

Research Article

A Method to Accelerate the Convergence of the Secant Algorithm

M. J. P. Nijmeijer

Heemraadssingel 182D, 3021 DM Rotterdam, The Netherlands

Correspondence should be addressed to M. J. P. Nijmeijer; mail@marconijmeijer.nl

Received 16 August 2014; Accepted 30 October 2014; Published 19 November 2014

Academic Editor: Raytcho Lazarov

Copyright © 2014 M. J. P. Nijmeijer. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present an acceleration technique for the Secant method. The Secant method is a root-searching algorithm for a general function f . We exploit the fact that the combination of two Secant steps leads to an improved, so-called first-order approximant of the root. The original Secant algorithm can be modified to a first-order accelerated algorithm which generates a sequence of first-order approximants. This process can be repeated: two n th order approximants can be combined in a $(n + 1)$ th order approximant and the algorithm can be modified to an $(n + 1)$ th order accelerated algorithm which generates a sequence of such approximants. We show that the sequence of n th order approximants converges to the root with the same order as methods using polynomial fits of f of degree n .

1. Introduction

The Secant algorithm is a textbook algorithm to find a numerical approximation of the root of a function $f(x)$. A root α is a solution of the equation $f(x) = 0$. Other such algorithms are, for example, the bisection algorithm, inverse quadratic interpolation, the regula-falsi algorithm, Muller's method, the Newton-Raphson algorithm, Steffensen's method, the Brent algorithm, and many more. These methods are discussed in many books and articles; see, for example, [1–11]. All the algorithms mentioned are intended for a general function f . They all take one, two, or more initial estimates of α as input and iteratively generate a sequence $\{x_n\}$ of approximants of α . The sequence converges to the root α for suitably chosen initial estimates and a function f meeting particular regularity requirements at and around α . The exact requirements differ from method to method. Root-finding plays a role in many problems, also when this is not immediately apparent. An example is the problem of solving a set of linear equations [12].

The Secant algorithm has the characteristics that (a) it is “derivative-free;” that is, it does not require the evaluation of a derivative of f and (b) it requires only one evaluation of f per iteration. The generated sequence $\{x_n\}$ converges superlinearly with order $\psi_0 = (1 + \sqrt{5})/2 \approx 1.6180$ for a large class of functions f .

It is important to stress that only one evaluation of f per iteration is needed. Situations which require an efficient root-finding algorithm are typically situations in which the execution time of the algorithm is dominated by the time needed to calculate the value of f . In these situations it is important that as few evaluations of f as possible are needed to estimate the root with a certain accuracy. An algorithm which requires m evaluations of f per iteration is therefore only competitive with the Secant algorithm if one iteration produces a better estimate than m subsequent Secant iterations. In other words, it must converge with an order larger than ψ_0^m . To the best of our knowledge, there are no derivative-free algorithms which achieve this, except for the generalizations of the Secant algorithm discussed below.

In this paper we derive a generalization of the Secant method with the following properties: (a) it is derivative-free, (b) it requires one evaluation of f per iteration, and (c) it achieves an order of convergence arbitrarily close to 2 for analytic functions f . The first two properties are the same as for the Secant algorithm. The last property shows that the method presented here will converge faster than the Secant method if f is sufficiently regular.

Other generalizations of the Secant algorithm with the same three properties are the method of inverse interpolation [2] and Sidi's method [13]. These two methods are based on polynomial fits to either the inverse of f (in the case of the

method of inverse interpolation) or to f itself (in the case of Sidi's method). Whereas the Secant method is based on straight-line fits to f , the polynomial fits of these methods can be of an arbitrary degree n . The resulting order of convergence is ψ_{n-1} for both methods. Hence the order is ψ_0 for the Secant method and $\psi_1 \approx 1.8393$ when a polynomial of degree 2 is used. By taking the order of the fitting polynomial large enough, the order of convergence becomes asymptotically quadratic if the function is sufficiently regular.

Another method which satisfies the three properties is the method of direct interpolation [2]. However this method requires that the root(s) of a polynomial of degree n are calculated in every iteration. This is not an attractive scheme except possibly for the case $n = 2$ which is known as Muller's method [1].

Our method is not based on polynomial fits. It was noted in [14, 15] that the results of two Secant steps can be combined into a better approximant of the root α in a way reminiscent of Aitken's delta-squared method [16] or Shank's transformation [17]. We take this idea further. We show that the process of combining approximants can be repeated. If we call the result of a Secant step an approximant of order zero, we demonstrate that two approximants of order n can be combined into an approximant of order $n + 1$. We devise an algorithm which generates these approximants. The n th order version of the algorithm generates a sequence of n th order approximants. We show that this sequence converges with order ψ_n to the root α if f is sufficiently regular.

Although our algorithm offers no specific advantage over the method of inverse interpolation or Sidi's method (all are derivative-free, require one evaluation of f per iteration, and achieve orders of convergence ψ_n), we think it is noteworthy that the Secant method can be sped up to higher orders of convergence without the use of polynomial fits. We suspect that the same acceleration technique can be applied to a broader set of iterative algorithms. We also see a possibility that our technique can lead to a parallel root-solving algorithm. These venues are however not explored in this paper.

The paper is organized as follows. We discuss preliminaries and recall the basic properties of the Secant sequence in Sections 2 and 3. We introduce the approximants in Section 4. The algorithm which generates these approximants is given in Section 5 and its convergence properties are derived in Section 6. We end with conclusions in Section 7.

