AD-A106 680 TEXAS UNIV AT AUSTIN CENTER FOR CYBERNETIC STUDIES F/G 12/1
NONPOLYNOMIAL AND INVERSE INTERPOLATION FOR LINE SEARCH: SYNTHE--ETC(U)
JUL 81 J BARZILAI, A BEN-TAL N00014-80-C-0242
UNCLASSIFIED CCS-RR-409 NL

1 OF 1
AD A
106680

END
DATE
FILMED
11-81
DTIC

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

LEVEL II

12

Research Report, CCS 409

NONPOLYNOMIAL AND INVERSE INTERPOLATION FOR LINE SEARCH: SYNTHESIS AND CONVERGENCE RATES

by

J. Barzilai*
A. Ben-Tal**

July 1981

(Revision of CCS 385 dated December 1980)

*The University of Texas at Austin
**Technion, Israel Institute of Technology, Haifa, Israel and University of British Columbia, Vancouver, B.C., Canada

This research was partly supported by ONR Contracts N00014-75-C-0569 and N00014-80-C-0242 with the Center for Cybernetic Studies, The University of Texas at Austin and by SSHRC Grant #66-3187, University of British Columbia. Reproduction in whole or in part is permitted for any purpose of the United States Government.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director
Business-Economics Building, 203E
The University of Texas at Austin
Austin, TX 78712
(512) 471-1821

ABSTRACT

The rate of convergence of line search algorithms based on general interpolating functions is derived, and is shown to be independent of the particular interpolating function used. This result holds for the root finding problem f(x) = 0 as well. We show how inverse interpolation can be used in conjunction with the line search problem, and derive its rate of convergence. Our analysis suggests that one-point line search algorithms (in particular Newton's method) are inefficient in a sense. <u>Two</u>-point algorithms using <u>rational</u> interpolating functions are recommended.

KEY WORDS

Nonpolynomial interpolation

Inverse interpolation

Convergence rates

Line search

Root finding

Mathematical programming

1. INTRODUCTION

An essential part of multidimensional minimization algorithms is a line search, i.e., a one-dimensional scheme for the solution of the equation

$$(1) \qquad f'(x) = 0.$$

Most of the line search algorithms in common use are based on polynomial interpolation of f. At iteration i, a polynomial $P_{n,s}(x)$ (the so-called hyperosculatory interpolation polynomial) which coincides with f and its derivatives up to order s-1, at each of the n+1 interpolation points $x_i, x_{i-1}, \ldots, x_{i-n}$, is constructed. The new interpolation point x_{i+1}, is the solution of

$$(2) \qquad P'_{n,s}(x_{i+1}) = 0.$$

In fact, to facilitate the solution of (2), a low degree polynomial is fitted, i.e., $r = s(n+1)$ is small; quadratic and cubic fit being most commonly used.

In recent years, the possibility of using <u>nonpolynomial</u> interpolation functions received some attention. One important situation arises in line searches associated with n-dimensional constrained problems, solved by barrier function methods. A fit by a polynomial cannot capture the singular behavior of the barrier objective function at the boundary of the feasible region. Wright [20] dealt with the case of the logarithmic barrier function. She suggests using the interpolating functions

$$(3) \qquad ax + b + r \log(x-c)$$

$$(4) \qquad ax^2 + bx + c + r \log(x-d).$$

Bjørstad and Nocedal [3] analyze the rate of convergence of an algorithm based on the interpolating function

$$(5) \qquad \frac{ax^2 + bx + c}{(dx + 1)^2} \quad .$$

This function is the one-dimensional restriction of the "conic" model function suggested by Davidon [5], who lists some important advantages of the conic model over the quadratic one.

Independently, we suggested [1] another rational interpolating function

$$(6) \qquad \frac{ax^2 + bx + c}{dx - 1} \quad ,$$

which we analyze in section 3.

Nonpolynomial interpolation was suggested much earlier for the root finding problem

$$(7) \qquad f(x) = 0.$$

Ostrowski [13, p. 82] used in this conjunction the rational function $\frac{ax + b}{cx + d}$, which Jarratt and Nudds [8] and Jarratt [9] generalized to

$$(8) \qquad \frac{x - a}{Q(x)} \quad ,$$

where Q(x) is a polynomial. Ben-Tal and Ben-Israel [2] describe nonpolynomial interpolations by certain types of generalized convex functions.

We formally define the $\underset{\sim}{T}_{n,s}$- interpolation algorithm as follows. Let $n \geqslant 0$, $s \geqslant 1$ be fixed integers and let $\underset{\sim}{T}$ be a family of s-1 times differentiable functions $T:R \to R$, depending on $r = s(n+1)$ parameters. At iteration i, the points x_i, x_{i-1}, ..., x_{i-n} are given, and a function $T \in \underset{\sim}{T}$ is chosen so as to satisfy the interpolation equations

$$(9) \qquad T^{(k)}(x_{i-j}) = f^{(k)}(x_{i-j}) \qquad j = 0, \ldots n;\ k = 0, \ldots, s-1.$$

A new interpolation point is computed from

$$(10) \qquad T'(x_{i+1}) = 0 \quad ,$$

and the oldest point x_{i-n} is deleted.

The practicality of using a particular class $\underset{\sim}{T}$ depends to a great extent on the degree of difficulty of solving the (generally nonlinear) system of equations (9) and equation (10). In the case of the logarithmic functions (3) and (4), equation (10) is easy to solve, but (9) is an ill-conditioned nonlinear system of equations. Wright [20] uses table look-ups, and applies Newton's method after operating some transformations on these equations, in order to solve them.

For the conic model studied by Bjørstad and Nocedal [3], equations (9) and (10) can be reduced to quadratic equations, while for the rational function (6) discussed in section 3, equations (9) are reduced to a linear system, and (10) is very easy to solve.

Note that in the polynomial case, (9) is a linear system. However, (10) is difficult to solve unless T is a low degree polynomial. It will be shown in section 4, that this difficulty can be circumvented by employing inverse interpolation.

Note also, that for the function (8), the interpolation equations can be reduced to a linear system, while the solution of $T(x_{i+1}) = 0$ is simply $x_{i+1} = a$.

In this paper we investigate the rate of convergence of these minimization algorithms. Here we say that the <u>rate of convergence</u> of a sequence $\{x_i\}$ converging to α is p, if there exists a positive number C, such that

$$\frac{|x_{i+1} - \alpha|}{|x_i - \alpha|^p} \longrightarrow C \quad ,$$

(see [19, pp. 1-13]). Ortega and Rheinboldt [11] refer to the rate p defined above as the <u>C-order</u> of the sequence $\{x_i\}$. When it exists, it coincides with their so-called <u>Q- and R- orders</u> (see [11, section 9]).

