

NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

EFFICIENT METHODS FOR FINDING ZEROS OF FUNCTIONS
WHOSE DERIVATIVES ARE EASY TO EVALUATE

Richard Brent
Computer Centre
Australian National University
Canberra, Australia

December 1974

This work was supported in part by the Office of Naval Research under Contract N0014-67-0314-0010, NR 044-422 and by the National Science Foundation under Grant GJ32111.

ABSTRACT

Some multipoint iterative methods without memory for approximating simple zeros of functions of one variable are described. For $m > 0$, $n \geq 0$, and $m+1 \geq k > 0$, there exist methods which use one evaluation of $f, f', \dots, f^{(m)}$ followed by n evaluations of $f^{(k)}$ for each iteration, and have order of convergence $m+2n+1$. In particular, there are methods of order $2(n+1)$ which use one function evaluation and $n+1$ derivative evaluations per iteration. These methods naturally generalize the known cases $n = 0$ (Newton's method) and $n = 1$ (Jarratt's fourth-order method), and are useful if derivative evaluations are less expensive than function evaluations. Explicit, nonlinear, Runge-Kutta methods for the solution of a special class of ordinary differential equations may be derived from the methods for finding zeros of functions. Numerical examples and some Fortran subroutines are given.

1. INTRODUCTION

Traub (1964) and Jarratt (1969) have considered multipoint iterative methods for approximating a simple zero of a function f which is more difficult to evaluate than its derivative f' . (Examples of such functions are given in Sections 8 and 9.) Jarratt improved Traub's results by giving a fourth-order method which uses one evaluation of f and two of f' per iteration and is "without memory" in the sense of Traub (1964). This is rather surprising, for the obvious method [evaluating $f(x_0)$ and $f'(x_0)$, computing the Newton-Raphson approximation $x'_0 = x_0 - f(x_0)/f'(x_0)$, evaluating $f'(x'_0)$, and taking x_1 as the appropriate zero of the quadratic $Q(x)$ which satisfies $Q(x_0) = f(x_0)$, $Q'(x_0) = f'(x_0)$, and $Q'(x'_0) = f'(x'_0)$] has order three rather than four. Jarratt showed that order four is attainable if we evaluate $f'((x_0+2x'_0)/3)$ instead of $f'(x'_0)$. For methods with two function evaluations and one derivative evaluation per iteration the results are less surprising: Ostrowski (1966) showed that order four is attainable by evaluating $f(x_0)$, $f'(x_0)$, and $f(x'_0)$ (see also Traub (1964, Sec. 8.5)).

In this paper we show that Jarratt's result can be generalized in a natural way: for all $\nu > 0$, there are multipoint iterative methods (without memory) which use one function evaluation and ν derivative evaluations per iteration, and have order 2ν . Jarratt's method is an example with $\nu = 2$, but the methods with $\nu > 2$ appear to be new.

Jarratt's result can also be extended to methods using higher derivatives. Our main result (Theorem 4.1) is that, for all $m > 0$, $n \geq 0$ and $m+1 \geq k > 0$, there are methods of order $m+2n+1$ which use, for each iteration, one evaluation of f , f' , \dots , $f^{(m)}$ (at the same point) followed by n evaluations of $f^{(k)}$ (at distinct points). These methods are described in Section 2, and the order of convergence is established in Section 4. The theoretical efficiencies of the different methods are compared in Section 5.

Special cases of practical interest ($k \leq 3$, $n \leq 3$) are given explicitly in Section 6, and some Fortran subroutines for methods of order four, six and eight (with $k = m = 1$ and $n = 1, 2$ and 3) are given in the Appendix. Numerical results for these methods are summarized in Section 9, and some possible extensions are mentioned in Section 7.

Since our methods are useful for functions whose derivatives can be evaluated easily, it is not surprising that they are related to certain Runge-Kutta methods for solving a restricted class of ordinary differential equations. This is discussed in Section 8, and numerical comparisons with well-known Runge-Kutta methods are included in Section 9.

The theory of most of our zero-finding methods depends on the theory of orthogonal and "almost orthogonal" polynomials. We assume several well-known properties of orthogonal polynomials, but some nonstandard results which we need later are proved in Section 3.

2. THE METHODS

Let k, m and n be integers satisfying $m > 0, n \geq 0, m+1 \geq k > 0$, and let f be a sufficiently smooth function of one real variable with a simple zero ζ . (It is sufficient for f to have a continuous $(m+2n+1)$ -th derivative in a neighbourhood of ζ .) We describe two classes of methods for improving an initial approximation x_0 to ζ , using evaluations of $f(x_0), f'(x_0), \dots, f^{(m)}(x_0)$ and $f^{(k)}(y_1), \dots, f^{(k)}(y_n)$, where the points y_1, \dots, y_n will be specified below. After generating an improved approximation x_1, x_0 may be replaced by x_1 and a new approximation x_2 generated in the same way, etc. Since all the methods considered are stationary and without memory, it is sufficient to describe how x_1 is generated from x_0 .

Methods in the first class, $C'(k, m, n)$, have order $\min(m+2n+1, 2m+n+1)$. The second class, $C(k, m, n)$, is a modification of $C'(k, m, n)$, and methods in $C(k, m, n)$ have order $m+2n+1$. Arbitrary methods in $C(k, m, n)$ and $C'(k, m, n)$ are denoted $M(k, m, n)$ and $M'(k, m, n)$ respectively.

We make frequent use of the "0" notation. Unless otherwise specified, an equation such as $x = y + 0(\delta)$ means that there is a constant K (possibly depending on k, m, n, f , etc., but independent of δ) such that $|x - y| \leq K\delta$ for all sufficiently small positive δ .

For our purposes it is sufficient to say that a method has *order of convergence* $\rho > 1$ if $x_1 - \zeta = O(|x_0 - \zeta|^\rho)$ for starting values x_0 sufficiently close to the simple zero ζ . (More general definitions are given in Brent (1973b) and Ortega and Rheinboldt (1970).) The order ρ is an integer for all methods considered below.

For $p+1 > q > 0$, the *Jacobi polynomial* $G_n(p, q, x)$ is the monic polynomial of degree n which is orthogonal to all polynomials of degree less than n with respect to the weight function $(1-x)^{p-q}x^{q-1}$ on the interval $[0, 1]$. (Our notation follows that of Abramowitz and Stegun (1964).) Thus $G_0(p, q, x) = 1$, $G_1(p, q, x) = x - q/(p+1)$, $G_2(p, q, x) = x^2 - 2(q+1)x/(p+3) + q(q+1)/((p+2)(p+3))$, etc. Let $\alpha_1, \dots, \alpha_n$ denote the zeros of $G_n(m+1, m+2-k, x)$ in a fixed but arbitrary order.

The class $C'(k, m, n)$

We say that a method belongs to $C'(k, m, n)$ if an iteration generates a new approximation x_1 to ζ from an old approximation x_0 in the following way:

1. Evaluate $f_0^{(i)} = f^{(i)}(x_0)$ for $i = 0, 1, \dots, m$.
2. If $f_0^{(0)} = 0$ set $x_1 = x_0$ and stop, otherwise set $\delta = |f_0^{(0)}/f_0^{(1)}|$.
3. Let z_1 be an approximate zero of the polynomial

$$p_1(x) = \sum_{i=0}^m (x - x_0)^i f_0^{(i)}/i! ,$$

satisfying the conditions

$$z_1 = x_0 + o(\delta) \quad (2.1)$$

and

$$p_1(z_1) = o(\delta^{m+1}) . \quad (2.2)$$

(We do not specify how z_1 is computed so long as (2.1) and (2.2) hold. One method is to perform $\lceil \log_2(m+1) \rceil - 1$ iterations of Newton's method, starting from the approximate zero $x_0 - f_0^{(0)}/f_0^{(1)}$.)

4. Evaluate $f_i^{(k)} = f^{(k)}(y_i)$, where $y_i = x_0 + \alpha_i(z_1 - x_0)$, for $i = 1, \dots, n$.

5. Let p_{n+1} be the polynomial of degree at most $m+n$, satisfying $p_{n+1}^{(i)}(x_0) = f_0^{(i)}$ for $i = 0, \dots, m$ and $p_{n+1}^{(k)}(y_i) = f_i^{(k)}$ for $i = 1, \dots, n$. Let x_1 be an approximate zero of p_{n+1} , satisfying

$$x_1 = x_0 + o(\delta) \quad (2.3)$$

and

$$p_{n+1}(x_1) = o(\delta^\rho), \quad (2.4)$$

where $\rho = \min(m+2n+1, 2m+n+1)$. (Again, we do not specify

how x_1 is computed, but $\lceil \log_2(\rho/(m+1)) \rceil$ iterations of Newton's method with the starting approximation z_1 would suffice.)

Comments on $C'(k, m, n)$

$p_1(x)$ is the Hermite interpolating polynomial agreeing with $f(x_0), \dots, f^{(m)}(x_0)$. Conditions (2.1) and (2.3) ensure that z_1 and x_1 are approximations to the correct zeros of p_1 and p_{n+1} respectively. If x_0 is sufficiently close to ζ then Newton's method gives the approximation $x'_0 = x_0 - f_0^{(0)}/f_0^{(1)}$ satisfying $x'_0 = \zeta + O((x_0 - \zeta)^2)$, but $|x'_0 - x_0| = \delta$, so

$$\delta/2 \leq |x_0 - \zeta| \leq 2\delta \quad (2.5)$$

for δ sufficiently small. Thus, any terms $O(\delta^i)$ are equivalent to terms $O(|x_0 - \zeta|^i)$. We use expressions involving δ rather than ζ in the definition of $C'(k, m, n)$ since ζ is, of course, unknown.

Since ζ is a simple zero of f , (2.2) is equivalent to $z_1 = \zeta_1 + O(\delta^{m+1})$, where ζ_1 is the zero of p_1 closest to ζ . By the theory of Hermite interpolation (Traub (1964)), $\zeta_1 = \zeta + O(\delta^{m+1})$, so (2.2) is equivalent to

$$z_1 = \zeta + O(\delta^{m+1}). \quad (2.6)$$

Theorem 2.1

If x_0 is sufficiently close to the simple zero ζ , then steps 1 to 5 of $M'(k, m, n)$ are well-defined and

$$x_1 = \zeta + O(|x_0 - \zeta|^\rho), \quad (2.7)$$

where

$$\rho = \min(m+2n+1, 2m+n+1). \quad (2.8)$$

Sketch of proof

We shall not give the proof of Theorem 2.1 in detail since it is similar to (but easier than) the proof of Theorem 4.1 below. We shall, however, indicate how the order ρ given by (2.8) arises.

From the definition of p_{n+1} (step 5 above) and the Taylor series expansion of f about x_0 , it is easy to show that

$$p_{n+1}(x) = f(x) + O(\delta^{m+n+1}) \quad (2.9)$$

and

$$p'_{n+1}(x) = f'(x) + O(\delta^{m+n}) \quad (2.10)$$

for all x in the region of interest (say $|x - x_0| \leq 4\delta$ in view of (2.5)). Using properties of the Jacobi polynomial

$G_n(m+1, m+2-k, x)$, as in Lemma 4.3 below, there is a kind of "superconvergence" phenomenon (de Boor and Swartz (1973), Osborne (1974)) at z_1 :

$$p_{n+1}(z_1) = f(z_1) + o(\delta^{m+2n+1}) \quad (2.11)$$

(in contrast to (2.9)).

