{T} \), we have
\[ ||E_{i}^{i}|| \leq \rho_{T} ||E_{i}^{i-1}|| (1 + \epsilon_{i-1}/\rho_{T}). \]
Thus
\[ ||E_{i}^{i}|| \leq \rho_{T}^{i-1} ||E_{i}^{i}|| (1 + \epsilon_{i}/\rho_{T})^{i-1} \]
and a similar argument shows that
\[ \lim_{i \to \infty} ||E_{i}^{i}||^{1/i} \leq \rho_{T}. \]
This concludes the proof except for the following point.

It is conceivable that there is an $E^0$ so that for all $i$ sufficiently large $E^i$ is in a subspace complementary to the subspace associated with $\rho_T$. Round-off effects will introduce components associated with the eigenvalue $\rho_T$ in any actual computation. However, the phrase "with probability one" has been included in the lemma’s conclusion to express this observation.

In order to apply this lemma to iterative methods, we need to derive a recursion relation for the errors $e_k^i$. This can be done neatly and in generality using Newton’s interpolation formula and divided differences. To simplify the notation we set

$$g(y) = f^{-1}(y), \quad g^{(p)}(y) = d^p g(y)/dy^p, \quad y_k = f(x_k).$$

The points entering into the determinations of $x_{k+1}^i$ are determined by $I_k$ and we temporarily fix $k$ and denote the elements of $I_k$ by $j_1, j_2, \ldots, j_p$. We have

$$g(y) = g(y_{j_1}) + g[y_{j_1}, y_{j_2}](y - y_{j_1}) + \cdots + g[y_{j_1}, \ldots, y_{j_p}]$$

$$\cdot (y - y_{j_1}) \cdots (y - y_{j_{p-1}}) + g[y_{j_1}, \ldots, y_{j_p}, y] (y - y_{j_1}) \cdots (y - y_{j_p}).$$

The final term in (6) is the error of interpolation made in using the other terms which constitute $P_k(y)$. With $x_{k+1}^i = P_k(0)$, we have

$$|x_{k+1}^i - x^*| = |g[y_{j_1}, \ldots, y_{j_p}, 0] y_{j_1} \cdots y_{j_p}|.$$ 

Note that the superscripts $i$ have been omitted on the $y$’s. This equation is somewhat more familiar if we note that

$$y_{i_m} = f'(\xi_{i_m})(x^* - x_{i_m}^i)$$

and replace the divided difference by a mean value of the $p$th derivative at a point $\eta_k$. We obtain

$$|x_{k+1}^i - x^*| = |g^{(p)}(\eta_k)| \prod_{i \in I_k} |f'(\xi_i)| |x^* - x_i^i|.$$ 

This may be rewritten in a form so as to apply the lemma as follows:

$$|x_{k+1}^i - x^*| = a \prod_{i \in I_k} (1 + \delta(|x^* - x_i^i|)) |x^* - x_i^i|$$

where

$$a = \frac{|g^{(p)}(0)| |f'(x^*)|^p}{1 + \delta(|x^* - x_i^i|)} \cdot \frac{1 + g^{(p+1)}(\tilde{\eta}_k)f'(\tilde{\eta}_k)(x^* - \tilde{\eta}_k)/g^{(p)}(0)}{1 + f''(\xi_i^i)(x^* - x_i^i)/f'(x^*)}.$$ 

We note that all the mean values $\eta_k$, $\tilde{\eta}_k$ and $\xi_i^i$ lie in the interval spanned by the $x_i^i$ and $x^*$. We now have immediately from the lemma the

**Corollary 1.** Assume that $f'(x^*) \neq 0$ and $f^{(j)}(x^*)$ exists and is continuous for $j = 2, \ldots, n + 1$ and that the spectral radius of the matrix representation $T$ of the iteration is greater than 1. Then there is an $\epsilon > 0$ so that $|x_k^i - x_k^0| < \epsilon, k = 1, 2, \ldots, n$, implies that the iteration defined by $T$ converges with order $\rho$ which is the spectral radius $\rho_T$ of $T$.

It is perhaps not obvious that this corollary does not imply that all components of $X^i$ converge to $x^*$ with order $\rho_T$, but rather only that some do. Useful methods are likely to have the same order of convergence for all components of $X^i$. 
We note that a slight modification of this argument can be carried through when inverse interpolation is replaced by direct interpolation. We have

**Corollary 2.** Assume that the polynomial interpolation method uses direct rather than inverse interpolation. Then Corollary 1 is still valid.

These two corollaries include Theorem 1 of [1]. The present shorter and simpler proof is due to the fact that the difficulties inherent in Hermite interpolation are taken care of by the use of standard results on divided differences.