Rate of convergence analysis is supplied by Bjørstad and Nocedal [3] for the conic function with $s = 2$, $n = 1$. The derivation, which uses a symbol manipulation computer program, is quite elaborate. Moreover, the analysis does not carry over naturally to the study of the convergence properties of an algorithm using the same interpolation function, but with different data say $s = 1$, $n = 3$.

Wright [20] gives no rate of convergence analysis for the algorithms using the logarithmic interpolating functions (3) and (4).

An outline of the paper is as follows. In section 2 we prove rate of convergence theorems for general $\underset{\sim}{T}_{n,s}$-interpolation methods. We show that the rate of convergence is given by the unique positive root of the <u>indicial equation</u>

$$(11) \qquad t^{n+1} - (s-1)t^n - s\sum_{j=0}^{n-1} t^j = 0 \quad .$$

Since this equation depends on n and s only, the rate is independent of the class $\underset{\sim}{T}$.

In section 3 we analyze the specific family of interpolating functions

$$\frac{ax^2 + bx + c}{dx - 1} \quad . \tag{6}$$

Inverse interpolation for minimization algorithms is introduced in section 4. We show that the rate of convergence in this case is again given as the positive root of (11).

Numerical examples illustrating the convergence theorems are given in section 5.

In section 6 we discuss the implications of the rate of convergence analysis to the design of algorithms.

2. RATES OF CONVERGENCE OF NONPOLYNOMIAL ALGORITHMS

Traub [19] studied the rate of convergence of algorithms that use polynomials to interpolate f, or its inverse function for the root finding problem (7). The natural modifications of these results for the minimization problem are discussed by Tamir [17, 18] for the direct polynomial case, i.e., when the interpolation requirements are given by (9), T being a polynomial of degree $< r = s(n+1)$.

The key result for this analysis is the product form formula of the error incurred in hyperosculatory polynomial interpolation (e.g. [6, p. 67]). Ostrowski [13, p. 12] generalized this formula to the case where the interpolating function is not necessarily a polynomial. However, no use of this generalized formula has been made to extend the analysis of Traub and Tamir to the nonpolynomial case. Using this formula, we will obtain a difference equation which differs from the one obtained by Tamir in its right hand side only. This implies that in the nonpolynomial case too, the rate is given by the positive root of the indicial equation (11). Tamir [17, 18] gives two separate proofs for the cases $s = 1$, $s > 1$. We will give a unified proof, and settle his conjectures in [17].

Stronger results than ours can evidently be obtained by relaxing some of our assumptions (compare for example Brent [4]). We have preferred, however, to keep the presentation unobscured by these technicalities. For the same reason, we have not stated explicitly the interval of (local) convergence. This is done in great detail in [17] and repeated in [18].

We will denote by α a solution of (1), and by J the interval $J = \{x: |x - \alpha| \leq L\}$ for some positive L. The error $x_k - \alpha$ will be denoted by e_k, and the open interval determined by $\{a_1, \ldots, a_m\}$ will be denoted by

$\langle a_n, \ldots, a_m \rangle$.

The following assumption will be used repeatedly.

Assumption 1 $r = s(n+1) \geqslant 3$; f and T have continuous derivatives of order $r + 1$ in J for all $T \in \mathfrak{T}$; $f''(\alpha) \neq 0$; $x_i \in J$ and $x_j \neq x_k$ for $j \neq k$, $i, j, k = 0, 1, 2, \ldots, n$; $e_k \neq 0$ for all k.

Note that if $e_i = 0$ for some i, x_i is a solution of (1), and the algorithm is terminated.

In order that the sequence $\{x_i\}$ defined by the algorithm be well defined, the interpolation equations (9), as well as equation (10) for x_{i+1}, must have solutions. If $\mathfrak{T}$ is the class $P_{n,s}$ of polynomials of degree less than $r = s(n+1)$, equations (9) have a solution if and only if $x_k \neq x_\ell$ for $k \neq \ell$. To quote Davis [6, p. 27], the hope that an interpolation problem can always be solved providing the number of parameters equals the number of conditions, is naive. $\mathfrak{T}$ can be replaced by $P_{n,s}$ in iterations at which (9) has no solution, but in practice this case is rather unlikely. We will assume henceforth that (9) has a solution for all i.

As for equation (10), we will prove that under Assumption 1, it has a solution for all i, if L is small enough. We need the following difference relation to prove this and other results.

Theorem 1 Under Assumption 1, if $T'' \neq 0$ on J, then the errors $e_i = x_i - \alpha$, induced by the $\mathfrak{T}_{n,s}$- interpolation algorithm, satisfy the recursion equation:

$$(12) \qquad e_{i+1} = M_i \sum_{k=0}^{n} e_{i-k}^{s-1} \prod_{\substack{j=0 \\ j\neq k}}^{n} e_{i-j}^{s} + N_i \prod_{j=0}^{n} e_{i-j}^{s} \qquad ,$$

where

$$M_i = \frac{M_1(\xi_i(\alpha))(-1)^{r-1}\cdot s}{T''(\theta(x_{i+1}))} \quad , \qquad N_i = \frac{N_1(\eta(\alpha))(-1)^{r}}{T''(\theta(x_{i+1}))} \quad ,$$

$$M_1(x) = \frac{f^{(r)}(x) - T^{(r)}(x)}{r!} \quad , \qquad N_1(x) = \frac{f^{(r+1)}(x) - T^{(r+1)}(x)}{(r+1)!} \quad ,$$

$\xi_i(t)$, $\eta_i(t) \in \langle t, x_i, x_{i-1}, \ldots, x_{i-n} \rangle$ and

$\Theta(x_{i+1}) \in \langle \alpha, x_{i+1} \rangle$.

Proof: The error in the interpolation (9) is given by (see [13, p. 12])

$$(13) \qquad T(t) = f(t) + \frac{T^{(r)}(\xi) - f^{(r)}(\xi)}{r!} \prod_{j=0}^{n} (t - x_{i-j})^s \quad .$$

Differentiating (13) we have

$$(14) \qquad f'(t) = T'(t) + M_1\big(\xi_i(t)\big)W'(t) + N_1\big(\eta_i(t)\big)\; W(t) \quad ,$$

where $W(t) = \prod_{j=0}^{n} (t - x_{i-j})^s$ and M_1, N_1, $\xi_i(t)$ and $\eta_i(t)$ are defined above (for proof see [1, section 6] where we generalize Ralston's result [14, 15] on the differentiation of the error term, to the hyperosculatory case). Substituting $t = \alpha$ in (14) and using

$T'(\alpha) = T'(\alpha) - T'(x_{i+1}) = -e_{i+1}\, T''\big(\Theta(x_{i+1})\big)$ we obtain (12).