Let x'_1 be the exact zero of p_{n+1} near x_1 . Using (2.6), (2.9) and (2.10), we have

$$|x'_1 - z_1| \leq |x'_1 - \zeta| + |z_1 - \zeta| = o(\delta^{m+1}). \quad (2.12)$$

Now

$$\begin{aligned} |f(x'_1)| &= |p_{n+1}(x'_1) - f(x'_1)| \\ &\leq |p_{n+1}(z_1) - f(z_1)| + |p'_{n+1}(\xi) - f'(\xi)| \cdot |x'_1 - z_1| \end{aligned} \quad (2.13)$$

for some ξ between x'_1 and z_1 . Using (2.10), (2.11) and (2.12), this gives

$$f(x'_1) = o(\delta^{m+2n+1}) + o(\delta^{2m+n+1}) = o(\delta^p)$$

and thus (as $f'(x)$ is bounded away from zero near ζ)

$$x'_1 = \zeta + o(\delta^p).$$

Since (2.3) and (2.4) ensure that $x_1 = x'_1 + o(\delta^p)$, the result

(2.7) follows.

The class $C(k, m, n)$

Methods in the class $C'(k, m, n)$ are unsatisfactory if $m < n$ since their order is $2m+n+1$, less than the order $m+2n+1$ which might be expected from (2.11). The higher order is attainable if z_1 is updated and the zeros $\alpha_1, \dots, \alpha_n$ are perturbed suitably after each evaluation of $f^{(k)}(y_j)$, so the second term in (2.13) is reduced to $O(\delta^{m+2n+1})$ or less, without substantially increasing the first term. We say that a method belongs to $C(k, m, n)$ if an iteration generates a new approximation x_1 to ζ from an old approximation x_0 in the following way:

1. Evaluate $f_0^{(i)} = f^{(i)}(x_0)$ for $i = 0, 1, \dots, m$.
2. If $f_0^{(0)} = 0$ set $x_1 = x_0$ and stop, otherwise set $\delta = |f_0^{(0)}/f_0^{(1)}|$.
3. For $i = 1, 2, \dots, n$ do steps 4 to 7.
4. Let p_i be the polynomial of degree at most $m+i-1$, satisfying $p_i^{(j)}(x_0) = f_0^{(j)}$ for $j = 0, 1, \dots, m$ and $p_i^{(k)}(y_j) = f_j^{(k)}$ for $j = 1, \dots, i-1$. Let z_i be an approximate zero of p_i , satisfying the conditions

$$z_i = x_0 + O(\delta) \quad (2.14)$$

and

$$p_i(z_i) = O(\delta^{m+i}). \quad (2.15)$$

(For example, z_i could be computed by Newton's method, using $\lceil \log_2(m+1) \rceil - 1$ iterations from the approximation $x_0 - f_0^{(0)}/f_0^{(1)}$ if $i = 1$, and one iteration from the approximation z_{i-1} if $i > 1$.)

5. Compute

$$\alpha_{i,j} = \alpha_{i-1,j} (z_{i-1} - x_0) / (z_i - x_0) \quad (2.16)$$

for $j = 1, \dots, i-1$.

6. Let q_i be the monic polynomial of degree $n+1-i$, satisfying

$$\int_0^1 P(x) q_i(x) x^{m+1-k} (1-x)^{k-1} \prod_{j=1}^{i-1} (x - \alpha_{i,j}) dx = 0 \quad (2.17)$$

for all polynomials P of degree $n-i$. Let $\alpha_{i,i}$ be an approximate zero of q_i , satisfying

$$\alpha_{i,i} = \alpha_i + o(\delta) \quad (2.18)$$

and

$$q_i(\alpha_{i,i}) = o(\delta^{m+i-1}). \quad (2.19)$$

7. Evaluate $f_i^{(k)} = f^{(k)}(y_i)$, where

$$y_i = x_0 + \alpha_{i,i} (z_i - x_0). \quad (2.20)$$

8. Let p_{n+1} be the polynomial of degree at most $m+n$, satisfying $p_{n+1}^{(i)}(x_0) = f_0^{(i)}$ for $i = 0, \dots, m$ and $p_{n+1}^{(k)}(y_i) = f_i^{(k)}$ for $i = 1, \dots, n$. Let x_1 be an approximate zero of p_{n+1} , satisfying (2.3) and

$$p_{n+1}(x_1) = O(\delta^{m+2n+1}). \quad (2.21)$$

(For example, x_1 could be computed by one iteration of Newton's method from the approximation z_n , if $n > 0$.)

Comments on $C(k, m, n)$

It is easy to see that the class $C(k, m, n)$ is the same as $C'(k, m, n)$ if and only if $n = 0$ or 1 .

The existence and uniqueness of q_i (for x_0 sufficiently close to ζ) is shown constructively below. In the cases of practical interest explicit formulae can be given for $\alpha_{i,j}$, so that there is no need to construct q_i (see (6.5), (6.9), and (6.10) for some examples).

From (2.16) and (2.20),

$$y_i = x_0 + \alpha_{j,i}(z_j - x_0) \quad (2.22)$$

for $j = i, i+1, \dots, n$, so the effect of replacing the approximation z_i to ζ by the approximation z_j is the same as if we had used a slightly perturbed node $\alpha_{j,i}$ in place of $\alpha_{i,i}$.

Before proving that a method in $C(k, m, n)$ has order $m+2n+1$ (Theorem 4.1), we need some results on orthogonal polynomials. The next two sections may be omitted without loss of continuity.

3. SOME RESULTS ON ORTHOGONAL POLYNOMIALS

Theorem 3.1 is a generalization of the well-known results that zeros of polynomials orthogonal with respect to a positive weight function interlace, and that the matrix A given by (3.1) is nonsingular (by unisolvency, this is true for any distinct $\alpha_1, \dots, \alpha_n$, not necessarily zeros of P_n).

Theorem 3.1

Let P_0, \dots, P_n be polynomials of degree $0, \dots, n$, orthogonal with respect to the weight function $w(x)$ on $[a, b]$. If $w(x)$ is positive a.e. on $[a, b]$, and $\alpha_1, \dots, \alpha_n$ are the zeros of P_n in any order, then all leading principal minors of

$$A = \begin{bmatrix} P_{n-1}(\alpha_1) & \dots & P_{n-1}(\alpha_n) \\ \vdots & & \vdots \\ P_0(\alpha_1) & \dots & P_0(\alpha_n) \end{bmatrix} \quad (3.1)$$

are nonsingular.

Proof

Since w is positive, there is a three-term recurrence relation of the form

$$P_j(x) = (A_j x + B_j)P_{j+1}(x) + C_j P_{j+2}(x) \quad (3.2)$$

for $j = 0, 1, \dots$, and $A_j \neq 0$ (see Isaacson and Keller (1966, Ch. 5) or Szegő (1959, Thm. 3.2.1)). Let x be a zero of P_n . Applying (3.2) with $j = n-2$ gives

$$P_{n-2}(x) = (A_{n-2}x + B_{n-2})P_{n-1}(x). \quad (3.3)$$

Applying (3.2) with $j = n-3$ and using (3.3) now gives

$$P_{n-3}(x) = [(A_{n-3}x + B_{n-3})(A_{n-2}x + B_{n-2}) + C_{n-3}]P_{n-1}(x).$$

Proceeding in this way, we find that

$$P_{n-k-1}(x) = \phi_{n,k}(x)P_{n-1}(x) \quad (3.4)$$

for $k = 0, 1, \dots, n-1$, where $\phi_{n,k}(x)$ is a polynomial of degree k in x . In fact, it is easily shown that $\phi_{n,0}(x) = 1$, $\phi_{n,1}(x) = A_{n-2}x + B_{n-2}$, and (for $k > 1$)

$$\phi_{n,k}(x) = (A_{n-k-1}x + B_{n-k-1})\phi_{n,k-1}(x) + C_{n-k-1}\phi_{n,k-2}(x),$$

so $\phi_{n,k}(x)$ has leading term $A_{n-2} \dots A_{n-k-1}x^k$.

Suppose $0 < s \leq n$, and let A_s be the leading principal minor of order s of A . Using (3.4) with $x = \alpha_1, \dots, \alpha_s$, we have

$$\det(A_s) = \left[\prod_{i=1}^s P_{n-1}(\alpha_i) \right] \det \begin{bmatrix} \phi_{n,0}(\alpha_1) & \dots & \phi_{n,0}(\alpha_s) \\ \vdots & & \vdots \\ \phi_{n,s-1}(\alpha_1) & \dots & \phi_{n,s-1}(\alpha_s) \end{bmatrix}.$$

By performing row operations and using the observation above on the leading term of $\phi_{n,k}$, this gives

$$\det(A_s) = \left(\prod_{i=1}^s P_{n-1}(\alpha_i) \right) \left(\prod_{i=2}^s A_{n-i}^{s+1-i} \right) \det(V_s), \quad (3.5)$$

where

$$V_s = \begin{bmatrix} 1 & \dots & 1 \\ \alpha_1 & & \alpha_s \\ \vdots & & \vdots \\ \alpha_1^{s-1} & \dots & \alpha_s^{s-1} \end{bmatrix} \quad (3.6)$$

is a Vandermonde matrix. Since the α_i are distinct and not zeros of P_{n-1} , the result follows from (3.5).

The idea of the following theorem is most easily seen by considering the special case $\beta_i = \alpha_i$ ($i = 1, \dots, n$) first. Then $d_0 = \dots = d_{n-1} = 0$, (3.11) says that $\gamma_1, \dots, \gamma_s$ are slight perturbations of $\alpha_1, \dots, \alpha_s$, and the theorem states that there exist slight perturbations $\gamma_{s+1}, \dots, \gamma_n$ of $\alpha_{s+1}, \dots, \alpha_n$ such that $\prod_{j=1}^n (x - \gamma_j)$ is exactly orthogonal to polynomials of degree less than $n-s$, and approximately orthogonal to polynomials of degree $n-s, \dots, n-1$. We state the more complicated result (with β_1, \dots, β_n slight perturbations of $\alpha_1, \dots, \alpha_n$) because in Section 4 we shall apply Theorem 3.2 several times, and the $\gamma_1, \dots, \gamma_n$ of one application will be (close to) the β_1, \dots, β_n of the next application. The only related results which we know are those of Micchelli and Rivlin (1973).