2. Preliminaries

2.1. Order of Convergence. When a sequence $\{x_n\}$ converges to a limit α and the sequence has the property

$$\lim_{n \rightarrow \infty} \frac{|x_n - \alpha|}{|x_{n-1} - \alpha|^\psi} = A \quad (1)$$

with $A \neq 0$, then ψ is called the "order of convergence" or "order" of the sequence. The condition $A \neq 0$ is necessary to define the order of convergence uniquely. A is called the "asymptotic error constant" [2, 18, 19]. The larger the order of convergence, the faster the sequence converges.

2.2. Divided Differences. Throughout this paper we use "divided differences." We remind the reader of their definition and two important properties.

Let I be an open interval of real values and let f be a function $f : I \rightarrow \mathbb{R}$. Let $x_1, \dots, x_n \in I$ with $x_i \neq x_j$ if $i \neq j$. The " n th divided difference" $f[x_1, \dots, x_n]$ of f is defined recursively as

$$f[x_1, \dots, x_n] = \frac{f[x_1, \dots, x_{n-1}] - f[x_2, \dots, x_n]}{x_1 - x_n} \quad (2)$$

with

$$f[x] = f(x) \quad (3)$$

to terminate the recursion.

We use the following two properties of divided differences.

(i) If $f \in C^{n-2}(I)$ and the $(n-2)$ th derivative $f^{(n-2)}$ is Lipschitz continuous on I , then $f[x_1, \dots, x_n]$ is bounded on I^n .

This follows from the property that there exist a $\zeta_1, \zeta_2 \in [\min(x_1, \dots, x_n), \max(x_1, \dots, x_n)]$ such that

$$f[x_1, \dots, x_n] = \frac{1}{(n-1)!} \frac{f^{(n-2)}(\zeta_1) - f^{(n-2)}(\zeta_2)}{\zeta_1 - \zeta_2} \quad (4)$$

if $f \in C^{n-2}(I)$.

(ii) If $f \in C^{n-1}(I)$ and $\alpha \in I$ then $f[x_1, \dots, x_n]$ is continuous in the point $(x_1, \dots, x_n) = (\alpha, \dots, \alpha)$ with

$$\lim_{x_1, \dots, x_n \rightarrow \alpha} f[x_1, \dots, x_n] = \frac{f^{(n-1)}(\alpha)}{(n-1)!}. \quad (5)$$

This property is cited in many text books [2, 5, 18, 20]. It follows, for example, from the previous property in combination with the mean value theorem.

3. The Secant Algorithm

Suppose we have an open interval I of real values and a function $f : I \rightarrow \mathbb{R}$. Suppose $\alpha \in I$ and $f(\alpha) = 0$. A Secant step S is defined as

$$\begin{aligned} S(x_1, x_2) &= x_1 - f(x_1) \frac{x_1 - x_2}{f(x_1) - f(x_2)} \\ &= \frac{-f(x_2)x_1 + f(x_1)x_2}{f(x_1) - f(x_2)}. \end{aligned} \quad (6)$$

The Secant algorithm generates a sequence $\{x_n\}_{n=-1}^{\infty}$ which starts with two initial values $x_{-1}, x_0 \in I$ and develops as

$$x_n = S(x_{n-1}, x_{n-2}). \quad (7)$$

We can develop the sequence as long as $x_n \in I$. It can be shown [2, 5] that

$$x_n - \alpha = (x_{n-1} - \alpha)(x_{n-2} - \alpha) \frac{f[\alpha, x_{n-1}, x_{n-2}]}{f[x_{n-1}, x_{n-2}]}. \quad (8)$$

It can also be shown [2, 5] that if $f \in C^2(I)$, the first derivative $f^{(1)}(\alpha)$ and second derivative $f^{(2)}(\alpha)$ are not equal to zero, and the start values x_{-1} and x_0 are chosen close enough to α , then the Secant sequence $\{x_n\}_{n=-1}^{\infty}$ converges to α with

$$\lim_{n \rightarrow \infty} \frac{|x_n - \alpha|}{|x_{n-1} - \alpha|^{\psi_0}} = \left| \frac{f^{(2)}(\alpha)}{2f^{(1)}(\alpha)} \right|^{\psi_0 - 1}, \quad (9)$$

where $\psi_0 = (1 + \sqrt{5})/2$.

This means that the sequence converges with order ψ_0 under the conditions stated. It can be expected [10] that the sequence converges with a higher order if $f^{(2)}(\alpha) = 0$. In case $f^{(1)}(\alpha) = 0$ the sequence still converges but no longer superlinearly [21].

4. General Order Approximant

We define what we call an approximant of general order of the root α of a function f in this section. This definition is recursive. To study this approximant, we express the approximant directly in terms of f in Section 4.1. Two expressions are obtained: one involving f and polynomials in Section 4.1.1 and one involving divided differences in Section 4.1.2. These forms allows us to cast the approximant in a form which exposes its properties when we are close to α in Lemmas 3 and 4.

We define an n th order approximant S_n as follows.

Definition 1. Let $I \subset \mathbb{R}$ be an open interval of real values and f a function $f : I \rightarrow \mathbb{R}$. Define the n th order approximant S_n for $n = 1, 2, 3, \dots$ as

$$S_n(x_1, \dots, x_{n+2}) = \frac{S_{n-1}(x_1, \dots, x_{n+1})x_{n+2} - x_1 S_{n-1}(x_2, \dots, x_{n+2})}{S_{n-1}(x_1, \dots, x_{n+1}) + x_{n+2} - x_1 - S_{n-1}(x_2, \dots, x_{n+2})} \quad (10)$$

for all values $x_1, \dots, x_{n+2} \in I$ for which $S_{n-1}(x_1, \dots, x_{n+1})$ and $S_{n-1}(x_2, \dots, x_{n+2})$ are defined and for which the denominator is unequal to zero. $S_0(x_1, x_2)$ is the Secant step $S(x_1, x_2)$ defined in (6).