□

Under Assumption 1, $f''(\alpha) \neq 0$. Since $f'(\alpha) = 0$, f' must change its sign at α. It follows by substituting $t = \alpha - L$ and $t = \alpha + L$ in (14), that T' also has opposite signs at these two points (for a detailed proof see Appendix A in [18]), if L is small enough. We summarize this result in

<u>Theorem 2</u> Under Assumption 1, if L is small enough, there exists $x_{i+1} \in J$ satisfying equation (10).

Using Theorem 1 in [7, chapter 6, section 5], it follows immediately from the difference equation (12), that if the initial errors $e_0, \ldots, e_n$ are small enough (i.e., L is small enough), the sequence e_i tends to zero, establishing the following local convergence result.

<u>Theorem 3</u> Under Assumption 1, if L is small enough, the sequence $\{x_i\}$ converges to the solution α of (1). $\square$

Also note that if L is small enough, and if $s > 1$, we have by (12) $|e_{i+1}| < |e_i|$, implying $x_{i+1} \neq x_i$. For $s = 1$ however, we have to <u>assume</u> $x_{i+1} \neq x_i$ (cf. [16]).

We now replace (12) by a more useful difference equation.

<u>Theorem 4</u> Under the assumptions of Theorem 3, and if the sequences $\{M_i\}$, $\{N_i\}$ are bounded, then

$$(15) \qquad e_{i+1} = A_i \; e_i^{s-1} \prod_{j=1}^{n} e_{i-j}^{s}$$

with $\{A_i\}$ bounded.

<u>Proof.</u> By assumption, the sequences M_i, N_i are bounded. If $s \geqslant 2$, (12) implies

$$(16) \qquad \frac{e_{i+1}}{e_i} \to 0 \; ,$$

(i.e. superlinear convergence). If $s = 1$, we must have $n \geq 2$ since we assumed $r = s(n+1) \geq 3$. For $n = 2$, (12) is the basic difference relation governing the behavior of the Quadratic Fit algorithm, which is known to converge superlinearly (see Theorem 3.4.1 in Brent [4]). It is evident from (12) that the rate for $n > 2$ is not less than the rate for $n = 2$. Therefore, (16) holds for all $s \geq 1$, $n \geq 0$ if $r = (n+1)s \geq 3$. Rewriting (12) in the form

$$(17) \qquad e_{i+1} = e_i^{s-1} \prod_{j=1}^{n} e_{i-j}^{s} \left[M_i + M_i \sum_{k=1}^{n} \frac{e_i}{e_{i-k}} + N_i e_i \right],$$

we see by (16) that (15) holds with

$$A_i = M_i \left[1 + \sum_{k=1}^{n} \frac{e_i}{e_{i-k}} \right] + N_i e_i \; .$$

□

Remark In [17, Appendix C], Tamir conjectures that his apriory assumption [17, Assumption 2] on the superlinear convergence of the sequence $\{e_i\}$ is redundant. Our proof shows that this assumption is indeed redundant.

We now state our main result.

Theorem 5 Under the assumptions of Theorem 4, the sequence $\{x_i\}$ generated by the $\underset{\sim}{T}_{n,s}$- interpolation algorithm converges to the solution α of (1), with Q- and R-rates of convergence at least p, where p is the unique positive root of the equation

$$t^{n+1} - (s-1)t^n - s\sum_{j=0}^{n-1} t^j = 0 \; .$$

Proof Convergence of $\{x_i\}$ to α is proved in Theorem 3.

Tamir [17,18] proves, under the additional assumption $A_i \to A \neq 0$, that the C- (and therefore Q- and R-) rate is exactly p.

Now $A_i \to A \neq 0$ is the worst possible case, for if the sequence $\{A_i\}$ is bounded (not necessarily convergent), equation (15) implies Q-rate of convergence at least p (even though the C-order may fail to exist).

Indeed, the proof in [17,18] is based on a similar result of Traub [19] for the root finding problem. We will indicate the slight modifications necessary in the proof, for the latter problem. First, it is evident that if the sequence $\{\beta_i\}$ of Theorem 3-1 in [19] is bounded above, so is the sequence $\{\sigma_i\}$. This in turn implies that the sequence $\{D_i\}$ of Theorem 3-3 of [19] is bounded above, or equivalently $\limsup \frac{|e_{i+1}|}{|e_i|^p} < +\infty$, hence the Q-order of the sequence $\{x_i\}$ is at least p.

The assertion on the R-order follows from Theorem 9.3.2 in [11, section 9].

$\square$

Remark If the mapping from $x_{i-n}, \ldots, x_i$ to the parameters of T defined by (9) is continuous, the sequences $\{M_i\}$, $\{N_i\}$ are convergent, hence bounded, as assumed in Theorem 5.

Corollary The rate of convergence of the sequence generated by the interpolation algorithm does not depend on the class of interpolating functions $\underset{\sim}{T}$.

Remark It is evident from our analysis that the above corollary holds for the root finding problem, as well as for the case when the number of pieces of information used at the interpolation point x_{i-j} depends on j (e.g. the False Position Method).

It follows from Theorem 5, that the rates of convergence of the interpolation algorithms using the conic interpolating function (5) is $p = 1.46$ for $s = 1$, $n = 3$ (4 interpolation points with no derivatives); $p = 2$ for $s = 2, n = 1$

(f and f' used at two points) and $p = 3$ for $s = 4$, $n = 0$ (f, f', f", f"' used at one interpolation point). Rates of convergence of algorithms using the interpolating functions mentioned in the introduction, can be computed likewise.

The behavior of the rate p as a function of n , for fixed s, is summarized in Theorem 6.

<u>Theorem 6</u> For fixed s, p is an increasing function of n. For $n = 0$, $p = s - 1$, while for $n = 1$, $p = s$. As n tends to infinity, p tends to $\frac{s}{2} + \sqrt{\left(\frac{s}{2}\right)^2 + 1}$.

Proof For $n = 0$, $n = 1$, the rate is obtained by solving the indicial equations $t - (s-1) = 0$ and $t^2 - (s-1)t - s = 0$ respectively. The remaining assertions are proved in Tamir [17, 18]. □

A few numerical values for p, are listed in Table 2.1.

TABLE 2.1

s	n	p
1	2	1.3
	3	1.4
	∞	1.6
2	1	2
	2	2.3
	∞	2.4
3	0	2
	1	3
	∞	3.3
s	0	s-1
	1	s
	∞	$\frac{s}{2} + \sqrt{\left(\frac{s}{2}\right)^2 + 1}$

Since $\frac{s}{2} + \sqrt{\left(\frac{s}{2}\right)^2 + 1}$ is close to s even for small values of s (see Table 1), Theorem 6 implies that algorithms using more than two interpolation points ($n > 1$) are inefficient. However, two points algorithms are substantially faster than one point (or memoryless) algorithms. Instead of making the last statement precise by defining a measure of efficiency (chosen carefully to suit the authors' purpose), we will note that the transition from $n = 0$ to $n = 1$ involves storage (but no computation) of s extra pieces of data. In addition to this, the system of equations (9) will involve 2s instead of s unknowns. However, this system is linear in the polynomial and rational cases (which are the most important ones) and need to be solved once only for the class T. The main difficulty is the solution of equation (10). This, in the case of $s = 3$, $n = 1$ (Newton's Method with memory) with polynomial interpolation, is a polynomial equation of degree 4. Solution of this equation can be avoided by using inverse interpolation, to be discussed in section 4. On the other hand, for line search algorithms, computation of $f^{(k)}(t)$ involves in fact computation of the derivatives of a function on R^n (i.e., gradient vectors and Hessian matrices,) making the extra effort worthwhile.