Theorem 3.2

Let P_0, \dots, P_n be orthonormal polynomials (of degree

$0, \dots, n$) with respect to the positive weight function w on $[a, b]$, and let $\alpha_1, \dots, \alpha_n$ be the zeros of P_n in any order. Suppose $0 < s < n$, δ sufficiently small, and β_1, \dots, β_n satisfy

$$|\beta_i - \alpha_j| \leq \delta \quad (3.7)$$

for $i = 1, \dots, n$. Suppose that there is a positive constant c_1 and numbers $\delta_1, \dots, \delta_s$ such that

$$\delta \geq \delta_1 \geq \dots \geq \delta_s \geq 0, \quad (3.8)$$

$$|d_i| \leq \begin{cases} \delta_s & \text{for } 0 \leq i < n-s, \\ \delta_{n-i} & \text{for } n-s \leq i < n, \end{cases} \quad (3.9)$$

and

$$d_n = 1,$$

where

$$d_i = c_1 \int_a^b P_i(x) \left[\prod_{j=1}^n (x - \beta_j) \right] w(x) dx. \quad (3.10)$$

Finally, suppose $\gamma_1, \dots, \gamma_s$ satisfy

$$|\gamma_i - \beta_i| \leq \delta_s \quad (3.11)$$

for $i = 1, \dots, s$. Then there is a positive constant c_2 and unique $\gamma_{s+1}, \dots, \gamma_n$ such that

$$\gamma_i = \beta_i + o(\delta_s) \quad (3.12)$$

for $i = s+1, \dots, n$ and

$$e_i = \begin{cases} 0 & \text{for } 0 \leq i < n-s, \\ o(\delta_{n-i}) & \text{for } n-s \leq i < n, \\ 1 & \text{for } i = n, \end{cases} \quad (3.13)$$

where

$$e_i = c_2 \int_a^b P_i(x) \left[\prod_{j=1}^n (x - \gamma_j) \right] w(x) dx. \quad (3.14)$$

Proof

Let

$$\gamma'_i = \begin{cases} \gamma_i & \text{if } i \leq s \\ \beta_i & \text{if } i > s \end{cases}, \quad q_1(x) = c_1 \prod_{i=1}^n (x - \gamma'_i),$$

and $q_2(x) = q_1(x) + \sum_{i=0}^{n-1} \mu_i P_i(x)$, where the constants $\mu_i = o(\delta_s)$

will be determined below. From (3.11),

$$q_1(x) = c_1 \prod_{i=1}^n (x - \beta_i) + o(\delta_s) \quad \text{for all } x \text{ in } [a, b],$$

so from (3.8) to (3.10) we have

$$f_i = \begin{cases} 0(\delta_s) & \text{for } 0 \leq i < n-s, \\ 0(\delta_{n-i}) & \text{for } n-s \leq i < n, \\ 1 + 0(\delta_s) & \text{for } i = n, \end{cases} \quad (3.15)$$

where

$$f_i = \int_a^b P_i(x) q_1(x) w(x) dx. \quad (3.16)$$

Let

$$g_i = \int_a^b P_i(x) q_2(x) w(x) dx. \quad (3.17)$$

Since the P_j are orthonormal, (3.16) and (3.17) give

$$g_i = f_i + \mu_i \quad (3.18)$$

for $i = 0, \dots, n-1$. Set $\mu_i = -f_i$ for $i = 0, \dots, n-s-1$.

From Wilkinson (1963, Sec. 2.7), the zeros γ_i^* of q_2 are analytic functions of $\mu_{n-s}, \dots, \mu_{n-1}$, given by

$$\gamma_i^* = \gamma_i' + \sum_{j=0}^{n-s-1} f_j P_j(\gamma_i') / q_1'(\gamma_i') - \sum_{j=n-s}^{n-1} \mu_j P_j(\gamma_i') / q_1'(\gamma_i') + 0(\delta_s^2) \quad (3.19)$$

($i = 1, \dots, n$), provided $\mu_j = 0(\delta_s)$ for $j = n-s, \dots, n-1$.

By (3.7), (3.8), (3.11) and Theorem 3.1, the matrix

$$A = \begin{bmatrix} P_{n-s}(\gamma_1') & \dots & P_{n-s}(\gamma_s') \\ \vdots & & \vdots \\ P_{n-1}(\gamma_1') & \dots & P_{n-1}(\gamma_s') \end{bmatrix}$$

is nonsingular for δ sufficiently small, so there exist $\mu_{n-s}, \dots, \mu_{n-1}$ (all $O(\delta_s)$) such that $\gamma_i'' = \gamma_i'$ for $i = 1, \dots, s$. Hence (by definition of γ_i'), $\gamma_i = \gamma_i''$ for $i = 1, \dots, s$. Take $\gamma_i = \gamma_i''$ for $i = s+1, \dots, n$ also, so

$$q_2(x) = c_1 \prod_{i=1}^n (x - \gamma_i). \quad (3.20)$$

By (3.19) and the construction of μ_j and γ_j , (3.12) holds. From (3.18) and the choice of $\mu_0, \dots, \mu_{n-s-1}$, $g_i = 0$ for $i < n-s$. From (3.8), (3.15) and (3.18), $g_i = O(\delta_{n-i})$ for $n-s \leq i < n$, and $g_n = 1 + O(\delta_s)$. Taking $c_2 = c_1/g_n$ and collecting these results, existence follows. Uniqueness (subject to (3.12)) follows from (3.19) and the nonsingularity of A .

The following corollary gives an extension to "almost orthogonal" polynomials of a classical result for orthogonal polynomials (the case $\delta = 0$, $\gamma_i = \alpha_i$). The proof follows that of the classical result up to equation (3.23), and then uses Theorem 3.2.

Corollary 3.1

Under the conditions of Theorem 3.2, there exist weights w_1, \dots, w_n such that

$$\sum_{i=1}^n w_i \gamma_i^j = \int_a^b x^j w(x) dx + r_j, \quad (3.21)$$

where

$$r_j = \begin{cases} 0 & \text{for } j = 0, 1, \dots, 2n-(s+1), \\ 0(\delta_{2n-j}) & \text{for } j = 2n-s, \dots, 2n-1. \end{cases}$$

Proof

We may assume δ sufficiently small that $\gamma_1, \dots, \gamma_n$ are distinct (in view of (3.7), (3.11) and (3.12)). For any function f we have the Lagrange interpolation formula

$$f(x) = \sum_{i=1}^n L_i(x) f(\gamma_i) + \left[\prod_{i=1}^n (x - \gamma_i) \right] f[\gamma_1, \dots, \gamma_n, x], \quad (3.22)$$

where

$$L_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - \gamma_j}{\gamma_i - \gamma_j}.$$

(See, for example, Isaacson and Keller (1966, Sec. 6.1).)

Let

$$w_i = \int_a^b L_i(x) w(x) dx,$$

and integrate both sides of (3.22), with $f(x) = x^j$. Thus (3.21) holds if

$$r_j = - \int_a^b \left[\prod_{i=1}^n (x - \gamma_i) \right] f[\gamma_1, \dots, \gamma_n, x] w(x) dx. \quad (3.23)$$

Since $f[\gamma_1, \dots, \gamma_n, x]$ vanishes for $j < n$, and is a polynomial of degree $j-n$ for $j \geq n$, the result follows by expanding this polynomial as a sum $\sum_{i=0}^{j-n} v_i P_i(x)$ (where $v_i = O(1)$) and applying Theorem 3.2.

4. ORDER OF CONVERGENCE

Before proving the main result (Theorem 4.1) we need some lemmas. The notation of Section 2 is assumed.

Lemma 4.1

If x_0 is sufficiently close to the simple zero ζ , then $M(k, m, n)$ is defined, and if $x_0 \neq \zeta$ then (4.1) to (4.5) hold:

$$\delta/2 \leq |x_0 - \zeta| \leq 2\delta; \quad (4.1)$$

$$z_i = \zeta + O(\delta^{m+i}) \quad (4.2)$$

for $i = 1, \dots, n$; q_i exists, is unique, and has a zero

$$\alpha_{i,j}^1 = \alpha_j + O(\delta) \quad (4.3)$$

for $i = 1, \dots, n$;

$$\alpha_{i,j} = \alpha_{i-1,j} + O(\delta^{m+i-2}) \quad (4.4)$$

for $0 < j < i \leq n$; and

$$x_1 = \zeta + O(\delta^{m+n+1}). \quad (4.5)$$

Proof

The exceptional case $x_0 = \zeta$ is covered by step 2 of $M(k, m, n)$, so we may assume $x_0 \neq \zeta$. The inequality (4.1)

follows in the same way as (2.5).

(4.2) to (4.4) clearly hold for $i = 1$. We shall assume that they hold for $i < t$ (where $1 < t \leq n$) and prove that they hold for $i = t$.

p_t is a well-defined polynomial of degree at most $m+t-1$, and the theory of polynomial interpolation (Traub (1964)) shows that there is a zero $\zeta_t = \zeta + O(\delta^{m+t})$ of p_t . Conditions (2.14) and (2.15) ensure that $z_t = \zeta_t + O(\delta^{m+t})$, so (4.2) with $i = t$ follows.

From (2.16),

$$\alpha_{t,j} = \alpha_{t-1,j} \left(1 + \frac{z_{t-1} - z_t}{z_t - x_0} \right)$$

for $0 < j < t$. Since

$$|z_{t-1} - z_t| \leq |z_{t-1} - \zeta| + |z_t - \zeta| = O(\delta^{m+t-1})$$

and (using (4.1))

$$\begin{aligned} |z_t - x_0| &\geq |x_0 - \zeta| - |z_t - \zeta| \\ &\geq \delta/2 - O(\delta^{m+t}) \geq \delta/4 \end{aligned}$$

for δ sufficiently small, (4.4) with $i = t$ follows.

From (4.4) with $i = t, t-1, \dots, j+1$ we have
 $\alpha_{t,j} = \alpha_{j,j} + 0(\delta^{m+j-1})$, but (by the inductive hypothesis)
 $\alpha_{j,j} = \alpha_j + 0(\delta)$, so $\alpha_{t,j} = \alpha_j + 0(\delta)$ for $j = 1, \dots, t-1$.
 Thus, Theorem 3.2 shows that q_t exists, is unique, and has a zero
 $\alpha'_{t,t} = \alpha_t + 0(\delta)$ which $\alpha_{t,t}$ approximates. This completes the
 proof of (4.2) to (4.4), by induction on i . Finally, (4.5)
 follows in the same way as (4.2).

Lemma 4.2

If x_0 is sufficiently close to ζ but $x_0 \neq \zeta$, there
 exist weights w_1, \dots, w_n (all $0(1)$) such that

$$\sum_{i=1}^n w_i \alpha_{n,i}^j = \frac{(j+m+1-k)!(k-1)!}{(j+m+1)!} + 0(\delta^{2n-j}) \quad (4.6)$$

for $j = 0, 1, \dots, 2n-1$.

Proof

The first time step 6 of $M(k, m, n)$ is performed we
 have $i = 1$, $q_1(x) = G_n(m+1, m+2-k, x)$ and, from (2.18) and
 (2.19), $\alpha_{1,1} = \alpha_1 + 0(\delta^m)$. For subsequent executions of step
 6 we have $i > 1$ and, from (2.22) and Lemma 4.1, it is easy to
 show (by induction on i) that Theorem 3.2 is applicable
 (with $s = i-1$, $\delta_j = 0(\delta^{m+j-1})$ for $j = 1, \dots, i-1$,
 $w(x) = (1-x)^{k-1} x^{m+1-k}$, etc.). After the n -th execution of
 step 6, Corollary 3.1 (with $s = n-1$) shows that there exist
 weights w_i such that

$$\sum_{i=1}^n w_i \alpha_{n,i}^j = \int_0^1 x^{j+m+1-k} (1-x)^{k-1} dx$$

$$= \begin{cases} 0 & \text{if } 0 \leq j \leq n, \\ 0(\delta^{m+2n-j-1}) & \text{if } n < j < 2n \end{cases} \quad (4.7)$$

Since $m+2n-j-1 \geq 2n-j$ and the integral on the left side of (4.7) is equal to $(j+m+1-k)!(k-1)!/(j+m+1)!$, the result follows. (In fact (4.7) is stronger than (4.6), but (4.6) is sufficient for later applications.)