The reason why we call this an approximant will become clear shortly.

4.1. The Approximant in Terms of the Function f

4.1.1. *First Form.* From (6) we can write S_0 as

$$S_0(x_1, x_2) = \frac{-x_1/f(x_1) + x_2/f(x_2)}{-1/f(x_1) + 1/f(x_2)} \quad (11)$$

for $f(x_1), f(x_2) \neq 0$ and $f(x_1) \neq f(x_2)$. Working out S_1 with our recursive definition for S_n we obtain

$$\begin{aligned} S_1(x_1, x_2, x_3) &= \left(-\frac{x_1}{f(x_1)}(x_2 - x_3) + \frac{x_2}{f(x_2)}(x_1 - x_3) \right. \\ &\quad \left. - \frac{x_3}{f(x_3)}(x_1 - x_2) \right) \\ &\quad \cdot \left(-\frac{1}{f(x_1)}(x_2 - x_3) + \frac{1}{f(x_2)}(x_1 - x_3) \right. \\ &\quad \left. - \frac{1}{f(x_3)}(x_1 - x_2) \right)^{-1} \end{aligned} \quad (12)$$

for $f(x_1), f(x_2), f(x_3) \neq 0$ and the denominator not equal to zero.

We show in Appendix A that the general form of (11) and (12) is

$$\begin{aligned} S_n(x_1, \dots, x_{n+2}) &= \frac{\sum_{i=1}^{n+2} (x_i/f(x_i)) a_{n,i}(x_1, \dots, x_{n+2})}{\sum_{i=1}^{n+2} (1/f(x_i)) a_{n,i}(x_1, \dots, x_{n+2})} \end{aligned} \quad (13)$$

for $f(x_1), \dots, f(x_{n+2}) \neq 0$ and the denominator not equal to zero. The $a_{n,i}$ are polynomials and are given by

$$a_{n,i}(x_1, \dots, x_{n+2}) = (-1)^i \prod_{\substack{j=1 \\ j \neq i}}^{n+2} \prod_{\substack{k=j+1 \\ k \neq i}}^{n+2} (x_j - x_k) \quad (14)$$

for $n = 0, 1, 2, \dots$ and $i = 1, \dots, n + 2$. Throughout the paper we follow the convention that $\prod_{i=j}^k (\dots) = 1$ for $k < j$.

The condition $f(x_i) \neq 0$ for all i must be imposed for the form (13) but can be lifted by multiplying both numerator and denominator with the product $\prod_{i=1}^{n+2} f(x_i)$.

Examples of $a_{n,i}$ for $n = 0, 1, 2$ are

$$\begin{aligned} a_{0,1}(x_1, x_2) &= -1, \\ a_{0,2}(x_1, x_2) &= 1, \\ a_{1,1}(x_1, x_2, x_3) &= -(x_2 - x_3), \\ a_{1,2}(x_1, x_2, x_3) &= (x_1 - x_3), \\ a_{1,3}(x_1, x_2, x_3) &= -(x_1 - x_2), \\ a_{2,1}(x_1, x_2, x_3, x_4) &= -(x_2 - x_3)(x_2 - x_4)(x_3 - x_4), \\ a_{2,2}(x_1, x_2, x_3, x_4) &= (x_1 - x_3)(x_1 - x_4)(x_3 - x_4), \\ a_{2,3}(x_1, x_2, x_3, x_4) &= -(x_1 - x_2)(x_1 - x_4)(x_2 - x_4), \\ &\vdots \end{aligned} \quad (15)$$

4.1.2. *Second Form.* We derive a second expression for S_n with the help of the lemma below.

Lemma 2. Let \mathcal{S}_n be defined as

$$\mathcal{S}_n(x_1, \dots, x_{n+2}) = \sum_{i=1}^{n+2} g(x_i) a_{n,i}(x_1, \dots, x_{n+2}) \quad (16)$$

for some function $g(x)$. Then

$$\begin{aligned} \mathcal{S}_n(x_1, \dots, x_{n+2}) \\ = - \left\{ \prod_{i=1}^{n+2} \prod_{j=i+1}^{n+2} (x_i - x_j) \right\} g[x_1, \dots, x_{n+2}]. \end{aligned} \quad (17)$$

Proof. The proof can be obtained by elementary manipulations if we remember [18] that

$$g[x_1, \dots, x_{n+2}] = \sum_{i=1}^{n+2} \frac{g(x_i)}{\prod_{\substack{j=1 \\ j \neq i}}^{n+2} (x_i - x_j)}. \quad (18)$$

□

Application of Lemma 2 to (13) with $g(x) = x/f(x)$ for the numerator and $g(x) = 1/f(x)$ for the denominator of (13) brings S_n in the form

$$S_n(x_1, \dots, x_{n+2}) = \frac{(x/f)[x_1, \dots, x_{n+2}]}{(1/f)[x_1, \dots, x_{n+2}]} \quad (19)$$

provided that $f(x_i) \neq 0$ for all $i = 1, \dots, n+2$ and that the denominator is not equal to zero. The numerator $(x/f)[x_1, \dots, x_{n+2}]$ in (19) is the $(n+2)$ th divided difference of the function $x/f(x)$. The denominator $(1/f)[x_1, \dots, x_{n+2}]$ is $(n+2)$ th divided difference of the function $1/f(x)$.

4.2. *The Approximant Near the Root.* Suppose the function f in the definition of S_n has a root at $\alpha \in I$: $f(\alpha) = 0$. We study S_n in the case that all its arguments x_1, \dots, x_{n+2} are in the neighbourhood of α . We write $x = \alpha + y$ and $f_Y(y) = f(\alpha + y)$. The function f_Y is the function f in the coordinate frame Y . The root is at $y = 0$ in this coordinate frame: $f_Y(0) = 0$. We can express $f(x)$ in the second divided difference of f_Y as $f(x) = y f_Y[0, y]$.