3. A CLASS OF RATIONAL INTERPOLATING FUNCTIONS

In this section we briefly discuss the four parameter rational interpolating function

$$R(x) = \frac{ax^2 + bx + c}{dx-1} . \tag{18}$$

Writing (18) in the form

$$(dx-1)R(x) = ax^2 + bx + c , \tag{19}$$

differentiating (19) implicitly and then using the interpolation equations (9), leads to a linear system of equations for the coefficients a,b,c,d. For example, with data $s = 4$, $n = 0$, the equations are

$$\begin{aligned}
&(dx_i-1)f(x_i) = ax_i^2 + bx_i + c \\
&(dx_i-1)f'(x_i) + df(x_i) = 2ax_i + b \\
&(dx_i-1)f''(x_i) + 2df'(x_i) = 2a \\
&(dx_i-1)f'''(x_i) + 3df''(x_i) = 0 .
\end{aligned}$$

Note that if $d = 0$, $R(x)$ has no singularity. Therefore, it may be expected that $R(x)$ will provide a good fit to functions with regular or singular behavior.

We now turn our attention to the solution of (10) for x_{i+1}. If $d = 0$, $R(x)$ is a quadratic and (10) yields $x_{i+1} = -\frac{b}{2a}$. For $d \neq 0$, it is convenient to rewrite $R(x)$ in the form

$$R(x) = \alpha x + \beta + \frac{\gamma}{x-\delta} \tag{20}$$

where $\alpha = \frac{a}{d}$, $\beta = \frac{b}{d} + \frac{a}{d^2}$, $\gamma = \frac{c}{d} + \frac{b}{d^2} + \frac{a}{d^3}$, $\delta = \frac{1}{d}$.

Differentiating (20) we have

$$R'(x) = \alpha - \frac{\gamma}{(x-\delta)^2} \tag{21}$$

$$R''(x) = \frac{2\gamma}{(x-\delta)^3} \quad . \tag{22}$$

From (22) we see that $R''(x)$ has exactly one change of sign at $x = \delta$. The point x_{i+1} will be a minimum of f if

$$R'(x_{i+1}) = 0 \tag{23}$$

$$R''(x_{i+1}) > 0 \ . \tag{24}$$

From (21)-(24) we have

$$x_{i+1} = \delta \pm \sqrt{\gamma/\alpha} \tag{25}$$

assuming

$$\alpha\gamma > 0 \ . \tag{26}$$

The two solutions in (25) correspond to the minimum point of the convex branch of R, and the maximum point of the concave branch of R. Multiplying (22) by $(x-\alpha)^4$, we see that in order for (24) to hold, we must have $\gamma(x_{i+1}-\delta) > 0$, which combined with (25) yields

$$x_{i+1} = \delta + \varepsilon\sqrt{\gamma/\alpha}\,, \quad \varepsilon = \text{sign}\,\gamma\ .$$

Condition (26) will hold near the solution under the assumptions of Theorem 2.

Remarks. Rational interpolations are particularly useful in cases where f, or its derivatives, have rapid changes, even when f has no singularities (see section 5).

Use of rational functions other than (5) and (6) suggests itself, especially when higher degree interpolation is needed, possibly combined with inverse interpolation (see section 4).

4. INVERSE INTERPOLATION FOR LINE SEARCH

Inverse interpolation methods for the root finding problem $f(x) = 0$ are well known. Assuming that f' is nonzero and $f^{(r)}(x)$ is continuous on an interval J mapped by f onto K, then f has an inverse F, and $F^{(r)}$ is continuous on K. If T is a hyperosculatory interpolating function satisfying

$$(27) \qquad \begin{cases} T^{(k)}(y_{i-j}) = F^{(k)}(y_{i-j}) & j = 0,\ldots,n;\ k = 0,\ldots,s-1\ , \\ y_{i-j} = f(x_{i-j}) & \end{cases}$$

then

$$(28) \qquad F(t) = T(t) + \frac{F^{(r)}(\theta_i(t)) - T^{(r)}(\theta_i(t))}{r!} \prod_{j=0}^{n} (t-y_{i-j})^s\ ,$$

with $\theta_i(t) \in \langle t, y_i, y_{i-1}, \ldots, y_{i-n} \rangle$. In the inverse interpolation algorithm for the root finding problem, we approximate $\alpha = F(0)$ by $x_{i+1} = T(0)$.

The derivatives of the inverse function F can be expressed in terms of the derivatives of f. Indeed, letting

$$\beta_k = F^{(k)}\ , \quad \alpha_k = f^{(k)}\ , \quad k = 1,2,\ldots\ ,$$

we have (see [12])

$$(29) \qquad \beta_k = \sum (-1)^{n-k_1-1} \frac{(2n-k_1-2)!}{n!k_2!k_3!\cdots k_n!} \alpha_1^{-(2n-k_1-1)} \cdot \alpha_2^{k_2} \cdot \ldots \cdot \alpha_n^{k_n}\ ,$$

where the summation is taken over all $k_1, k_2, \ldots, k_n$ satisfying

$$\sum_{i=1}^{n} k_i = n-1\ , \quad \sum_{i=1}^{n} ik_i = 2n-2\ , \quad k_i \geq 0\ .$$

Let T be a polynomial $Q_{n,s}$ of degree $< r$. By the above and since $y_{i-j} = f(x_{i-j})$ and $F(y_{i-j}) = x_{i-j}$, $Q_{n,s}$ can be expressed in terms of the data x_{i-j} and $f^{(k)}(x_{i-j})$. If T is not a polynomial, we first construct the interpolating polynomial $Q_{n,s}$ satisfying (27), and proceed to solve the system

$$T^{(k)}(y_{i-j}) = Q^{(k)}_{n,s}(y_{i-j}) \quad j = 0,\ldots,n;\ k = 0,\ldots,s-1 ,$$

$$y_{i-j} = f(x_{i-j}) .$$

Traub [19] shows that the rate of convergence of the polynomial inverse interpolation algorithm is given by the positive root of the (root finding) indicial equation $t^{n+1} - s\sum_{j=0}^{n} t^j = 0$, exactly as in the case of direct polynomial interpolation. Similar to our derivation in section 2, it can be shown that the rate of convergence is independent of the interpolating class of functions.