We also note

**Corollary 3.** The order $\rho$ of convergence of a composite method $T_kT_{k-1} \cdots T_2T_1$ is the spectral radius $\rho_T$ of the product of the matrices, i.e. of

$$T = T_kT_{k-1} \cdots T_2T_1.$$ 

Thus, as noted earlier, the composition of Newton's and the secant method has order of convergence equal to the spectral radius of $T_ST_N$, which is $3.**$

### 4. Analysis of Old and New Iterative Methods

This section considers a number of iterative methods which are either of practical interest or of an illustrative nature.

- **a. Standard n-Point Newton and Secant Methods.** Both of these are included and the matrix representations are given for the case of 4-point methods (using cubic polynomials).

  \[
  T = \begin{bmatrix}
  1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1 \\
  1 & 1 & 1 & 1
  \end{bmatrix}, \quad T = \begin{bmatrix}
  1 & 1 & 1 & 1 \\
  1 & 0 & 0 & 0 \\
  0 & 1 & 0 & 0 \\
  0 & 0 & 1 & 0
  \end{bmatrix},
  \]

  $\rho = 4$, \quad $\rho = \text{largest zero of } t^4 - t^3 - t^2 - t - 1 = 1.976$, \quad $\omega = 4^{1/4} = 2^{1/2} = 1.414$, \quad $\omega = \rho$.

  Note that $T$ for the secant method is simply the companion matrix for the polynomial $t^4 - t^3 - t^2 - t - 1$.

- **b. Recent Parallel Computation Methods.** The first attempt to use parallel computation for solving nonlinear equations appears to be [3]. The approach there does not make effective use of parallel computation in that the order of convergence is not essentially increased, but the computational effort is. The situation is illustrated for the second method described in [3]. We choose $N = 5$ and the scheme is as follows (given an initial pair of points $a_0, b_0$ so that $f(a_0) > 0, f(b_0) < 0$).

  1. Choose 5 points $x_i$ equally spaced in the interval $a_k, b_k$.
  2. Pair $x_i$ with $a_k$ or $b_k$ so that the signs of the values of $f(x)$ are opposite.
  3. Compute $y_i$ by the secant method from this pair of points.
  4. Select the pair (out of the 12 points now generated) of points $a_{k+1}, b_{k+1}$ so that $f(x)$ changes sign between the pair and $f(x)$ is minimized on the pair.

  ** Theorem 2 of [1] implies that the order of convergence is $3.24 = (1.62)^2$ and thus we note that this theorem is incorrect.
It is intuitively plausible that the order of this method is the same as for the secant method. The use of parallel computation has only reduced the constants involved in the decay of error, not the order. This method is difficult to analyze due to step 4, however, one can decompose it into a composite method with matrix representation

\[ T = \begin{bmatrix} 2 - d_1 - d_2 & d_1 + d_2 \\ 2 - d_3 - d_4 & d_3 + d_4 \end{bmatrix} \]

where \(0 \leq d_i \leq 1\). The matrix \(T\) is the product of \(T_3T_2T_1\), corresponding to step 1, step 2 plus step 3 and step 4. We note that \(T_1\) is 7 by 2, \(T_2\) is 12 by 7 and \(T_3\) is 2 by 12. The matrix \(T_3\) is not constant, but for all possibilities one finds that \(T\) is of the above form for appropriate values of the \(d_i\). A simple analysis shows that the spectral radius of \(T\) is less than or equal to 2 and the computational order is thus at most \(2^{\log_2 1.072} = 1.072\).

A class of methods of higher order are described in [2]. This class has two parameters \(m\) and \(r\) and the matrix \(T\) is of order \(n = m + r - 1\), defined as follows:

for \(i = 1, 2, \cdots, r\),

\[ t_{i,j} = 1 \quad \text{for } i + j \leq n, \quad t_{i,j} = 0 \quad \text{for } i + j \geq n, \]

and for \(i = r + 1, \cdots, m + r - 1\),

\[ t_{i,j} = 1 \quad \text{for } i - j = r, \quad t_{i,j} = 0 \quad \text{otherwise}. \]

This is illustrated for \(m = r = 3\), where we have

\[
T = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \quad \rho = 3.32.
\]

Thus, \(x_i^{r+1}\) is determined by quartic interpolation on \(x_i^j, j = 1, 2, \cdots, 5\); \(x_i^{r+1}\) is determined by cubic interpolation on \(x_i^j, j = 1, 2, 3, 4\); \(x_i^{r+1}\) is determined by quadratic interpolation on \(x_i^j, j = 1, 2, 3\), and \(x_i^{r+1}, x_i^{r+1}\) are taken to be \(x_i^r, x_i^r\), respectively. Note that this class always includes at least one of the points of \(X^i\) in \(X^{i+1}\). Note also that the notation here differs from that of [3].