Substituting $x_i = \alpha + y_i$ in (13) we have

$$\begin{aligned} S_n(\alpha + y_1, \dots, \alpha + y_{n+2}) \\ = \alpha + \left\{ \prod_{i=1}^{n+2} y_i \right\} K_{Yn}(y_1, \dots, y_{n+2}) \end{aligned} \quad (20)$$

with

$$\begin{aligned} K_{Yn}(y_1, \dots, y_{n+2}) \\ = \frac{\sum_{i=1}^{n+2} (1/f_Y[0, y_i]) a_{n,i}(y_1, \dots, y_{n+2})}{\sum_{i=1}^{n+2} \left(\prod_{j=1, j \neq i}^{n+2} y_j / f_Y[0, y_i] \right) a_{n,i}(y_1, \dots, y_{n+2})}. \end{aligned} \quad (21)$$

K_{Yn} is defined in the coordinate frame Y . The corresponding K_n in the original coordinate frame is

$$\begin{aligned} K_n(x_1, \dots, x_{n+2}) \\ = \frac{\sum_{i=1}^{n+2} (1/f[\alpha, x_i]) a_{n,i}(x_1, \dots, x_{n+2})}{\sum_{i=1}^{n+2} \left(\prod_{j=1, j \neq i}^{n+2} (x_j - \alpha) / f[\alpha, x_i] \right) a_{n,i}(x_1, \dots, x_{n+2})}. \end{aligned} \quad (22)$$

If we can show that K_{Yn} is bounded for small values of the y_i , we see from (20) that S_n is a good approximation of α . Establishing the boundedness of K_{Yn} is therefore the major task of the remainder of this section.

Application of Lemma 2 with $g(y) = 1/f_Y[0, y]$ in the numerator and $g(y) = (\prod_{i=1}^{n+2} y_i) / (y f_Y[0, y])$ in the denominator of (21) brings K_{Yn} in the form

$$\begin{aligned} K_{Yn}(y_1, \dots, y_{n+2}) \\ = \frac{(1/f_Y[0, y])[y_1, \dots, y_{n+2}]}{\left\{ \prod_{i=1}^{n+2} y_i \right\} (1/y f_Y[0, y])[y_1, \dots, y_{n+2}]}. \end{aligned} \quad (23)$$

The numerator is a divided difference of $1/f[0, y]$. The denominator contains a divided difference of $1/(y f[0, y])$. Call the denominator \mathcal{D} . We express the divided difference of $1/(y f[0, y])$ in divided differences of $1/f[0, y]$ with the use of Leibniz rule for divided differences [22–24]

$$\begin{aligned} \mathcal{D}(y_1, \dots, y_{n+2}) \\ = \left\{ \prod_{i=1}^{n+2} y_i \right\} \frac{1}{y f_Y[0, y]} [y_1, \dots, y_{n+2}] \\ = \left\{ \prod_{i=1}^{n+2} y_i \right\} \sum_{i=1}^{n+2} \frac{1}{y} [y_1, \dots, y_i] \frac{1}{f_Y[0, y]} [y_i, \dots, y_{n+2}]. \end{aligned} \quad (24)$$

Using

$$\frac{1}{y} [y_1, \dots, y_i] = (-1)^{i+1} \prod_{j=1}^i \frac{1}{y_j}, \quad (25)$$

we arrive at

$$\begin{aligned} \mathcal{D}(y_1, \dots, y_{n+2}) \\ = \sum_{i=1}^{n+2} (-1)^{i+1} \left\{ \prod_{j=i+1}^{n+2} y_j \right\} \frac{1}{f_Y[0, y]} [y_i, \dots, y_{n+2}]. \end{aligned} \quad (26)$$

We are now in a position to establish the properties of K_n in the following two lemmas.

Lemma 3. Let $I \subset \mathbb{R}$ be an open interval of real values and f a function $f: I \rightarrow \mathbb{R}$ with $f \in C^{n+1}(I)$. Let $f^{(n+1)}$ be Lipschitz continuous on I . Let $\alpha \in I$, $f(\alpha) = 0$, and $f^{(1)}(\alpha) \neq 0$. Then the n th order approximant S_n takes the form

$$\begin{aligned} S_n(x_1, \dots, x_{n+2}) \\ = \alpha + \left\{ \prod_{i=1}^{n+2} (x_j - \alpha) \right\} K_n(x_1, \dots, x_{n+2}), \end{aligned} \quad (27)$$

where there exists an $\epsilon > 0$ such that K_n is bounded if $|x_i - \alpha| < \epsilon$ for all $i = 1, \dots, n + 2$.

Proof. We have already shown the form of S_n in (20). All that remains to be done is to show that K_{Y_n} is bounded in an $(n + 2)$ -dimensional hypercube around the point $(y_1, \dots, y_{n+2}) = (0, \dots, 0)$. Denote this point by $\vec{0}$.

From the properties of divided differences we know that the numerator in (23) is bounded in a hypercube around $\vec{0}$ if the function $1/(f_Y[0, y])$ is n -times continuously differentiable around $y = 0$ and the n th derivative is Lipschitz continuous.

This is also a sufficient condition for all divided differences that appear in (26) to be bounded. Therefore all terms in (26) can be made arbitrarily small by choosing the y_i small enough, except for the last term. The last term has a limiting value $(-1)^{n+3}/f_Y^{(1)}(0) \neq 0$ if all y_i become equal to zero; compare or confer (35). This shows that there is a lower bound \underline{D} such that $0 < \underline{D} < |\mathcal{D}|$ on a hypercube around $\vec{0}$. Combining this with the bound on the numerator proves that K_{Y_n} is bounded.