Inverse interpolation has not been applied so far to the solution of line search problems. We will define the $\underset{\sim}{T}_{n,s}$-inverse interpolation algorithm, and prove that under the appropriate assumptions, its rate of convergence is given by the positive solution of the indicial equation (11).

A difficulty in applying inverse interpolation to the line search problem is that one cannot assume that f has an inverse near an extremum point α, since necessarily $f'(\alpha) = 0$. Denoting, however, $g = f'$ we can write equation (1) as $g(\alpha) = 0$. Assuming that α is a simple zero of g, g has an inverse G defined on a neighborhood of $g(\alpha)$. Since the solution α of (1) satisfies $g(\alpha) = 0$, it is given by

$$\alpha = G(0) . \tag{30}$$

The assumption on the differentiability of g implies that G is differentiable. Hence G is continuous and has a primitive function F (i.e., an indefinite integral of G), satisfying

$$F'(t) = G(t) \ . \tag{31}$$

Equation (31) determines F up to an additive constant. By (30) α is given in terms of any solution F of (31) by

$$\alpha = F'(0) \ . \tag{32}$$

Now let F be any solution of (31), and let T be a hyperosculatory interpolating function satisfying

$$\begin{cases} T^{(k)}(y_{i-j}) = F^{(k)}(y_{i-j}) & j = 0,\ldots,n; \ k = 0,\ldots,s-1 \ , \\ y_{i-j} = g(x_{i-j}) \ . \end{cases} \tag{33}$$

The inverse interpolation process for the solution of (1) consists of approximating α in (32) by

$$x_{i+1} = T'(0) \ . \tag{34}$$

Evidently, x_{i+1} as defined by (34) is independent of the particular integration constant associated with F. Let $Q_{n,s}$ be the interpolation polynomial of degree $< r$ satisfying (33). We will later express (34) in this case in terms of the data. If T is not a polynomial, we can express equations (33) in terms of the data by first constructing $Q_{n,s}$ (i.e., replace T by $Q_{n,s}$ in (33)) and then interpolate $Q_{n,s}$ by T (i.e., replace F by $Q_{n,s}$ in (33)).

In order to write (34) explicitly in the polynomial case, we can proceed to construct $P_{n,s}(x)$, the direct interpolating polynomial determined by (9), differentiate it to obtain

$$P'_{n,s}(x) = \sum_{k=0}^{r-2} \alpha_k (x-x_i)^k$$

from which we obtain directly by [12] <u>the inverse interpolation formula for line search</u>

(35) $$x_{i+1} = x_i + \sum_{k=1}^{r-2} \beta_k (-\alpha_0)^k$$

where β_k is given in terms of the α_k's in (29).

<u>Remark</u>. If $P(x)$ is a <u>quadratic</u> (which is the case for the classical Newton, False Position and Quadratic Fit methods), $P'(x)$ is a linear function, with linear inverse, so that in this case the direct and inverse interpolation formulas coincide.

The inverse interpolation formula for $s=4$, $n=0$ (with rate 3), will differ by the above argument from the direct interpolation formula for this case. It is given by

(36) $$x_{i+1} = x_i - \frac{f'_i}{f''_i} - \frac{1}{2}\frac{(f'_i)^2 f'''_i}{(f''_i)^3} .$$

Note that omitting the term with the third order derivative in (36) yields Newton's method. Note also that the direct interpolation formula in this case is given by the solution of the quadratic equation

$$P'_{0,3}(x_{i+1}) = 0, \quad \text{where}$$

$$P_{0,3}(x) = f_i + (x-x_i)f'_i + \tfrac{1}{2}(x-x_i)^2 f''_i + \frac{1}{6}(x-x_i)^3 f'''_i \quad .$$

We now turn to the analysis of rate of convergence of this class of algorithms, starting with the derivation of a basic difference equation.

Theorem 7. Let $f'' \neq 0$ and let $f^{(r+1)}$, $T^{(r+1)}$ be continuous on an interval J. Let the derivative G of F be the inverse of $g = f'$, and let $x_{i+1}, x_i, \ldots, x_{i-n} \in J$, where $x_{i+1} = T'(0)$ and T satisfies (33). Then

$$e_{i+1} = \sum_{k=0}^{n} K_{i-k}\, e_{i-k}^{s-1} \prod_{\substack{j=0\\ j\neq k}}^{n} e_{i-j}^{s} + L_i \prod_{j=0}^{n} e_{i-j}^{s}, \tag{37}$$

where

$$K_{i-k} = (-1)^r s K_1(\xi_i(0))[f''(\theta_{i-k})]^{s-1} \prod_{\substack{j=0\\ j\neq k}}^{n} [f''(\theta_{i-j})]^s ,$$

$$L_i = (-1)^{r+1} L_1(\eta_1(0)) \prod_{j=0}^{n} [f[(\theta_{i-j})]^s ,$$

$$K_1(x) = \frac{F^{(r)}(x) - T_{n,s}^{(r)}(x)}{r!} , \quad L_1(x) = \frac{F^{(r+1)}(x) - T_{n,s}^{(r+1)}(x)}{(r+1)!} ,$$

$$\xi_i(t), \eta_i(t) \in \langle t, y_i, y_{i-1}, \ldots, y_{i-n}\rangle, \text{ and } \theta_{i-j} \in \langle x_{i-j}, \alpha\rangle .$$

Proof. The proof is similar to the proof of Theorem 1 and will be omitted.

□

The interested reader can find the proof of Theorem 7 as well as the inverse interpolation formulas for the cases $n = 1$, $s = 2$ and $n = 1$, $s = 3$ in [1].

The following theorem characterizes the behavior of the inverse interpolatory process for the line search problem (1).

Theorem 8. Let f and T have continuous derivatives of order $r+1$, in the interval $J = \{x: |x-\alpha| \leq L\}$. Let $f''(\alpha) \neq 0$. If L is small enough, if $x_0, \ldots, x_n \in J$ and the sequence $\{x_i\}$ is constructed by the inverse interpolation algorithm for line search (i.e. $x_{i+1} = T'(0)$, where T satisfies (27)), then $x_{i+1} \in J$. Furthermore, if the algorithm does not terminate, and the sequence $\{K_i\}$ defined in Theorem 7 is bounded, then $x_i \to \alpha$ with Q- and R-rates of convergence at least p, where p is the unique positive root of

$$t^{n+1} - (s-1)t^n - s\sum_{j=0}^{n-1} t^j = 0 \ . \tag{38}$$

Proof. The proof is identical with the proof of Theorem 5.

Equation (38) is identical of course with equation (11) which is the indicial equation of the derived difference equation (15).