A composite method simpler and more interesting than those of [3] is given in [1]. It is denoted there by \((1, 2) \circ (2, 1)\). It is the composition of the methods represented by

\[
T_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad T_2 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix};
\]

we have
which has spectral radius of 4.56 and thus the average temporal order is \( \rho = 2.14 \)
and the average computational order is \( \omega = 1.66. \)

c. Two New Methods for Parallel Computation. A little reflection shows that for
a given value of \( n \) (or size of \( T \)), one wants to make the spectral radius of \( T \) as large
as possible. It is easily seen that the largest spectral radius is achieved by Newton’s
method. This method requires that derivatives be available and thus one is naturally
led to ask for the largest spectral radius which does not require derivatives. Two
attractive methods from this point of view are illustrated for the case \( n = 4 \).

Simultaneous \((n - 2)\) degree

\[
T = \begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 0
\end{bmatrix},
\]

\( \rho = n - 1, \quad \omega = (n - 1)^{1/n} \)

Coupled \((n - 1), (n - 2)\) degree

\[
T = \begin{bmatrix}
0 & 1 & 1 & 1 \\
1 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1
\end{bmatrix},
\]

\( \rho = [n - 1 + (4 + (n - 1)^2)^{1/2}] / 2, \quad \omega = \rho^{1/n} \)

d. A Combination Secant and Hermite-Secant Method. We generate a sequence
of 4-tuples of only two distinct points. Given two distinct points, two new ones are
determined by (i) the ordinary secant method and (ii) interpolation of value and
derivative at each of the two points. The rows of \( T \) that represent these two computa-
tions are \((1, 1, 1, 1)\) and \((1, 0, 1, 0)\) and a square matrix may be obtained by adding
any two convex combinations of these rows. These two rows simply add eigenvalues
of zero to the matrix \( T \). The result is

\[
T = \begin{bmatrix}
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0
\end{bmatrix}, \quad \rho = 3, \quad \omega = 3^{1/4} = 1.32.
\]

e. Another Composite Iteration. (This example is taken from the Ph.D. qualifying
exam at Stanford University [4].) Explicitly, we have in the usual notation

\[ x_{n+1} = x_n - f(x_n)/f'(x_n), \quad x_{n+2} = x_{n+1} - f(x_{n+1})/f'(x_n). \]

In order to analyze this method with the current technique, we consider the composi-
tion of three methods represented by

\[
(x_n, x_n) \rightarrow (x_{n+1}, x_n), \quad T_1 = \begin{bmatrix}
1 & 1 \\
1 & 0
\end{bmatrix},
\]

\[ (x_n, x_n) \rightarrow (x_{n+1}, x_{n+2}), \quad T_2 = \begin{bmatrix}
3 & 1 & 1 \\
3 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix},
\]

\[ T = T_2T_1 = \begin{bmatrix}
3 & 1 & 1 \\
1 & 1 & 1
\end{bmatrix},
\]

The values 4.83 and 1.69 of [1] are incorrect and are to be replaced by 4.56 and 1.66, respectively.
We have $T = T_3T_2T_1 = (2^3)^{1/2}$ which has spectral radius $3^{1/2}$.

This method corresponds to replacing the second equation by

$$\tilde{x}_{n+2} = x_{n+1} - \frac{f(x_{n+1})}{f'(\xi_n)} [1 - f''(\xi_n)(\xi_n - x_n)].$$

where $\xi_n$ is a mean value. This results in a "high-order perturbation" of the original method as we have

$$\tilde{x}_{n+2} = x_{n+1} - \frac{f(x_{n+1}) - f(x_n)}{x_{n+1} - x_n} = x_{n+1} - \frac{f(x_{n+1})}{f'(\xi_n)}$$

Although it is not established here, such a "higher-order perturbation" does not affect the order of the method.

We close with a comparison of the three most promising methods for parallel computation, namely Newton, simultaneous $(n - 2)$ degree and coupled $(n - 1)$, $(n - 2)$ degree. We note that none of these schemes are really competitive with the secant method as far as computational order is concerned. The temporal and computational orders are given in Table 1 for these three methods.

<table>
<thead>
<tr>
<th>Coupled $(n - 1)$, $(n - 2)$ Degree Method</th>
<th>Simultaneous $n - 2$ Degree Method</th>
<th>Newton’s Method Degree $(n - 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$n$</td>
<td>Temporal order</td>
<td>Computational order</td>
</tr>
<tr>
<td>----</td>
<td>----------------</td>
<td>---------------------</td>
</tr>
<tr>
<td>3</td>
<td>2.414</td>
<td>1.342</td>
</tr>
<tr>
<td>4</td>
<td>3.303</td>
<td>1.348</td>
</tr>
<tr>
<td>5</td>
<td>4.236</td>
<td>1.335</td>
</tr>
<tr>
<td>6</td>
<td>5.193</td>
<td>1.316</td>
</tr>
<tr>
<td>7</td>
<td>6.162</td>
<td>1.297</td>
</tr>
<tr>
<td>9</td>
<td>8.123</td>
<td>1.262</td>
</tr>
<tr>
<td>11</td>
<td>10.099</td>
<td>1.234</td>
</tr>
</tbody>
</table>