It remains to be shown that $1/(f_Y[0, y])$ is n -times continuously differentiable with the n th order derivative Lipschitz continuous around $y = 0$.

Let J be the interval around $y = 0$ corresponding to the interval I around $x = \alpha$: $J = \{y \in \mathbb{R} \mid y + \alpha \in I\}$. Because $f \in C^{n+1}(I)$ and $f^{(n+1)}$ is Lipschitz continuous on I we have $f_Y \in C^{n+1}(J)$ and $f_Y^{(n+1)}$ is Lipschitz continuous on J .

Because $f^{(1)}(\alpha) \neq 0$ there is an finite open interval $J' \subset J$ with $0 \in J'$ such that $\inf\{|f_Y[0, y]| \mid y \in J'\} > 0$. Because $f_Y^{(n+1)}$ is Lipschitz continuous on J' and J' is finite, $f_Y^{(n+1)}$ is bounded on J' . Because $f_Y^{(n+1)}$ is bounded on J' , all lower-order derivatives $f_Y^{(k)}$ ($k = 1, \dots, n$) and f_Y itself will be bounded on J' .

First we show that $1/(f_Y[0, y])$ is n -times continuously differentiable on J' . We use [18]

$$\frac{\partial}{\partial y} f_Y \left[0, \underbrace{y, \dots, y}_{k \text{ times}} \right] = k f_Y \left[0, \underbrace{y, \dots, y}_{(k+1) \text{ times}} \right] \quad (28)$$

which is defined if f_Y is k -times differentiable in the point y and $y \neq 0$. If we allow $y = 0$ as well then f_Y must be $(k + 1)$ -times differentiable in y . Hence we find

$$\frac{\partial}{\partial y} \frac{1}{f_Y[0, y]} = -\frac{f_Y[0, y, y]}{f_Y[0, y]^2} \quad (29)$$

if $y \in J'$ (which includes $y = 0$) and $f_Y \in C^2(J')$, and

$$\begin{aligned} \frac{\partial^2}{\partial y^2} \frac{1}{f_Y[0, y]} \\ = 2 \frac{-f_Y[0, y, y, y] f_Y[0, y] + f_Y[0, y, y]^2}{f_Y[0, y]^3}. \end{aligned} \quad (30)$$

if $y \in J'$ and $f_Y \in C^3(J')$. Continuing the differentiation we easily see that $1/(f_Y[0, y]) \in C^n(J')$ if $f_Y \in C^{n+1}(J')$.

Next we show that the n th derivative to y of $1/(f_Y[0, y])$ is Lipschitz continuous on J' . As we see from examples (29) and (30), the derivative can be expressed as a nonlinear combination of $f_Y[0, \underbrace{y, \dots, y}_{k \text{ times}}]$ for $k = 1, \dots, n + 1$. Combine

the following to see that the derivative is Lipschitz continuous on J' .

(i) $f_Y[0, \underbrace{y, \dots, y}_{k \text{ times}}]$ is Lipschitz continuous on J' for $k =$

$1, \dots, n + 1$. To show this take $y_1, y_2 \in J'$ and consider the difference of the divided difference between the two points:

$$\begin{aligned} f_Y \left[0, \underbrace{y_1, \dots, y_1}_{k \text{ times}} \right] - f_Y \left[0, \underbrace{y_2, \dots, y_2}_{k \text{ times}} \right] \\ = \frac{1}{(k-1)!} \\ \cdot \left\{ \left(\frac{\partial^{k-1}}{\partial y^{k-1}} f_Y[0, y] \right)_{y=y_1} - \left(\frac{\partial^{k-1}}{\partial y^{k-1}} f_Y[0, y] \right)_{y=y_2} \right\} \quad (31) \\ = \frac{1}{(k-1)!} (y_1 - y_2) \left(\frac{\partial^k}{\partial y^k} f_Y[0, y] \right)_{y=\zeta} \\ = k (y_1 - y_2) f_Y \left[0, \underbrace{\zeta, \dots, \zeta}_{(k+1) \text{ times}} \right] \end{aligned}$$

for some $\zeta \in (\min(y_1, y_2), \max(y_1, y_2))$. Since $f_Y \in C^{(n+1)}(J')$ and $f_Y^{(n+1)}$ Lipschitz continuous on J' , $f_Y[0, \underbrace{\zeta, \dots, \zeta}_{(k+1) \text{ times}}]$ is bounded on J' for $k = 1, \dots, n + 1$.

(ii) Because $f_Y[0, y]$ is Lipschitz continuous on J' and the infimum of its absolute value on J' is larger than zero, $1/(f_Y[0, y])$ is Lipschitz continuous on J' .

(iii) If two functions f_1 and f_2 are Lipschitz continuous and bounded on J' , then $f_1 + f_2$ and $f_1 f_2$ are also Lipschitz continuous on J' . This is easy to see for the sum of the two functions. For the product we have

$$\begin{aligned} & |f_1(y_1) f_2(y_1) - f_1(y_2) f_2(y_2)| \\ & = |(f_1(y_1) - f_1(y_2)) f_2(y_1) - f_1(y_2) (f_2(y_2) - f_2(y_1))| \\ & \leq |f_2(y_1)| |f_1(y_1) - f_1(y_2)| \\ & \quad + |f_1(y_2)| |f_2(y_1) - f_2(y_2)| \\ & \leq M (L_1 + L_2) |y_1 - y_2|, \end{aligned} \quad (32)$$

where M is the maximum of the suprema of f_1 and f_2 on J' and L_i is the Lipschitz constant of f_i on J' .