5. NUMERICAL EXAMPLES

The purpose of this section is to illustrate that the theoretical rate of convergence predicted by the preceding theorems, is well reflected in the actual behavior of the various direct and inverse algorithms. Ten algorithms (without safeguards) are applied to minimizing two functions:

$$f(x) = \frac{1}{6}x^6 - x^3 + 2x$$

and

$$f(x) = x + \frac{1}{e^{x-1}-1} .$$

The first function, although nonsingular, behaves very much like a singular one in the interval $[0,2]$, due to rapid changes of f and its derivatives in the interval. This in particular caused the cubic fit method to diverge.

The second function is highly singular at $x=1$. For this function, three of the methods based on polynomial interpolation diverged. In contrast, all four methods based on rational interpolation worked well.

The results are summarized in Tables 5.1, and 5.2. The Rational and Conic functions referred to in these tables are $\frac{ax^2+bx+c}{dx-1}$ and $\frac{ax^2+bx+c}{(dx+1)^2}$ respectively. Initial values used in Table 5.1 are $\{2\}$, $\{2,2.1\}$, $\{1.9,2,2.1\}$, and $\{2,2.1,2.3,2.3\}$ according to the number of interpolation points. Initial values used in Table 5.2 are $\{1.75\}$, $\{1.7,1.8\}$, $\{1.7,1.75,1.8\}$, and $\{1.7,1.73,1.77,1.8\}$. The $Q_p(x)$'s are the quotients $\frac{\|x_{i+1}-x^*\|}{\|x_i-x^*\|^p}$, where x^* are the solutions 1.120742611 and 1.962423650 respectively.

The algorithms were coded in APLSF-V01 on a Digital Equipment Corporation DEC-10 computer, using double precision arithmetic. We stopped when $|f'(x)| \leq 10^{-8}$, except for the Quadratic Fit algorithm which was terminated after 12 steps on the first function.

The following theorem characterizes the behavior of the inverse interpolatory process for the line search problem (1).

Theorem 8. Let f and T have continuous derivatives of order $r+1$, in the interval $J = \{x: |x-\alpha| \leq L\}$. Let $f''(\alpha) \neq 0$. If L is small enough, if $x_0, \ldots, x_n \in J$ and the sequence $\{x_i\}$ is constructed by the inverse interpolation algorithm for line search (i.e. $x_{i+1} = T'(0)$, where T satisfies (27)), then $x_{i+1} \in J$. Furthermore, if the algorithm does not terminate, and the sequence $\{K_i\}$ defined in Theorem 7 is bounded, then $x_i \to \alpha$ with Q- and R-rates of convergence at least p, where p is the unique positive root of

$$t^{n+1} - (s-1)t^n - s\sum_{j=0}^{n-1} t^j = 0 . \tag{38}$$

Proof. The proof is identical with the proof of Theorem 5.

Equation (38) is identical of course with equation (11) which is the indicial equation of the derived difference equation (15).

TABLE 5.1: Solution of $f'(x) = 0$, $f(x) = \frac{1}{6}x^6 - x^3 + 2x$

Algorithm	Iterations No.	$f'(x)$	$x - x^*$	$Q_p(x)$
	0	2.961101000E1	9.792573887E¯1	
	1	6.728546951E0	5.529105297E¯1	5.684783867E¯1
	2	4.972863922E0	4.863864194E¯1	1.066328870E0
Polynomial	3	2.882038163E0	3.815942441E¯1	9.914324335E¯1
(Quadratic Fit)	4	1.346951822E0	2.647918769E¯1	9.487772154E¯1
	5	7.684644291E¯1	1.976246895E¯1	1.149025754E0
	6	3.992167669E¯1	1.364989189E¯1	1.169340571E0
Data: f at 3 points	7	1.997530860E¯1	8.876782058E¯2	1.241553450E0
	8	1.005668367E¯1	5.535730732E¯2	1.369114301E0
	9	4.718635448E¯2	3.104070036E¯2	1.435060514E0
	10	2.034304172E¯2	1.523054207E¯2	1.515237702E0
Rate: 1.3	11	7.613347509E¯3	6.175096344E¯3	1.577732360E0
	12	2.210373360E¯3	1.865705167E¯3	1.576220344E0
	0	5.049343000E1	1.179257389E0	
	1	3.709091269E0	4.277185667E¯1	3.359004532E¯1
Rational	2	2.281684841E0	3.422946382E¯1	1.188404178E0
	3	1.040389540E0	2.320417711E¯1	1.116698663E0
	4	3.936613400E¯1	1.353720412E¯1	1.151693577E0
	5	1.297609663E¯1	6.634405486E¯2	1.243392255E0
Data: f at 4 points	6	5.030661751E¯2	3.267536431E¯2	1.741620451E0
	7	1.550631811E¯2	1.194496127E¯2	1.797630915E0
	8	3.409830478E¯3	2.851697807E¯3	1.875539531E0
Rate: 1.4	9	4.146773605E¯4	3.550433290E¯4	1.905497807E0
	10	2.060197138E¯5	1.769587929E¯5	2.012243570E0
	11	2.394566790E¯7	2.057134096E¯7	1.896026810E0
	12	3.616562856E¯10	3.106937485E¯10	1.959848404E0
	0	2.200000000E1	8.792573887E¯1	
Polynomial	1	6.811135228E0	5.557279770E¯1	7.188366439E¯1
(Newton)	2	2.037118820E0	3.243497512E¯1	1.050241190E0
	3	5.799609462E¯1	1.692501469E¯1	1.608799476E0
	4	1.528615287E¯1	7.426608718E¯2	2.592581600E0
Data: f, f', f" at 1 point	5	3.407897860E¯2	2.375875756E¯2	4.307672124E0
	6	4.649413295E¯3	3.852395563E¯3	6.824697750E0
Rate: 2	7	1.546338457E¯4	1.326761876E¯4	8.939870659E0
	8	1.945599880E¯7	1.671434406E¯7	9.495183683E0
	9	3.094629587E¯13	2.658559137E¯13	9.516289591E0

TABLE 5.1 (continued): Solution of $f'(x) = 0$, $f(x) = \frac{1}{6}x^6 - x^3 + 2x$