Lemma 4.3

Suppose K constant, $P(x)$ a polynomial of degree at most $m+2n$ with coefficients bounded by K and satisfying $P(0) = \dots = P^{(m)}(0) = 0$ and $P^{(k)}(\beta_1 \epsilon) = \dots = P^{(k)}(\beta_n \epsilon) = 0$, where

$$\sum_{i=1}^n w_i \beta_i^j = (j+m+1-k)!(k-1)!/(j+m+1)! + O(\epsilon^{2n-j}) \quad (4.8)$$

for $j = 0, 1, \dots, 2n-1$. Then $P(\epsilon) = O(\epsilon^{m+2n+1})$.

Proof

Let $P(x) = \sum_{j=0}^{2n-1} a_j x^{j+m+1}$, so $P^{(k)}(\beta_i \epsilon) = 0$ gives

$$\sum_{j=0}^{2n-1} a_j \frac{(j+m+1)!}{(j+m+1-k)!} (\beta_i \epsilon)^{j+m+1-k} = 0. \quad (4.9)$$

Multiplying each side of (4.9) by $w_i \beta_i^{k-m-1} \epsilon^k / (k-1)!$ and

summing over $i = 1, \dots, n$ gives

$$\sum_{j=0}^{2n-1} \left(a_j \epsilon^{j+m+1} \sum_{i=1}^n \frac{w_i \beta_i^j (j+m+1)!}{(j+m+1-k)! (k-1)!} \right) = 0.$$

Thus, the result follows from (4.8).

Lemma 4.4

If $n > 0$, x_0 is sufficiently close to the simple zero ζ of f , and $M(k, m, n)$ is applied, then (assuming $x_0 \neq \zeta$)

$$f(z_n) = p_{n+1}(z_n) + o(\delta^{m+2n+1}) \quad (4.10)$$

and

$$\sup_{\xi \in [x_0 - 4\delta, x_0 + 4\delta]} |f'(\xi) - p'_{n+1}(\xi)| = o(\delta^{m+n}). \quad (4.11)$$

Proof

Let $f(x) = f_1(x) + f_2(x)$, where

$$f_1(x) = \sum_{j=0}^{m+2n} (x-x_0)^j f^{(j)}(x_0)/j!. \quad (4.12)$$

For $i = 1, 2$, let $r_i(x)$ be the polynomial of degree at most $m+n$ satisfying $r_i^{(j)}(x_0) = f_i^{(j)}(x_0)$ for $j = 0, \dots, m$ and $r_i^{(k)}(y_j) = f_i^{(k)}(y_j)$ for $j = 1, \dots, n$, where y_j is defined by (2.20). Thus, from the definition of p_{n+1} (step 8 of $M(k, m, n)$)

$$p_{n+1}(x) = r_1(x) + r_2(x). \quad (4.13)$$

Since $P(x) = f_1(x_0+x) - r_1(x_0+x)$ is a polynomial of degree at most $m+2n$ in x , and the conditions of Lemma 4.3 are satisfied with $\varepsilon = z_n - x_0 = O(\delta)$, $\beta_i = \alpha_{n,i}$ ($i = 1, \dots, n$), and w_i given by Lemma 4.2, we have $P(\varepsilon) = O(\varepsilon^{m+2n+1})$, so

$$f_1(z_n) = r_1(z_n) + O(\delta^{m+2n+1}). \quad (4.14)$$

We may write $r_2(x) = \sum_{j=0}^{m+n} a_j(x-x_0)^j$, and from (4.12) and the definitions of f_2 and r_2 we have $a_0 = \dots = a_m = 0$. The coefficients a_{m+1}, \dots, a_{m+n} are determined by the linear equations

$$\sum_{j=m+1}^{m+n} \alpha_{n,i}^{j-k} \varepsilon^j a_j j! / (j-k)! = \varepsilon^k f_2^{(k)}(y_i) \quad (4.15)$$

for $i = 1, \dots, n$. From (4.12) and the definition of f_2 , the right hand side of (4.15) is $O(\delta^{m+2n+1})$, and $\alpha_{n,1}, \dots, \alpha_{n,n}$ are distinct, so $\varepsilon^j a_j = O(\delta^{m+2n+1})$ for $j = m+1, \dots, m+n$. Thus $r_2(z_n) = O(\delta^{m+2n+1})$, and from (4.14) we have

$$\begin{aligned} |f(z_n) - p_{n+1}(z_n)| &\leq |f_1(z_n) - r_1(z_n)| + |r_2(z_n)| + |f_2(z_n)| \\ &= O(\delta^{m+2n+1}), \end{aligned}$$

so (4.10) is established. The proof of (4.11) is straightforward

and does not use the special properties of $\alpha_{n,1}, \dots, \alpha_{n,n}$ (except for their being distinct), so is omitted.

Using Lemmas 4.1 and 4.4 it is easy to prove our main result:

Theorem 4.1

If x_0 is sufficiently close to the simple zero ζ then $M(k, m, n)$ is defined and $x_1 = \zeta + O(|x_0 - \zeta|^{m+2n+1})$.

Proof

Suppose $n > 0$, for otherwise the result follows from Lemma 4.1. From equations (4.2) and (4.5) of Lemma 4.1 we have

$$x_1 = z_n + O(\delta^{m+n}). \quad (4.16)$$

Now

$$\begin{aligned} |f(x_1)| &\leq |p_{n+1}(x_1)| + |f(x_1) - p_{n+1}(x_1)| \\ &\leq |p_{n+1}(x_1)| + |f(z_n) - p_{n+1}(z_n)| \\ &\quad + |f'(\xi) - p'_{n+1}(\xi)| \cdot |x_1 - z_n| \end{aligned}$$

for some ξ between x_1 and z_n . From (2.21), Lemma 4.4 and (4.16), this gives $f(x_1) = O(\delta^{m+2n+1})$. Since ζ is a simple zero of f , we may suppose that $f'(x)$ is bounded away from zero in the region of interest, so $x_1 = \zeta + O(\delta^{m+2n+1})$. The result now follows from (4.1).

5. THEORETICAL COMPARISON OF VARIOUS METHODS

If an iterative method of order $\rho > 1$ requires w units of work per iteration, its *efficiency* is

$$E = \frac{\log \rho}{w}. \quad (5.1)$$

The motivation for this definition (and a more general definition) is given in Brent (1973b). In this section we compare the efficiencies of methods in the classes $C(1, 1, n)$ for $n = 0, 1, \dots$. The extension to methods in $C(k, m, n)$ is straightforward.

Theorem 4.1 shows that a method M_n in $C(1, 1, n)$ has order at least $2(n+1)$, and we shall assume that the order is exactly $2(n+1)$ (this is usually true: see Section 6). If the work for one evaluation of $f(x)$ is $w(f)$ and the overhead for one iteration is $w_0(n)$, then the total work per iteration is

$$w = w(f) + (n+1)w(f') + w_0(n),$$

so (from (5.1)) the efficiency is

$$E_n = \log[2(n+1)]/[w(f) + (n+1)w(f') + w_0(n)]. \quad (5.2)$$

We expect $w_0(n)$ to be an increasing function of n , and it can be estimated for any particular implementation of M_n .

For the sake of simplicity, we shall assume $w_0(n) = 0$ below.

This is a reasonable approximation if n is small and f is difficult to evaluate (see also Kung and Traub (1973b, c)).

With our simplifying assumption, (5.2) gives

$$E_n/E_0 = (1+r)(1 + \log_2(n+1))/(n+1+r), \quad (5.3)$$

where $r = w(f)/w(f')$ and E_0 is the efficiency of Newton's method. Some values of E_n/E_0 are given in Table 5.1.

Table 5.1: E_n/E_0 for various n and $r = w(f)/w(f')$

$r \backslash n$	1	2	3	4	5
0.0	1.000	0.862	0.750	0.664	0.597
0.5	1.200	1.108	1.000	0.906	0.827
1.0	1.333	1.292	1.200	1.107	1.024
2.0	1.500	1.551	1.500	1.424	1.344
5.0	1.714	1.939	2.000	1.993	1.955
10.0	1.833	2.187	2.357	2.436	2.465
∞	2.000	2.585	3.000	3.322	3.585

From (5.3) it follows that M_n is the optimal method (from M_0, M_1, \dots) if

$$\phi(n) < r < \phi(n+1),$$

where

$$\phi(n) = \frac{\log(2n)}{\log(1+1/n)} - n.$$

Thus the optimal values of n are

$$n = \begin{cases} 1 & \text{if } 0 < r < 1.419, \\ 2 & \text{if } 1.419 < r < 3.228, \\ 3 & \text{if } 3.228 < r < 5.319, \\ 4 & \text{if } 5.319 < r < 7.629, \\ 5 & \text{if } 7.629 < r < 10.120, \text{ etc.} \end{cases}$$

In particular, Jarratt's method ($n=1$) is always more efficient than Newton's method ($n=0$), but it is less efficient than one of our sixth-order methods ($n=2$) if $w(f) > 1.419w(f')$, etc.

An obvious conjecture is that methods in $C(k, m, n)$ have the optimal order for multipoint methods without memory using the same number of function and derivative evaluations per iteration. Care has to be taken in phrasing the conjecture to avoid Winograd's encoding trick: one way to eliminate encoding is to suitably restrict the class of allowable iteration functions (see Kung and Traub (1973c)). Kung and Traub (1973a) show that there are multipoint methods without memory which use $v+1$ function evaluations per iteration and have order 2^v , and this order is conjectured to be optimal. Brent, Winograd and Wolfe (1973) have shown that the optimal order is at most 2^{v+1} , even if memory is permitted. In contrast to these results, our conjecture is that methods without memory which use one function evaluation and v derivative evaluations per iteration have order at

most 2ν . Kung and Traub (1973c) have shown that the conjectures are true if $\nu = 1$.

Our methods are only of practical interest for small ν (say $\nu = n+1 \leq 4$), and some such methods are described in detail in the next section. Related methods with memory are given by King (1971, 1972).