This concludes the proof. \square

Lemma 4. Let $I \subset \mathbb{R}$ be an open interval of real values and f a function $f : I \rightarrow \mathbb{R}$ with $f \in C^{n+2}(I)$. Let $\alpha \in I$, $f(\alpha) = 0$,

and $f^{(1)}(\alpha) \neq 0$. Then $K_n(x_1, \dots, x_n)$ as defined in Lemma 3 is continuous in $(x_1, \dots, x_{n+2}) = (\alpha, \dots, \alpha)$ with

$$\begin{aligned} & \lim_{x_1, \dots, x_{n+2} \rightarrow \alpha} K_n(x_1, \dots, x_{n+2}) \\ &= \frac{(-1)^{n+1}}{(n+1)!} f^{(1)}(\alpha) \left(\frac{\partial^{n+1}}{\partial x^{n+1}} \frac{1}{f[\alpha, x]} \right)_{x=\alpha}. \end{aligned} \tag{33}$$

Proof. Taking the limit for the numerator of (23) we find

$$\begin{aligned} & \lim_{y_1, \dots, y_{n+2} \rightarrow 0} \frac{1}{f_Y[0, y]} [y_1, \dots, y_{n+2}] \\ &= \frac{1}{(n+1)!} \left(\frac{\partial^{n+1}}{\partial y^{n+1}} \frac{1}{f_Y[0, y]} \right)_{y=0} \\ &= \frac{1}{(n+1)!} \left(\frac{\partial^{n+1}}{\partial x^{n+1}} \frac{1}{f[\alpha, x]} \right)_{x=\alpha}. \end{aligned} \tag{34}$$

Taking the limit for the denominator \mathcal{D} in (26) we obtain

$$\lim_{y_1, \dots, y_{n+2} \rightarrow 0} \mathcal{D}(y_1, \dots, y_{n+2}) = \frac{(-1)^{n+1}}{f_Y[0, 0]} = \frac{(-1)^{n+1}}{f^{(1)}(\alpha)}. \tag{35}$$

Dividing the result for numerator by the result for the denominator yields the proof. \square

$$\begin{aligned} & x_{-1} \\ & x_0 \\ & x_{1,0} = S(x_0, x_{-1}) \\ & x_{2,0} = S(x_{1,0}, x_0) \quad x_{2,1} = S_1(x_{1,0}, x_0, x_{-1}) \\ & x_{3,0} = S(x_{2,1}, x_{1,0}) \quad x_{3,1} = S_1(x_{2,1}, x_{1,0}, x_0) \quad x_{3,2} = S_2(x_{2,1}, x_{1,0}, x_0, x_{-1}). \end{aligned} \tag{39}$$

5. The Algorithm

We construct an algorithm which generates a sequence of n th order approximants S_n . The algorithm starts with two initial approximants x_{-1} and x_0 of the root α . In the first iteration we simply carry out a Secant step:

$$\begin{aligned} & x_{-1} \\ & x_0 \\ & x_{1,0} = S(x_0, x_{-1}). \end{aligned} \tag{36}$$

The second iteration also starts with a Secant step:

$$\begin{aligned} & x_{-1} \\ & x_0 \\ & x_{1,0} = S(x_0, x_{-1}) \\ & x_{2,0} = S(x_{1,0}, x_0), \end{aligned} \tag{37}$$

but next we combine the two Secant steps in a first-order approximant using the iterative definition of the n th order approximants:

$$\begin{aligned} & x_{-1} \\ & x_0 \\ & x_{1,0} = S(x_0, x_{-1}) \\ & x_{2,0} = S(x_{1,0}, x_0) \quad x_{2,1} = S_1(x_{1,0}, x_0, x_{-1}). \end{aligned} \tag{38}$$

The third iteration first carries out a Secant step $x_{3,0}$, then combines this Secant step with the previous step $x_{2,0}$ in a first-order approximant $x_{3,1}$, and finally combines $x_{2,1}$ and $x_{3,1}$ in a second-order approximant $x_{3,2}$:

We continue this way with the fourth and the following iterations and generate a scheme which looks like

$$\begin{array}{ccccccc} & x_{-1} & & & & & \\ & x_0 & & & & & \\ & x_{1,0} & & & & & \\ & x_{2,0} & x_{2,1} & & & & \\ & x_{3,0} & x_{3,1} & x_{3,2} & & & \\ & x_{4,0} & x_{4,1} & x_{4,2} & x_{4,3} & & \\ & \vdots & \vdots & \vdots & \vdots & \ddots & \\ & x_{n,0} & x_{n,1} & \cdots & \cdots & \cdots & x_{n,n-1} \\ x_{n+1,0} & x_{n+1,1} & \cdots & \cdots & \cdots & x_{n+1,n-1} & x_{n+1,n} \\ x_{n+2,0} & x_{n+2,1} & \cdots & \cdots & \cdots & x_{n+2,n-1} & x_{n+2,n} \\ & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{array} \tag{40}$$

Since we aim at generating a sequence of n th order approximants, we calculate at most $n + 1$ columns in an iteration. The first iteration in which we calculate all $n + 1$ columns is the $(n + 1)$ th iteration.

If we parametrize x_{-1} as $x_{-1,0}$ and x_0 as $x_{0,0}$ we have parametrized all values in our scheme as $x_{p,i}$ with p running over the values $p = -1, 0, 1, \dots$ and i running over the values $i = 0, \dots, i_{\max}(p)$ with

$$i_{\max}(p) = \begin{cases} 0, & p = -1, 0, \\ p - 1, & p = 1, \dots, n, \\ n, & p > n. \end{cases} \quad (41)$$

For simplicity we will denote $x_{p,i_{\max}(p)}$ by $x_{p,i_{\max}}$. This means that, for example, $x_{p-1,i_{\max}}$ must be read as $x_{p-1,i_{\max}(p-1)}$.