Algorithm	No.	$f'(x)$	$x - x^*$	$Q_p(x)$
Rational	0	2.961101000E1	9.792573887E¯1	
	1	2.810902098E0	3.772360335E¯1	3.933865033E¯1
Data: f, f' at 2 points	2	8.691731487E¯1	2.111129278E¯1	1.483503206E0
	3	1.523995812E¯1	7.411334564E¯2	1.662902405E0
	4	2.864145851E¯2	2.050884987E¯2	3.733777795E0
Rate: 2	5	2.435379612E¯3	2.052034818E¯3	4.878677532E0
	6	3.036533579E¯5	2.607995338E¯5	6.193517372E0
	7	4.411705579E¯9	3.790033105E¯9	5.572234465E0
Inverse Polynomial	0	2.961101000E1	9.792573887E¯1	
	1	5.486944138E0	5.073422766E¯1	5.290629378E¯1
	2	1.395258096E¯1	6.976612359E¯2	2.710456778E¯1
Data: f, f' at 2 points	3	6.937587796E¯1	¯4.193049971E¯1	¯8.614713769E1
	4	1.185756774E¯1	¯2.000676860E¯1	¯1.137933473E0
	5	2.169400795E¯2	¯1.399295964E¯1	¯3.495873294E0
Rate: 2	6	8.943116552E¯4	¯1.216314003E¯1	¯6.211929863E0
	7	1.741909068E¯6	¯1.207443531E¯1	¯8.161602061E0
	8	6.889045107E¯12	¯1.207426113E¯1	¯8.281841327E0
Conic	0	2.961101000E1	9.792573887E¯1	
	1	2.216090916E0	3.376086171E¯1	3.520625326E¯1
	2	6.181358832E¯1	1.753877494E¯1	1.538764680E0
Data: f, f' at 2 points	3	9.276089320E¯2	5.219436396E¯2	1.696778292E0
	4	1.397758065E¯2	1.086966678E¯2	3.989964315E0
	5	6.087564621E¯4	5.203953372E¯4	4.404543820E0
Rate: 2	6	1.681494620E¯6	1.444528151E¯6	5.334076214E0
	7	1.126799608E¯11	9.680174631E¯12	4.639072635E0
Inverse Polynomial	0	2.961101000E1	9.792573887E¯1	
Data: f,f',f"	1	2.803096043E0	3.767533945E¯1	4.012052457E¯1
at 2 points	2	2.236092670E¯1	9.550086875E¯2	1.785812261E0
	3	9.894209873E¯3	7.900627329E¯3	9.070675067E0
Rate: 3	4	1.778781896E¯6	1.528103896E¯6	3.098618810E0
	5	¯8.673617380E¯19	0	0
Inverse Polynomial	0	2.200000000E1	8.792573887E¯1	
	1	3.896702674E0	4.372031449E¯1	6.431839487E¯1
Data: f,f',f",f"' at	2	6.374399616E¯1	1.784081162E¯1	2.134837269E0
1 point	3	8.957458010E¯2	5.087163547E¯2	8.958429431E0
	4	7.053276768E¯3	5.743407965E¯3	4.362571583E1
Rate: 3	5	3.047780125E¯5	2.617652282E¯5	1.381665641E2
	6	3.552651662E¯12	3.052032585E¯12	1.701583625E2

TABLE 5.1 (continued): Solution of $f'(x) = 0$, $f(x) = \frac{1}{6}x^6 - x^3 + 2x$

Algorithm	No.	f'(x)	$x - x^*$	$Q_p(x)$
Rational Data: f,f',f",f''' at 1 point Rate: 3	0	2.200000000E1	8.792573887E¯1	
	1	2.418567463E0	3.517996559E¯1	5.175440627E¯1
	2	2.478962995E¯1	1.019923738E¯1	2.342510081E0
	3	1.858685452E¯2	1.405709876E¯2	1.324928972E1
	4	1.651599946E¯4	1.416954968E¯4	5.101160170E1
	5	2.300722529E¯10	1.976517860E¯10	6.947564682E1

TABLE 5.2: Solution of $f'(x) = 0$, $f(x) = x + 1/(e^{x-1} - 1)$

Algorithm	No.	f'(x)	$x - x^*$	$Q_p(x)$
Polynomial (Quadratic Fit) Data: f at 3 points Rate: 1.3	0	¯4.817670946E¯1	¯1.624236501E¯1	
	1	¯1.595324554E¯1	¯6.425557875E¯2	7.138022810E¯1
	2	¯6.867413860E¯2	¯2.931057467E¯2	1.112259825E0
	3	¯1.980314467E¯2	¯8.735336800E¯3	9.376484668E¯1
	4	¯3.136116528E¯3	¯1.399442557E¯3	7.467562867E¯1
	5	¯3.604098014E¯4	¯1.611395109E¯4	9.727620112E¯1
	6	¯1.616424827E¯5	¯7.228789794E¯6	7.646332165E¯1
	7	¯2.762386271E¯7	¯1.235376457E¯7	7.981553749E¯1
	8	¯2.808337059E¯9	¯1.255926503E¯9	1.779843092E0
Rational Data: f at 4 points Rate: 1.4	0	¯4.817670946E¯1	¯1.624236501E¯1	
	1	¯1.793161898E¯4	¯8.018257350E¯5	1.150614171E¯3
	2	¯4.093437144E¯5	¯1.830588290E¯5	1.842693963E1
	3	¯1.806096932E¯8	¯8.077110918E¯9	7.083809281E¯2
	4	¯1.465487887E¯12	¯6.553780955E¯13	4.753184221E¯1
Polynomial (Newton) Data: f,f',f" at one point Rate: 2	0	¯6.967368901E¯1	¯2.124236501E¯1	
	1	¯1.623305337E¯1	¯6.527012224E¯2	1.446467539E0
	2	¯1.470965849E¯2	¯6.511392281E¯3	1.528428081E0
	3	¯1.480594988E¯4	¯6.620735904E¯5	1.561559526E0
	4	¯1.534158280E¯8	¯6.860964325E¯9	1.565210065E0
	5	¯1.647987302E¯16	¯6.548581122E¯17	1.391159384E0

TABLE 5.2 (continued): Solution of $f'(x) = 0$, $f(x) = x + 1/(e^{x-1} - 1)$

Algorithm	No.	$f'(x)$	$x - x^*$	$Q_p(x)$
Rational Data: f, f' at 2 points Rate: 2	0	¯4.817670946E¯1	¯1.624236501E¯1	
	1	¯1.754343274E¯4	¯7.844698287E¯5	2.973566893E¯3
	2	¯2.441789574E¯8	¯1.092001475E¯8	1.774478474E0
	3	¯1.864827737E¯17	0	0
Conic Data: f, f' at 2 points Rate: 2	0	¯4.817670946E¯1	¯1.624236501E¯1	
	1	1.617959193E¯2	7.318732791E¯3	2.774197391E¯1
	2	¯1.324951426E¯4	¯5.924813411E¯5	1.106121656E0
	3	1.862712947E¯9	8.330305638E¯10	2.373075636E¯1
Inverse Polynomial Data: f, f', f'', f''' at one point Rate: 3	0	¯6.967368901E¯1	¯2.124236501E¯1	
	1	¯5.067809765E¯2	¯2.189048593E¯2	2.283740747E0
	2	¯6.296381781E¯5	¯2.815703434E¯5	2.684236045E0
	3	¯1.364355677E¯13	¯6.100762256E¯14	2.732897665E0
Rational Data: f, f', f'', f''' at one point Rate: 3	0	¯6.967368901E¯1	¯2.124236501E¯1	
	1	¯1.846146248E¯4	¯8.255150214E¯5	8.612245055E¯[illegible]
	2	¯1.397579968E¯14	¯6.241968747E¯15	1.109549416E¯[illegible]

6. CONCLUDING REMARKS

Our analysis points to the inefficiency of interpolation algorithms based on more than two interpolation points (or more than three points if function values only are used). Two-point algorithms are significantly faster than one-point algorithms, the latter are therefore useful only if computation of the derivatives of f are relatively very cheap.