6. SOME METHODS OF PRACTICAL INTEREST

In this section we use the notation of Section 2 as far as possible, and temporary variables used in the description below are denoted Δ_i , t_i , v_i , a , b , c , etc. Specific methods in $C(k, m, n)$ (or in $C'(k, m, n)$ if $m \geq n$) are referred to as "method kmna", "method kmnb", etc. If a method has order ρ , the *asymptotic error constant* (if it exists) is

$$K = \lim_{x_0 \rightarrow \zeta} (x_1 - \zeta)/(x_0 - \zeta)^\rho. \quad (6.1)$$

(It is usual to put absolute value signs in (6.1), but we omit them since ρ is an integer for all the methods considered below.) Asymptotic error constants may be obtained, in terms of $f'(\zeta)$, $f''(\zeta)$, etc., by substituting the Taylor series expansion of $f(x)$ about ζ in the definition of the method. They may also be found by other, less tedious, methods which are described in Brent (1974). The error constants for the methods considered below can all be expressed as sums of products of the form

$c \prod_{i=2}^{\rho} \phi_i^{r_i}$, where (from Traub (1964))

$$\sum_{i=2}^{\rho} (i-1)r_i = \rho - 1 \quad (6.2)$$

and

$$\phi_i = \frac{f^{(i)}(\zeta)}{i! f'(\zeta)}. \quad (6.3)$$

6.1 Fourth-order methods

If $m = n = 1$ and $k = 1$ or 2 , the relevant Jacobi polynomial is $G_1(2, 3-k, x) = x - (1-k/3)$, so

$$\alpha_1 = 1 - k/3. \quad (6.4)$$

Some fourth-order methods are summarized in Table 6.1. In all cases α_1 is given by (6.4), $\Delta_1 = -f_0^{(0)}/f_0^{(1)}$, and $f_1^{(k)} = f^{(k)}(x_0 + \alpha_1 \Delta_1)$. In some cases the auxiliary variable

$$\Delta_2 = \begin{cases} 3(f_1^{(1)} - f_0^{(1)}) / (6f_1^{(1)} - 2f_0^{(1)}) & \text{if } k = 1, \\ \Delta_1 f_1^{(2)} / [2(f_0^{(1)} + \Delta_1 f_1^{(2)})] & \text{if } k = 2 \end{cases}$$

is used. The formulae for x_1 and the asymptotic error constants K are given in the table. The only difference between the methods with $k = 1$ is in the approximation used for the zero of the interpolating quadratic

$$p_2(x) = f_0^{(0)} + f_0^{(1)}(x-x_0) + \frac{3(f_1^{(1)} - f_0^{(1)})}{4\Delta_1} (x-x_0)^2.$$

Method 111a is Jarratt's method (Jarratt (1969)), method 111b uses the approximation $x'_0 - p_2(x'_0)/p_2'(x'_0)$ where $x'_0 = x_0 + \Delta_1$, method 111c uses the (better) approximation $x'_0 - p_2(x'_0)/p_2'(x'_0) - \frac{1}{2} p_2^2(x'_0) p_2''(x'_0) / (p_2'(x'_0))^3$, and method 111d solves the quadratic exactly if it has real roots. Similarly, the difference between the methods with $k = 2$ is in the approximation used

for the zero of the interpolating quadratic $f_0^{(0)} + f_0^{(1)}(x-x_0) + \frac{1}{2}f_1^{(2)}(x-x_0)^2$.

Table 6.1: Some fourth-order methods

Method	k	α_1	$x_1 - x_0$	K
111a	1	2/3	$\Delta_1(5+3(f_0^{(1)}/f_1^{(1)})^2)/8$	$\phi_4/9 - \phi_2\phi_3 + 13\phi_2^3/9$
111b	1	2/3	$\Delta_1(1 - \Delta_2)$	$\phi_4/9 - \phi_2\phi_3 + \phi_2^3$
111c	1	2/3	$\Delta_1(1 - \Delta_2(1 + \Delta_2^2))$	$\phi_4/9 - \phi_2\phi_3$
111d	1	2/3	$2\Delta_1/\{1 + [\max(0, 3f_1^{(1)}/f_0^{(1)} - 2)]^{1/2}\}$	$\phi_4/9 - \phi_2\phi_3$
211a	2	1/3	$\Delta_1(1 - \Delta_2)$	$\phi_4/3 - \phi_2\phi_3 + \phi_2^3$
211b	2	1/3	$\Delta_1(1 - \Delta_2(1 + \Delta_2^2))$	$\phi_4/3 - \phi_2\phi_3$

6.2 Sixth-order methods

If $k = m = 1$, $n = 2$, the relevant Jacobi polynomial is $G_2(2, 2, x) = x^2 - 6x/5 + 3/10$, with zeros $(6 \pm \sqrt{6})/10$.

Method 112a

$$\alpha_1 = (6 - \sqrt{6})/10, \Delta_1 = -f_0^{(0)}/f_0^{(1)}, f_1^{(1)} = f'(x_0 + \alpha_1\Delta_1),$$

$$\Delta_2 = \frac{1}{2}\Delta_1(f_1^{(1)} + (2\alpha_1 - 1)f_0^{(1)}) / (f_1^{(1)} + (\alpha_1 - 1)f_0^{(1)}), \alpha_{2,1} = \alpha_1\Delta_1/\Delta_2,$$

$$\alpha_{2,2} = (3 - 4\alpha_{2,1}) / (4 - 6\alpha_{2,1}), \quad (6.5)$$

$$f_2^{(1)} = f'(x_0 + \alpha_{2,2}\Delta_2),$$

$$t_1 = (f_1^{(1)} - f_0^{(1)})/(\alpha_1 \Delta_1), \quad t_2 = (f_2^{(1)} - f_0^{(1)})/(\alpha_{2,2} \Delta_2),$$

$$v_1 = (\alpha_{2,2} t_1 - \alpha_{2,1} t_2)/(\alpha_{2,2} - \alpha_{2,1}), \quad v_2 = (t_2 - t_1)/(\alpha_{2,2} - \alpha_{2,1}),$$

$$\Delta_3 = f_0^{(0)} + f_0^{(1)} \Delta_2 + (3v_1 + 2v_2) \Delta_2^2/6,$$

$$\Delta_4 = f_0^{(1)} + \Delta_2(v_1 + v_2),$$

and

$$x_1 = x_0 + \Delta_2 - \Delta_3/\Delta_4 - \frac{1}{2} \Delta_3^2 v_1 / \Delta_4^3. \quad (6.6)$$

The error constant is

$$K = \phi_6/100 + (1 - 5\alpha_1)\phi_2\phi_5/10 + (3\alpha_1 - 2)\phi_3\phi_4/5. \quad (6.7)$$

Method 112b

This method is the same as 112a except $\alpha_1 = (6 + \sqrt{6})/10$.

The error constant is still given by (6.7).

Comments on methods 112a and 112b

Most of the equations above are obtained in a straightforward manner from the general description in Section 2. We should explain equation (6.5). From (2.17) we need a linear

polynomial $q_2(x)$ such that $\int_0^1 q_2(x)x(x - \alpha_{2,1})dx = 0$, and

it is easily verified that $q_2(x) = x - (3 - 4\alpha_{2,1})/(4 - 6\alpha_{2,1})$ is the required polynomial. $\alpha_{2,1} \neq 2/3$ if x_0 is sufficiently close to ζ , so $\alpha_{2,2}$ is well-defined.

Δ_3 and Δ_4 are respectively the value of the interpolating polynomial $p_3(x)$ and its derivative $p_3'(x)$ at $x = z_2 = x_0 + \Delta_2$, so (6.6) is (to sufficient accuracy) the approximation

$$z_2 - p_3(z_2)/p_3'(z_2) - \frac{1}{2}p_3^2(z_2)p_3''(z_2)/(p_3'(z_2))^3$$

6.3 Eighth-order methods

If $k = m = 1$, $n = 3$, the relevant Jacobi polynomial is $G_3(2, 2, x) = (35x^3 - 60x^2 + 30x - 4)/35$, with zeros

$$\left. \begin{aligned} \alpha &= 0.21234053823915294397\dots, \\ \beta &= 0.59053313555926528913\dots, \\ \gamma &= 0.91141204048729605260\dots \end{aligned} \right\} \quad (6.7)$$

and

Method 113a

$$\alpha_1 = \alpha \quad (6.8)$$

where α is given by (6.7),

Δ_1 , $f_1^{(1)}$, Δ_2 , and $\alpha_{2,1}$ as for method 112a,

$$a = 100\alpha_{2,1}^2 - 120\alpha_{2,1} + 30,$$

$$b = 60\alpha_{2,1}^2 - 75\alpha_{2,1} + 20,$$

$$c = 30\alpha_{2,1}^2 - 40\alpha_{2,1} + 12,$$

$$\alpha_{2,2} = (b - (b^2 - ac)^{1/2})/a, \quad (6.9)$$

$f_2^{(1)}$, t_1 , t_2 , v_1 , v_2 , Δ_3 , and Δ_4 as for method 112a,

$$\Delta_5 = \Delta_2 - \Delta_3/\Delta_4,$$

$$\alpha_{3,1} = \alpha_1 \Delta_1 / \Delta_5,$$

$$\alpha_{3,2} = \alpha_{2,2} \Delta_2 / \Delta_5,$$

$$\alpha_{3,3} = \frac{12 - 15(\alpha_{3,1} + \alpha_{3,2}) + 20\alpha_{3,1}\alpha_{3,2}}{15 - 20(\alpha_{3,1} + \alpha_{3,2}) + 30\alpha_{3,1}\alpha_{3,2}}, \quad (6.10)$$

$$f_3^{(1)} = f'(x_0 + \alpha_{3,3}\Delta_5),$$

$$t_3 = (f_3^{(1)} - f_0^{(1)}) / (\alpha_{3,3}\Delta_5),$$

$$a_{1,1} = g(\alpha_{3,1}, \alpha_{3,2}, \alpha_{3,3}) \equiv \frac{6\alpha_{3,2}\alpha_{3,3} - 4(\alpha_{3,2} + \alpha_{3,3}) + 3}{12(\alpha_{3,2} - \alpha_{3,1})(\alpha_{3,3} - \alpha_{3,1})},$$

$$a_{1,2} = g(\alpha_{3,2}, \alpha_{3,3}, \alpha_{3,1}),$$

$$a_{1,3} = g(\alpha_{3,3}, \alpha_{3,1}, \alpha_{3,2}),$$

$$a_{2,1} = h(\alpha_{3,1}, \alpha_{3,2}, \alpha_{3,3}) \equiv \frac{(1 - \alpha_{3,2})(1 - \alpha_{3,3})}{(\alpha_{3,2} - \alpha_{3,1})(\alpha_{3,3} - \alpha_{3,1})},$$

$$a_{2,2} = h(\alpha_{3,2}, \alpha_{3,3}, \alpha_{3,1}),$$

$$a_{2,3} = h(\alpha_{3,3}, \alpha_{3,1}, \alpha_{3,2}),$$

$$\begin{bmatrix} \Delta_6 \\ \Delta_7 \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \end{bmatrix} \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix},$$

$$\Delta_8 = f_0^{(0)} + f_0^{(1)}\Delta_5 + \Delta_5^2\Delta_6.$$

$$\Delta_9 = f_0^{(1)} + \Delta_5\Delta_7,$$

and

$$x_1 = x_0 + \Delta_5 - \Delta_8/\Delta_9 - \frac{1}{2}\Delta_8^2 v_1/\Delta_9.$$

The error constant is

$$\begin{aligned} K(\alpha, \beta, \gamma) = & [3\phi_8 - 21\phi_2\phi_7/(1-\alpha) \\ & + 9(35(1-\gamma) - 3/(1-\beta))\phi_3\phi_6 \\ & - 25(9 - 44\gamma + 42\gamma^2)\phi_4\phi_5]/3675. \end{aligned} \quad (6.11)$$

Comments on method 113a

It is easy to verify that the polynomial

$$(100t^2 - 120t + 30)x^2 - 2(60t^2 - 75t + 20)x + (30t^2 - 40t + 12)$$

is orthogonal to 1 and x with respect to the weight function $x(x-t)$ on $[0, 1]$. In view of (2.17), this explains (6.9).