The choice of n in (41) sets the order of the algorithm. Choosing $n = 0$ results in the Secant algorithm, choosing $n = 1$ results in the first-order accelerated Secant algorithm, $n = 2$ results in the second-order accelerated Secant algorithm, and so forth. An n th order accelerated Secant algorithm generates a sequence of n th order approximants.

The algorithm described above can be formulated as

$$x_{p,i} = \begin{cases} S(x_{p-1,i_{\max}}, x_{p-2,i_{\max}}), & i = 0, \\ \frac{x_{p-1,i-1}x_{p-1,i_{\max}} - x_{p-1,i}x_{p-2,i_{\max}}}{x_{p-1,i-1} + x_{p-1,i_{\max}} - x_{p-1,i} - x_{p-2,i_{\max}}}, & i = 1, \dots, i_{\max}(p), \end{cases} \quad (42)$$

with start values $x_{-1,0}$ and $x_{0,0}$.

Note that with $x_{p,0} = S(x_{p-1,i_{\max}}, x_{p-2,i_{\max}})$ we have

$$x_{p,1} = \frac{x_{p-1,0}x_{p-1,i_{\max}} - x_{p,0}x_{p-3,i_{\max}}}{x_{p-1,0} + x_{p-1,i_{\max}} - x_{p,0} - x_{p-3,i_{\max}}} \quad (43)$$

$$= S_1(x_{p-1,i_{\max}}, x_{p-2,i_{\max}}, x_{p-3,i_{\max}}),$$

and by recursion we easily show that

$$x_{p,i} = S_i(x_{p-1,i_{\max}}, \dots, x_{p-i-2,i_{\max}}) \quad (44)$$

for $i = 2, \dots, i_{\max}(p)$.

Each Secant step S after the first Secant step $x_{1,0} = S(x_0, x_{-1})$ requires exactly one evaluation of f . Namely, the calculation of $x_{p,0}$ requires the calculation $f(x_{p-1,i_{\max}})$ while $f(x_{p-2,i_{\max}})$ has already been calculated when we evaluated $x_{p-1,0}$. The calculation of $x_{p,i}$ for $i > 0$ does not require a calculation of f . Hence one iteration of the algorithm requires one evaluation of f , except for the first iteration which requires the evaluation of $f(x_{-1})$ and $f(x_0)$.

According to Brezinski and Zaglia [16] it is recommended to calculate the second line in (42) in the following form. It is mathematically equivalent to the second line in (42) but less susceptible to round-off errors according to [16]:

$$x_{p,i} = x_{p,i-1} + \frac{(x_{p-1,i_{\max}} - x_{p,i-1})(x_{p-1,i-1} - x_{p,i-1})}{x_{p-1,i-1} + x_{p-1,i_{\max}} - x_{p,i-1} - x_{p-i-2,i_{\max}}} \quad (45)$$

for $i = 1, \dots, i_{\max}(p)$. Alternative forms in which the leading term is $x_{p-1,i-1}$, $x_{p-1,i_{\max}}$, or $x_{p-i-2,i_{\max}}$ are also readily derived.

A pseudocode for the accelerated Secant algorithm is provided in Appendix B. Examples of sequences generated by this algorithm are given in the tables in Appendix C.

6. Convergence Properties

6.1. Basic Convergence. The following lemma establishes sufficient conditions under which the algorithm generates a convergent sequence.

Lemma 5. *Let $I \subset \mathbb{R}$ be an open interval of real values and f a function $f : I \rightarrow \mathbb{R}$ with $f \in C^{n+1}(I)$. Let $f^{(n+1)}$ be Lipschitz continuous on I . Let $\alpha \in I$, $f(\alpha) = 0$ and $f^{(1)}(\alpha) \neq 0$. Then there exists an $\epsilon > 0$ such that the sequence $\{x_{p,n}\}_{p=n+1}^{\infty}$ generated by the n th order accelerated Secant algorithm converges to α if the start values x_{-1} and x_0 are within a distance ϵ of α .*

Proof. We develop our proof in the coordinate frame Y . With $y_{p,i} = x_{p,i} - \alpha$ (44) reads in the coordinate frame Y :

$$y_{p,i} = S_{Y_i}(y_{p-1,i_{\max}}, \dots, y_{p-i-2,i_{\max}}) \quad (46)$$

with

$$S_{Y_i}(y_1, \dots, y_{i+2}) = S_i(\alpha + y_1, \dots, \alpha + y_{i+2}) - \alpha. \quad (47)$$

We have to prove that the sequence $\{y_{p,n}\}_{p=n+1}^{\infty}$ converges to zero if y_{-1} and y_0 are chosen close enough to zero.

Putting $i = i_{\max}$ in (46) we obtain a closed recursion for $y_{p,i_{\max}}$:

$$y_{p,i_{\max}} = S_{Y_{i_{\max}(p)}}(y_{p-1,i_{\max}}, \dots, y_{p-i-2,i_{\max}}). \quad (48)$$

For $p \geq n + 1$ we have $i_{\max}(p) = n$ and hence

$$y_{p,i_{\max}} = S_{Y_n}(y_{p-1,i_{\max}}, \dots, y_{p-n-2,i_{\max}}) \quad (49)$$

$$= \left\{ \prod_{j=1}^{n+2} y_{p-j,i_{\max}} \right\} K_{Y_n}(y_{p-1,i_{\max}}, \dots, y_{p-n-2,i_{\max}}).$$