Use of inverse interpolation is recommended if equation (10) is difficult to solve. Note that even in the Cubit Fit case where the interpolating function is a cubic, solution of equation (10) involves computation of square roots (see [10. p. 142]), in itself a relatively costly operation on the computer.

Moreover, our results allow the design of a one-dimensional minimization algorithm without regard to the choice of the r-parameter family of interpolating functions used. Special structure of the problem can be taken advantage of by using appropriate r-parameter interpolants with the assurance that the rate of convergence will not be impaired. Also, when combined with a safeguarding technique, one might want to compute several guesses to the minimizer based upon different r-parameter interpolation of the same data, and then use the "best" guess. This can be used to great advantage, for example, when one guess is outside the interval of uncertainty or undefined. Clearly, the rate of convergence will be unhampered even though iterates may be selected from a (finite) number of r-parameter families of interpolants.

Note that the procedure of safeguarding by bracketing as suggested in [10, section 7.3], may severely affect the rate of convergence, since the basic difference equations may be fundamentally changed by such modifications.

Assume, for example, that we modify the Quadratic Fit algorithm so that one of the points $x_{i+1}, x_i, x_{i-1}, x_{i-2}$ (not necessarily x_{i-2}) is dropped, in such a manner that the remaining points bracket the solution α. Then we may choose $e_3 < 0 < e_2 < e_1$ and small enough L, such that equation (15) of Theorem 4 would imply that, for $M > 0$, we have $e_{i+1} > 0$ for all i. Hence, in the bracketing algorithm, one of the three interpolation points is fixed as x_3, and in the difference relation (15) one of the indexes should be replaced by 3, leading to difference equation with an indicial equation different than (11).

Thus the statement in Tamir [17], that bracketing algorithms do not lend themselves to the difference equation approach, and the conjecture made there that the interpolation and the bracketing algorithms have the same rates of convergence, are both false.

A bracketing procedure that aims at maintaining the rate of convergence of the underlying interpolation, should coincide with it near the solution.

REFERENCES

[1] J. Barzilai and A. Ben-Tal, Nonpolynomial and inverse interpolation for lime search: synthesis and convergence rates, Research Report 385, Center for Cybernetic Studies, The University of Texas at Austin, 1980.

[2] A. Ben-Tal and A. Ben-Israel, F-convex functions: properties and applications, Generalized Concavity in Optimization and Economics, M. Avriel, S. Schaible and W.T. Ziemba, eds., Academic Press, New York, 1981.

[3] P. Bjørstad and J. Nocedal, Analysis of a new algorithm for one-dimensional minimization, Computing, 22(1979), pp. 93-100.

[4] R.P. Brent, Algorithms for Minimization without Derivatives, Prentice-Hall, Englewood Cliffs, N.J., 1973.

[5] W.C. Davidon, Conic approximations and collinear scalings for optimizers, SIAM J. Numer. Anal., 17(1980), pp. 268-281.

[6] P.J. Davis, Interpolation and Approximation, Blaisdell Publishing Company, a division of Ginn and Company, Waltham, Mass., 1963.

[7] E. Isaacson and H.B. Keller, Analysis of Numerical Methods, Wiley, New York, 1966.

[8] P. Jarratt and D. Nudds, The use of rational functions in the iterative solution of equations on a digital computer, Computer Journal, 8(1965), pp. 62-65.

[9] P. Jarratt, A rational iteration function for solving equations, Computer Journal, 9(1966), pp. 304-307.

[10] D.G. Luenberger, Introduction to Linear and Nonlinear Programming, Addison-Wesley, Reading, Mass., 1973.

[11] J.M. Ortega and W.C. Rheinboldt, Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York, 1970.

[12] A.M. Ostrowski, La developpment de Taylor de fonction inverse, C.R. Acad. Sci. Paris, 255(1962), pp. 238-240.

[13] A.M. Ostrowski, Solution of Equations and Systems of Equations, 2nd ed., Academic Press, New York, 1966.

[14] A. Ralston, On differentiating error terms, American Mathematical Monthly, 70(1963), pp. 187-188.

[15] A. Ralston, A First Course in Numerical Analysis, McGraw Hill, New York, 1965.

[16] S.M. Robinson, Quadratic interpolation is risky, SIAM J. Numer. Anal., 16(1979), pp. 377-379.

[17] A Tamir, A one-dimensional search based on interpolating polynomials using function values only, Management Science, 22(1976), pp. 576-586.

[18] A Tamir, Rates of convergence of a one-dimensional search based on interpolating polynomials, Journal Opt. Theory Appl., 27(1979), pp. 187-203.

[19] J.F. Traub, Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, N.J., 1964.

[20] M.A.H. Wright, Numerical methods for nonlinearly constrained optimization, Research Report STAN-CS-76-566, Stanford Univ., Stanford, CA, 1976.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CCS 409	2. GOVT ACCESSION NO. AD-A106680	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NONPOLYNOMIAL AND INVERSE INTERPOLATION FOR LINE SEARCH: SYNTHESIS AND CONVERGENCE RATES		5. TYPE OF REPORT & PERIOD COVERED Research
		6. PERFORMING ORG. REPORT NUMBER CCS 408 9
7. AUTHOR(s) J. Barzilai, A. Ben-Tal		8. CONTRACT OR GRANT NUMBER(s) N00014-75-C-0569 N00014-80-C-0242
9. PERFORMING ORGANIZATION NAME AND ADDRESS Center for Cybernetic Studies, UT Austin Austin, Texas 78712		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research (Code 434) Washington, D.C.		12. REPORT DATE July 1981
		13. NUMBER OF PAGES 33
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

This document has been approved for public release and sale; its distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Nonpolynomial interpolation, inverse interpolation, convergence rates, line search, root finding, mathematical programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The rate of convergence of line search algorithms based on general interpolating functions is derived, and is shown to be independent of the particular interpolating function used. This result holds for the root finding problem $f(x) = 0$ as well. We show how inverse interpolation can be used in conjunction with the line search problem, and derive its rate of convergence. Our analysis suggests that one-point line search algorithms (in particular Newton's method) are inefficient in a sense. Two-point (cont'd)

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

20. (cont'd) algorithms using rational interpolating functions are recommended.

E

ED

8