It may also be verified that, if $\alpha_{2,2}$ is defined by (6.9), then $b^2 > ac$ for all real $\alpha_{2,1}$, and $\alpha_{2,2} \rightarrow \beta$ as $\alpha_{2,1} \rightarrow \alpha$.

Similarly,

$$\int_0^1 [(15-20(t+u)+30tu)x - (12-15(t+u)+20tu)]x(x-t)(x-u)dx = 0,$$

explaining (6.10), and $\alpha_{3,3} \rightarrow \gamma$ as $\alpha_{3,1} \rightarrow \alpha$ and $\alpha_{3,2} \rightarrow \beta$.

Methods 113a - 113f

By taking $\alpha_1 = \alpha, \beta$ or γ in (6.8), and either sign before the square root in (6.9), we get six different methods, one of which is method 113a. Table 6.2 summarizes these methods.

The error constants

$$K = A\phi_8 + B\phi_2\phi_7 + C\phi_3\phi_6 + D\phi_4\phi_5 \quad (6.12)$$

are obtained by suitably permuting α, β and γ in (6.11).

Numerical values of A, B, C and D are given in the table.

Table 6.2: Some eighth-order methods

Method	α_1	Sign in (6.9)	Error Constant	A	B	C	D
113a	α	-	$K(\alpha, \beta, \gamma)$	0.000816	-0.007255	-0.010349	-0.025756
113b	α	+	$K(\alpha, \gamma, \beta)$	0.000816	-0.007255	-0.047837	0.015897
113c	β	-	$K(\beta, \gamma, \alpha)$	0.000816	-0.013955	-0.015420	-0.010549
113d	β	+	$K(\beta, \alpha, \gamma)$	0.000816	-0.013955	-0.001734	-0.025756
113e	γ	-	$K(\gamma, \alpha, \beta)$	0.000816	-0.064504	0.025770	0.015897
113f	γ	+	$K(\gamma, \beta, \alpha)$	0.000816	-0.064504	0.049571	-0.010549

Comparison of error constants

It is natural to ask which of methods 113a - 113f has minimal error constant $|K|$. From (6.11), this depends on the behaviour of ϕ_2, \dots, ϕ_8 . Suppose that f is holomorphic in a closed disk $|z - \zeta| \leq r$, so there is a constant c such that

$$|\phi_j| \leq cr^{1-j} \quad (6.13)$$

for all $j \geq 2$. (Conversely, if (6.13) holds then f is holomorphic in the open disk $|z - \zeta| < r$.) From (6.12) and (6.13),

$$|K| \leq (|A| + |B| + |C| + |D|)cr^{-7}, \quad (6.14)$$

so a reasonable criterion is to choose the method with minimal $\sigma = |A| + |B| + |C| + |D|$. (A similar but slightly different criterion is given by King (1966).) On our criterion, method 113c ($\sigma = 0.0399$) is slightly better than methods 113d ($\sigma = 0.0414$) and 113a ($\sigma = 0.0434$), but the difference is small. On the same criterion method 112a is better than method 112b.

6.4 A seventh-order method

If $k = 1, m = n = 2$, the relevant Jacobi polynomial is $G_2(3, 3, x) = x^2 - 4x/3 + 2/5$, with zeros $(10 \pm \sqrt{10})/15$. By Theorem 2.1, methods in $C'(1, 2, 2)$ have order 7. One such method is the following.

Method 122

$$\alpha_1 = (10 - \sqrt{10})/15, \quad \alpha_2 = (10 + \sqrt{10})/15,$$

$$\Delta_1 = -f_0^{(0)}/f_0^{(1)}, \quad \Delta_2 = \Delta_1 - \frac{1}{2}f_0^{(2)}\Delta_1^2/f_0^{(1)},$$

$$f_1^{(1)} = f'(x_0 + \alpha_1\Delta_2), \quad f_2^{(1)} = f'(x_0 + \alpha_2\Delta_2),$$

x_1 an approximate zero of p_3 , satisfying (2.3) and

$$p_3(x_1) = O(\delta^8) \tag{6.15}$$

(an explicit formula for x_1 , similar to those above, is easy to derive).

Provided (6.15) holds, instead of merely (2.4), the error constant is $-\phi_7/225 - 2\phi_2^2\phi_5/3 + \phi_3\phi_5/3$. Unlike some of the methods above, method 122 remains the same when α_1 and α_2 are interchanged.

7. OTHER METHODS

The obvious method which uses evaluations of f , f' and f'' at x_0 , followed by evaluations of f' and f'' at another point $y_1 = x_0 + O(\delta)$, has order five. It is natural to ask if there is a choice of y_1 for which the order is six. Theorems 2.1 and 4.1 are not applicable, but for a similar analysis to go through we need a nonzero number α such that $P(1) = 0$ for all fifth-degree polynomials $P(x)$ satisfying $P(0) = P'(0) = P''(0) = P'(\alpha) = P''(\alpha) = 0$ (compare Lemma 4.3). This condition gives

$$\det \begin{bmatrix} 1 & 1 & 1 \\ 3 & 4\alpha & 5\alpha^2 \\ 6 & 12\alpha & 20\alpha^2 \end{bmatrix} = 0, \quad (7.1)$$

which (using $\alpha \neq 0$) reduces to

$$10\alpha^2 - 15\alpha + 6 = 0. \quad (7.2)$$

Since (7.2) has no real roots, we do not know any real method of order six.

Similarly, it is natural to ask if there is an eighth-order method which uses evaluations of f , f' , f'' , and f''' at x_0 , followed by evaluations of f' , f'' and f''' at some point y_1 . In this case we need a nonzero α satisfying

$$\det \begin{bmatrix} 1 & 1 & 1 & 1 \\ 4 & 5\alpha & 6\alpha^2 & 7\alpha^3 \\ 12 & 20\alpha & 30\alpha^2 & 42\alpha^3 \\ 24 & 60\alpha & 120\alpha^2 & 210\alpha^3 \end{bmatrix} = 0, \quad (7.3)$$

which reduces to

$$35\alpha^3 - 84\alpha^2 + 70\alpha - 20 = 0, \quad (7.4)$$

and (7.4) has one real root $\alpha = 0.74494327207110343664\dots$

We shall not give the details of this method, but note that, provided the polynomial approximations are solved sufficiently accurately, the error constant is simply $(1 - \alpha)^4 \phi_8$. Some numerical results for this method (S_8) are given in Section 9.

These examples suggest several open questions. For example, there are methods of order $2m+1$ which use evaluations of $f(x_0), \dots, f^{(m)}(x_0)$ and $f'(y_1), \dots, f^{(m)}(y_1)$, but for which m are there (real) methods of order $2m+2$? (There are such methods for $m = 1, 3, 5$, etc., but none is known for even m .) Similarly, for which n are there methods of order $3(n+1)$ using evaluations of $f(x_0), f'(x_0), f''(x_0)$ and $f'(y_i), f''(y_i)$ for suitable real points y_1, \dots, y_n ? (No such methods are known for $n > 0$.)

A possible extension of our results is to methods where

the evaluation of derivatives $f^{(k)}(y_i)$ is replaced by the evaluation of definite integrals

$$f^{(-1)}(y_i) = \int_{x_0}^{y_i} f(t) dt, \quad f^{(-2)}(y_i) = \int_{x_0}^{y_i} \left[\int_{x_0}^u f(t) dt \right] du = \int_{x_0}^{y_i} (y_i - t) f(t) dt,$$

etc. For example, if $m \geq 1$, $n = 1$ and $0 < k \leq m+1$, our theory gives

$$\alpha_1 = \frac{m+2-k}{m+2}. \quad (7.5)$$

It is suggestive that the fourth-order methods of King (1973) have $\alpha_1 = 1$, which is obtained formally by setting $k = 0$ in (7.5). Similarly, $\alpha_1 = (m+3)/(m+2)$ is obtained formally by setting $k = -1$ in (7.5), and there is in fact a method of order $m+3$ which uses evaluations of $f(x_0), \dots, f^{(m)}(x_0)$ and $f^{(-1)}(y_1)$, where y_1 is determined in the usual way from this value of α_1 . (For details of this method, see Kacwicz (1974) and Wozniakowski (1974).) However, we do not expect the formal analogy to hold for large n . One reason for this is that an order of at least $2^{\lfloor n/2 \rfloor}$ is attainable by methods using one evaluation of f and n evaluations of $f^{(-1)}$ per iteration, for two evaluations of $f^{(-1)}$ can be used to approximate one evaluation of f to any desired accuracy. Note that finding a zero of f using evaluations of $f^{(-1)}$ is equivalent to finding a turning point of $g = f^{(-1)}$ using evaluations of g , so the methods discussed in Brent (1973a) may also be used.

8. A CLASS OF NONLINEAR RUNGE-KUTTA METHODS

Consider the ordinary differential equation

$$dx/dt = g(x) \quad (8.1)$$

with initial condition $x(t_0) = x_0$. If $t_1 = t_0 + h$ and we want to find $x(t_1)$, we need a zero of the function

$$f(x) = \int_{x_0}^x \frac{du}{g(u)} - h. \quad (8.2)$$

Since $f'(x) = 1/g(x)$ may be computed almost as easily as $g(x)$, and $f(x_0) = -h$ is known without any computation, a zero-finding method using evaluations of $f(x_0)$, $f'(x_0)$, $f'(y_1)$, ..., $f'(y_n)$ is applicable. For example, one iteration of method 111a (see Section 6) may be written as

$$\left. \begin{aligned} g_0 &= g(x_0), \\ \Delta_1 &= hg_0, \\ g_\alpha &= g(x_0 + 2\Delta_1/3), \\ x_1 &= x_0 + \Delta_1(5 + 3(g_\alpha/g_0)^2)/8, \end{aligned} \right\} \quad (8.3)$$

and

when f is defined by (8.2). The equations (8.3) give an explicit method of Runge-Kutta type for solving the differential equation (8.1). The method is "nonlinear" because the formula for x_1 is nonlinear in g_0 and g_α . Since the zero-finding method is

fourth-order, $x_1 = x(t_1) + O(h^4)$, so the Runge-Kutta method (8.3) has order three. Note the difference in the definitions of order for differential equation methods (Henrici (1962)) and methods for finding zeros.

Similarly, for any zero-finding method $M(1, 1, n)$ there is a corresponding nonlinear Runge-Kutta method. Numerical results for some of these methods are given in Section 9. By Theorem 4.1, the order of the zero-finding method is $2(n+1)$, so the order of the Runge-Kutta method is $2n+1$. Thus (with $\nu = n+1$) we have the following theorem, which is related to some results of Nørsett (1974) and Osborne (1967).

Theorem 8.1

If $\nu > 0$, there is an explicit, nonlinear, Runge-Kutta method of order $2\nu-1$, using ν function evaluations per iteration, for single differential equations of the form (8.1).

Theorem 8.1 contrasts with the known results for (linear) Runge-Kutta methods for systems of differential equations of the form

$$dx/dt = g(x, t). \quad (8.4)$$

From Butcher (1965), the highest order attainable by such methods using ν evaluations of $g(x, t)$ per iteration is

48.

$$p^*(v) = \begin{cases} v & \text{if } 1 \leq v \leq 4, \\ v-1 & \text{if } 5 \leq v \leq 7, \\ v-2 & \text{if } 8 \leq v \leq 9, \end{cases}$$

and

$$p^*(v) \leq v-2 \quad \text{if } v \geq 10.$$

Thus, it seems unlikely that a generalization of our nonlinear methods to systems of differential equations is possible, although an extension to single equations of the form $dx/dt = g(x, t)$ may be possible.

9. NUMERICAL RESULTS

In this section we summarize the results of numerical tests of some of the methods described in Sections 6 to 8. Table 9.1 gives $\epsilon_i = x_i - \zeta$ ($i = 1, \dots, 4$) for the function

$$f(x) = x^2 - x - 3 + 4/x - \log_2 x, \quad (9.1)$$

with a simple zero at $\zeta = 2$, from the initial approximation $x_0 = 10$. Multiple-precision arithmetic was used to obtain ϵ_3 and ϵ_4 accurately in order to demonstrate the superlinear convergence. The order ρ and asymptotic error constant K are as given above.

Table 9.1: Numerical results for the function (9.1)

Method	Order	K	ϵ_1	ϵ_2	ϵ_3	ϵ_4
111a	4	3.61	1.56	$1.80 \cdot 10^{-1}$	$1.33 \cdot 10^{-3}$	$1.12 \cdot 10^{-11}$
111b	4	2.60	1.44	$1.43 \cdot 10^{-1}$	$5.02 \cdot 10^{-4}$	$1.65 \cdot 10^{-13}$
111c	4	$3.32 \cdot 10^{-1}$	$9.87 \cdot 10^{-1}$	$4.09 \cdot 10^{-2}$	$8.18 \cdot 10^{-7}$	$1.49 \cdot 10^{-25}$
111d	4	$3.32 \cdot 10^{-1}$	$4.50 \cdot 10^{-1}$	$3.53 \cdot 10^{-3}$	$5.05 \cdot 10^{-11}$	$2.16 \cdot 10^{-42}$
112a	6	$1.12 \cdot 10^{-2}$	$3.86 \cdot 10^{-1}$	$5.86 \cdot 10^{-5}$	$4.55 \cdot 10^{-28}$	$9.94 \cdot 10^{-167}$
113a	8	$3.69 \cdot 10^{-4}$	$1.49 \cdot 10^{-1}$	$1.03 \cdot 10^{-10}$	$4.77 \cdot 10^{-84}$	$9.81 \cdot 10^{-671}$
122	7	$6.90 \cdot 10^{-2}$	$6.27 \cdot 10^{-1}$	$1.79 \cdot 10^{-3}$	$4.02 \cdot 10^{-21}$	$1.17 \cdot 10^{-144}$
S_8	8	$2.82 \cdot 10^{-5}$	$6.44 \cdot 10^{-1}$	$2.66 \cdot 10^{-3}$	$4.52 \cdot 10^{-21}$	$4.94 \cdot 10^{-168}$

All the methods converge although x_0 is not very close to ζ . The higher order methods give good approximations after two iterations (e.g. $\epsilon_2 = 1.03 \cdot 10^{-10}$ for method 113a). Method 111d is the best of the fourth-order methods, at least for the function (9.1).

Table 9.2 illustrates equation (6.1). For f given by (9.1) and various $\epsilon_0 = x_0 - \zeta$, the table gives the computed values $K(\epsilon_0) = \epsilon_1/\epsilon_0^p$, and the predicted asymptotic error constant $K = \lim_{\epsilon_0 \rightarrow 0} K(\epsilon_0)$. The agreement between the predicted and computed values is good.

Table 9.2: Computed and predicted error constants

Method	Order	$K(10^{-4})$	$K(10^{-8})$	$K(10^{-12})$	K
112a	6	1.12131 ⁻²	1.12045 ⁻²	1.12045 ⁻²	1.12045 ⁻²
113a	8	3.68987 ⁻⁴	3.68889 ⁻⁴	3.68889 ⁻⁴	3.68889 ⁻⁴
122	7	6.89218 ⁻²	6.89766 ⁻²	6.89766 ⁻²	6.89766 ⁻²
S_8	8	7.11402 ⁻²	3.53173 ⁻⁵	2.81896 ⁻⁵	2.81889 ⁻⁵

Finally, Table 9.3 gives numerical results for some of the Runge-Kutta methods described in Section 8 and some more usual Runge-Kutta methods. Suppose we want to tabulate solutions of

$$(2\pi)^{-\frac{1}{2}} \int_0^x e^{-u^2/2} du = t \quad (9.2)$$

for $t = 0.0(0.1)0.4$ or $t = 0.0(0.01)0.49$. Equivalently, we want to solve the differential equation

$$dx/dt = (2\pi)^{1/2} e^{x^2/2} \quad (9.3)$$

with initial condition $x(0) = 0$. The following methods are possible:

1. Numerical integration of the left side of (9.2), followed by interpolation. This would be appropriate if the solution were to be tabulated for given values of x , but it is inconvenient if the solution is required for given values of t .
2. Using some method for second-order differential equations (or systems of first order equations) applied to the equation $d^2x/dt^2 = x(dx/dt)^2$ with appropriate initial conditions. This avoids the repeated evaluation of exponentials, but depends on special properties of the integrand in (9.2), so is not generally applicable.
3. Using some method for first-order differential equations applied to (9.3). We compare some such methods.

In Table 9.3, method 111d' is the (third-order) nonlinear Runge-Kutta method derived from the (fourth-order) zero-finding method 111d as described in Section 8, and similarly for 112a' and 113a'. Method RK4 is the classical fourth-order method of Kutta (1901), and RK7 is the seventh-order method of Shanks (1966). The number of evaluations of $e^{x^2/2}$ per iteration is

denoted by ν . If $\hat{x}_h(t)$ is the computed solution (using step size h), the error $e_h(t)$ is defined by

$$e_h(t) = (2\pi)^{-\frac{1}{2}} \int_0^{\hat{x}_h(t)} e^{-u^2/2} du - t.$$

Computations were performed with double-precision floating-point arithmetic on a Univac 1108 computer (fraction length 60 bits).

Table 9.3: Comparison of Runge-Kutta methods

Method	ν	Order	$e_{0.1}(0.2)$	$e_{0.01}(0.2)$	$e_{0.1}(0.4)$	$e_{0.01}(0.4)$
111d'	2	3	-4.59'-5	-5.66'-8	-9.45'-6	1.49'-7
112a'	3	5	1.22'-7	2.54'-12	3.16'-6	-2.47'-11
113a'	4	7	-6.28'-10	-6.29'-17	3.86'-8	3.69'-15
RK4	4	4	-3.74'-7	2.12'-11	1.95'-5	7.90'-9
RK7	9	7	-1.76'-9	-2.86'-16	-5.19'-7	-1.67'-13

The table suggests that our methods are more accurate than standard Runge-Kutta methods with the same number of function evaluations per iteration, and more efficient than standard methods with the same order. For example, method 113a' is considerably more accurate than RK4, though both methods require four function evaluations per iteration; and 113a' is slightly more accurate than RK7, which requires nine function evaluations per iteration. This is not surprising, for our methods are applicable only

53.

to single differential equations of the special form (8.1),
but the standard methods are applicable to general systems of
the form (8.4). The use of method RK4 to solve nonlinear
equations was suggested by Kizner (1964).

ACKNOWLEDGEMENT

I wish to thank J.F. Traub, H.T. Kung and M.R. Osborne for several useful suggestions.

REFERENCES

- M. Abramowitz and I.A. Stegun (1964), Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, National Bureau of Standards, Washington, D.C.
- C.W.R. de Boor and B. Swartz (1973), Collocation at Gaussian points, SIAM J. Numer. Anal., 10, pp. 582-606.
- R.P. Brent (1973a), Algorithms for Minimization without Derivatives, Prentice Hall, Englewood Cliffs, New Jersey.
- R.P. Brent (1973b), Some efficient algorithms for solving systems of nonlinear equations, SIAM J. Numer. Anal., 10, pp. 327-344.
- R.P. Brent (1974), Computer Solution of Nonlinear Equations, in preparation.
- R.P. Brent, S. Winograd and P. Wolfe (1973), Optimal iterative processes for root-finding, Numer. Math., 20, pp. 327-341.
- J.C. Butcher (1965), On the attainable order of Runge-Kutta methods, Math. Comp., 19, pp. 408-417.
- P.J. Davis (1963), Interpolation and Approximation, Blaisdell, New York.
- P. Henrici (1962), Discrete Variable Methods in Ordinary Differential Equations, Wiley, New York.
- E. Isaacson and H.B. Keller (1966), Analysis of Numerical Methods, Wiley, New York.

- P. Jarratt (1969), Some efficient fourth order multipoint methods for solving equations, BIT, 9, pp. 119-124.
- B. Kacewicz (1974), An integral-interpolatory iterative method for the solution of scalar equations, to appear.
- R.F. King (1966), Runge-Kutta methods with constrained minimum error bounds, Math. Comp., 20, pp. 386-391.
- R.F. King (1971), A fifth-order family of modified Newton methods, BIT, 11, pp. 409-412.
- R.F. King (1972), Tangent methods for nonlinear equations, Numer. Math., 18, pp. 298-304.
- W. Kizner (1964), A numerical method for finding solutions of nonlinear equations, J. SIAM, 12, pp. 424-428.
- H.T. Kung and J.F. Traub (1973a), Optimal order of one-point and multipoint iteration, Computer Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pennsylvania (to appear in J. ACM).
- H.T. Kung and J.F. Traub (1973b), Computational complexity of one-point and multipoint iteration, in Complexity of Real Computation (R. Karp, ed.), American Math. Soc., New York.
- H.T. Kung and J.F. Traub (1973c), Optimal order and efficiency for iterations with two evaluations, Computer Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pennsylvania.
- W. Kutta (1901), Beitrag zur näherungsweise Integration totaler Differentialgleichungen, Z. Math. Phys., 46, pp. 435-452.

- C.A. Micchelli and T.J. Rivlin (1973), Numerical integration rules near Gaussian quadrature, Report RC4387, IBM Research Center, Yorktown Heights, New York.
- S.P. Nørsett (1974), One-step methods of Hermite type for numerical integration of stiff systems, BIT, 14, pp 63-77.
- J.M. Ortega and W.C. Rheinboldt (1970), Iterative Solution of Nonlinear Equations in Several Variables, Academic Press, New York.
- M.R. Osborne (1967), Minimising truncation error in finite difference approximations to ordinary differential equations, Math. Comp., 21, pp. 135-145.
- M.R. Osborne (1974), Collocation, difference equations, and stitched function representations, in Proceedings of a Symposium held at Dublin (J.J.H. Miller, ed.), Academic Press.
- A.M. Ostrowski (1966), Solution of Equations and Systems of Equations (2nd ed.), Academic Press, New York.
- E.B. Shanks (1966), Solutions of differential equations by evaluations of functions, Math. Comp., 20, pp. 21-38.
- G. Szegö (1959), Orthogonal Polynomials, AMS Colloquium Publication, vol. 23, American Math. Soc., New York.
- J.F. Traub (1964), Iterative Methods for the Solution of Equations, Prentice-Hall, Englewood Cliffs, New Jersey.

J.H. Wilkinson (1963), *Rounding Errors in Algebraic Processes*, HMSO, London.

H. Wozniakowski (1974), *Generalized information and maximal order of iteration for operator equations*, Computer Sci. Dept., Carnegie-Mellon Univ., Pittsburgh, Pennsylvania.

APPENDIX: FORTRAN SUBROUTINES

The Fortran subroutines DZ111D, DZ112A and DZ113A are double-precision implementations of the methods 111d, 112a and 113a described in Section 6. They have been tested on a Univac 1108 computer using the Fortran V compiler.

```

1      FUNCTION DZ111D (FP, X0, F0, F0P)
2      C
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
      DZ111D IMPROVES AN APPROXIMATION X0 TO A ZERO OF A
      FUNCTION F(X) WITH DERIVATIVE F'(X) = FP(X) WHICH
      MAY BE COMPUTED BY THE FUNCTION FP. IT IS ASSUMED
      THAT F0 = F(X0), F0P = F'(X0), AND F0P.NE.0
      THE METHOD USED IS OF FOURTH ORDER AND CALLS FP ONCE.
      OBVIOUSLY, DZ111D SHOULD ONLY BE USED IF FT IS
      EASIER TO EVALUATE THAN F (E.G. IF F IS DEFINED BY AN
      INTEGRAL).
      NOTE THAT EVERYTHING IS DOUBLE-PRECISION AND FP MUST
      BE DECLARED EXTERNAL IN THE CALLING PROGRAM.
      IMPLICIT REAL*8 (A-H, O-Z)
      DEL1 = 0D0
      CHECK FOR EXACT ZERO
      IF (F0.EQ.0D0) GO TO 10
      CHECK THAT F0P NONZERO
      IF (F0P.EQ.0D0) GO TO 20
      COMPUTE NEWTON STEP
      DEL1 = -F0/F0P
      EVALUATE F' AT X0 + 2*DEL1/3
      F11 = FP(X0 + (2D0/3D0)*DEL1)
      COMPUTE FOURTH ORDER APPROXIMATION
      D = 3D0*F11/F0P - 2D0
      S = 0D0
      IF (D.GT.0D0) S = DSQRT(D)
      DZ111D = X0 + 2D0*DEL1/(1D0 + S)
      RETURN
      C
      10 ABNORMAL EXIT, RETURN NEWTON STEP
      DZ111D = X0 + DEL1
      RETURN
      C
      ERROR IF F0P = 0
      20 PRINT 30
      30 FORMAT (/, ' *** F0P = 0.0 IN CALL TO DZ111D ***')
      STOP
      END

```



```

1      FUNCTION DZ113A (FP, X0, F0, F0P)
2
3      C
4      C
5      C
6      C
7      C
8      C
9      C
10     C
11     C
12     C
13     C
14     C
15     C
16     C
17     C
18     C
19     C
20     C
21     C
22     C
23     C
24     C
25     C
26     C
27     C
28     C
29     C
30     C
31     C
32     C
33     C
34     C
35     C
36     C
37     C
38     C
39     C
40     C
41     C
42     C
43     C
44     C
45     C
46     C
47     C
48     C
49     C
50     C
51     C
52     C
53     C
54     C
55     C
56     C
57     C
58     C

```

FUNCTION DZ113A (FP, X0, F0, F0P)
 DZ113A IMPROVES AN APPROXIMATION X0 TO A ZERO OF A
 FUNCTION F(X) WITH DERIVATIVE F'(X) = FP(X) WHICH
 MAY BE COMPUTED BY THE FUNCTION FP. IT IS ASSUMED
 THAT F0 = F(X0), F0P = F'(X0), AND F0P.NE.0
 THE METHOD USED IS OF EIGHTH ORDER AND CALLS FP THREE
 TIMES. OBVIOUSLY, DZ113A SHOULD ONLY BE USED IF F' IS
 EASIER TO EVALUATE THAN F (E.G. IF F IS DEFINED BY AN
 INTEGRAL).
 NOTE THAT EVERYTHING IS DOUBLE-PRECISION AND FP MUST
 BE DECLARED EXTERNAL IN THE CALLING PROGRAM.
 IMPLICIT REAL*8 (A-H, O-Z)
 DEL1 = 0.0D0
 CHECK FOR EXACT ZERO
 IF (F0.EQ.0.0D0) GO TO 10
 CHECK THAT F0P NONZERO
 IF (F0P.EQ.0.0D0) GO TO 20
 COMPUTE NEWTON STEP
 DEL1 = -F0/F0P
 ALP = 0.212340538239152943974758110124 ... IS A ROOT OF
 35*ALP**3 - 60*ALP**2 + 30*ALP - 4
 AD = 0.2123405382391529400*DEL1
 EVALUATE F' AT X0 + ALP*DEL1
 F11 = FP(X0 + AD)
 D = F11 + (0.2123405382391529400-100)*F0P
 IF (D.EQ.0.0D0) GO TO 10
 DEL2 IS A THIRD ORDER APPROXIMATION TO THE DISTANCE TO
 A ZERO.
 DEL2 = 0.500*DEL1*(F11 + (200*0.2123405382391529400-100)
 *F0P)/D
 IF (DEL2.EQ.0.0D0) GO TO 10
 A21 = AD/DEL2
 COMPUTE COEFFICIENTS OF QUADRATIC DETERMINING A22
 A = 3000 - A21*(12000 - A21*10000)
 B = 2000 - A21*(7500 - A21*6000)
 C = 1200 - A21*(4000 - A21*3000)
 DETERMINE A22 BY SOLVING QUADRATIC (B*B > A*C ALWAYS)
 D = B + DSQRT(B*B - A*C)
 IF ((C.EQ.0.0D0).OR.(D.EQ.0.0D0)) GO TO 10
 A22 = C/D
 BD = A22*DEL2
 PERFORM NEXT EVALUATION OF F'
 F21 = FP(X0 + BD)
 NOW COMPUTE FOURTH ORDER APPROXIMATION
 T1 = (F11 - F0P)/AD
 T2 = (F21 - F0P)/BD
 DBA = A22 - A21
 IF (DBA.EQ.0.0D0) GO TO 10
 V1 = (A22*T1 - A21*12)/DBA
 V2 = (T2 - T1)/DBA
 DEL4 = F0P + DEL2*(V1 + V2)
 IF (DEL4.EQ.0.0D0) GO TO 10
 DEL5 = DEL2 = (F0 + DEL2*(F0P + DEL2
 *(0.500*V1 + (100/300)*V2)))/DEL4

```

59      IF (DEL5, EQ, 000) GO TO 10
60      C      UPDATE A21 AND A22
61      A31 = A0/DEL5
62      A32 = A0/DEL5
63      C      COMPUTE A33
64      SAR = A31 + A32
65      PAB = A31*A32
66      O = 1500 + 3000*PAB - 2000*SAB
67      IF (O, EQ, 000) GO TO 10
68      A33 = (1200 + 2000*PAB - 1500*SAB)/O
69      IF (A33, EQ, 000) GO TO 10
70      GD = A33*DEL5
71      C      FINAL EVALUATION OF F1
72      F31 = FP(X0 + GD)
73      C      NOW COMPUTE 8TH ORDER APPROXIMATION
74      T3 = (F31 - F0P)/GD
75      DBA = A32 - A31
76      DGB = A33 - A32
77      DAG = A31 - A33
78      IF ((DBA, EQ, 000), OR, (DGB, EQ, 000), OR, (DAG, EQ, 000))
79      2      GO TO 10
80      T1 = T1/(DBA*DAG)
81      T2 = T2/(DGB*DBA)
82      T3 = T3/(DAG*DGB)
83      DEL8 = F0 + DEL5*(F0P - DEL5 +
84      2      (T1*(0.500*A32*A33+0.2500-(100/300)*(A32+A33))
85      3      +T2*(0.500*A33*A31+0.2500-(100/300)*(A33+A31))
86      4      +T3*(0.500*A31*A32+0.2500-(100/300)*(A31+A32)))
87      A1 = A31 - 100
88      B1 = A32 - 100
89      G1 = A33 - 100
90      DEL9 = F0P - DEL5*(T1*B1+G1+T2*G1+A1+T3*A1+B1)
91      IF (DEL9, EQ, 000) GO TO 10
92      D = DEL8/DEL9
93      DZ113A = X0 + (DEL5 - 0*(100 + 0.500*0*V1/DEL9))
94      RETURN
95      C      ABNORMAL EXIT, RETURN NEWTON STEP
96      10  DZ113A = X0 + DEL1
97      RETURN
98      C      ERROR IF F0P = 0
99      20  PRINT 30
100     30  FORMAT (/, ' *** F0P = 0.0 IN CALL TO DZ113A *** ')
101     STOP
102     END

```

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) EFFICIENT METHODS FOR FINDING ZEROS OF FUNCTIONS WHOSE DERIVATIVES ARE EASY TO EVALUATE		5. TYPE OF REPORT & PERIOD COVERED Interim
7. AUTHOR(s) Richard Brent		6. PERFORMING ORG. REPORT NUMBER
9. PERFORMING ORGANIZATION NAME AND ADDRESS Carnegie-Mellon University Dept. of Computer Science Pittsburgh, PA 15213		8. CONTRACT OR GRANT NUMBER(s) N0014-67-0314-0010, NR440-422
11. CONTROLLING OFFICE NAME AND ADDRESS Office of Naval Research Arlington, VA 22217		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		12. REPORT DATE December 1974
		13. NUMBER OF PAGES 64
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution 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)		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Some multipoint iterative methods without memory for approximating simple zeros of functions of one variable are described. for $m \geq 0$, $n \geq 0$, and $m+1 \leq k \leq 0$, there exist methods which use on evaluation of f , f' , \dots , $f^{(m)}$ followed by n evaluations of $f^{(k)}$ for each iteration and have order of convergence $m+2n+1$. In particular, there are methods of order $2(n+1)$ which use one function evaluation and $n+1$ derivative evaluations per iteration. These methods naturally generalize the known cases $n = 0$ (Newton's method) and $n = 1$ (Jarratt's fourth-order method), and are useful if derivative evaluations are less expensive than function evaluations. Explicit, nonlinear, Runge-Kutta methods for		

20.

ABSTRACT-continued

the solution of a special class of ordinary differential equations may be derived from the methods for finding zeros of functions. Numerical examples and some Fortran subroutines are given.