Starting at $p = n + 1$ (49) generates a sequence $\{y_{p,n}\}_{p=n+1}^{\infty}$ from a set of start values $V = \{y_{-1,0}, y_{0,0}, y_{1,0}, y_{2,1}, y_{3,2}, \dots, y_{n,n-1}\} = \{y_{p,i_{\max}} \mid p = -1, \dots, n\}$. It must first be noted that all values in the set can be made arbitrarily small by choosing $y_{-1} = y_{-1,0}$ and $y_0 = y_{0,0}$ close enough to zero. We can see this for $y_{1,0}$ because $y_{1,0} = y_{-1}y_0K_{Y_1}(y_{-1}, y_0)$ and $|K_{Y_1}|$ is bounded in a 2-dimension volume around $(0, 0)$ according to Lemma 3. In the same way we can see this for $y_{2,1}$ because $y_{2,1} = y_{-1}y_0y_{1,0}K_{Y_2}(y_{-1}, y_0, y_{1,0})$ and $|K_{Y_2}|$ is bounded in a 3-dimension volume around $(0, 0, 0)$. Continuing the argument we find that all values in the set V can be made arbitrarily small.

Lemma 3 states that there is an $\epsilon > 0$ such that

$$|K_{Y_n}(y_1, \dots, y_{n+2})| < \bar{K} \quad (50)$$

if $|y_j| < \epsilon$ for all $j = 1, \dots, n+2$. Define the interval $J = \{y \in \mathbb{R} \mid |y| < \min(\epsilon, \bar{K}^{-1/(n+1)})\}$. Choose y_{-1} and y_0 small enough such that $y_{p,i_{\max}} \in J$ for all $y_{p,i_{\max}} \in V$.

From (49) with $p = n+1$ we see

$$\begin{aligned} & |y_{n+1,n}| \\ &= \left\{ \prod_{j=1}^{n+2} |y_{n+1-j,i_{\max}}| \right\} |K_{Yn}(y_{n,i_{\max}}, \dots, y_{-1,i_{\max}})| \\ &< \left\{ \prod_{j=1}^{n+2} |y_{n+1-j,i_{\max}}| \right\} \bar{K} \\ &< |y_{n,i_{\max}}| \left\{ \prod_{j=2}^{n+2} \bar{K}^{-1/(n+1)} \right\} \bar{K} = |y_{n,i_{\max}}| = |y_{n,n-1}|. \end{aligned} \quad (51)$$

Since $y_{n,n-1} \in J$ this guarantees that also $y_{n+1,n} \in J$. Repeating the argument recursively we see that $|y_{p+1,n}| < |y_{p,n}|$ for all $p \geq n+1$. This means that the sequence $\{|y_{p,n}\}_{p=n+1}^{\infty}$ is monotonically decreasing. Since it is bounded from below it must converge.

It remains to show that the sequence converges to zero. Suppose $|y_{p,n}| \rightarrow a > 0$ for $p \rightarrow \infty$. Since $|y_{p,n}| \in J$ and the sequence is decreasing we must have $a \in J$. From (49) we see $|K_{Yn}(y_{p-1,i_{\max}}, \dots, y_{p-n-2,i_{\max}})| \equiv K_p \rightarrow a^{-(n+1)}$ for $p \rightarrow \infty$. However $K_p < \bar{K} < a^{-(n+1)}$ where the last inequality follows from the fact $a \in J$. This is a contradiction with $K_p \rightarrow a^{-(n+1)}$ and therefore we must have $a = 0$. \square

6.2. Order of Convergence. Our main result regarding the convergence of the sequence generated by the algorithm is the following theorem.

Theorem 6. *Let $I \subset \mathbb{R}$ be an open interval of real values and f a function $f: I \rightarrow \mathbb{R}$ with $f \in C^{n+2}(I)$. Let $\alpha \in I$, $f(\alpha) = 0$, and $f^{(1)}(\alpha) \neq 0$. Define*

$$K_{\infty n} = \frac{(-1)^{n+1}}{(n+1)!} f^{(1)}(\alpha) \left(\frac{\partial^{n+1}}{\partial x^{n+1}} \frac{1}{f[\alpha, x]} \right)_{x=\alpha} \quad (52)$$

and let $K_{\infty n} \neq 0$. Then there is an $\epsilon > 0$ such that the sequence $\{x_{p,n}\}_{p=n+1}^{\infty}$ generated by the n th order accelerated Secant algorithm converges to α if the start values x_{-1} and x_0 are within a distance ϵ of α . The sequence converges with order ψ_n :

$$\lim_{p \rightarrow \infty} \frac{|x_{p,n} - \alpha|}{|x_{p-1,n} - \alpha|^{\psi_n}} = |K_{\infty n}|^{(\psi_n - 1)/(n+1)}, \quad (53)$$

where ψ_n is the real, positive solution of $t^{n+2} - \sum_{i=0}^{n+1} t^i = 0$.

Proof. First of all note that if $f \in C^{n+2}(I)$ then $f \in C^{n+1}(J)$ with $f^{(n+1)}$ Lipschitz continuous on J for any finite interval $J \subset I$. Convergence of the sequence $\{x_{p,n}\}_{p=n+1}^{\infty}$ is therefore guaranteed by Lemma 5.

Reverting to the coordinate frame Y , we see from (49) that

$$y_{p,n} = \left\{ \prod_{j=1}^{n+2} y_{p-j,n} \right\} K_{p,n} \quad (54)$$

with

$$K_{p,n} = K_{Yn}(y_{p-1,n}, \dots, y_{p-n-2,n}) \quad (55)$$

for $p \geq 2n+3$. Since $\lim_{p \rightarrow \infty} y_{p,n} = 0$ we have from Lemma 4 that $\lim_{p \rightarrow \infty} K_{p,n} = K_{\infty n}$. Therefore we satisfy the conditions of Theorem 3-3 in [2]. This theorem proves the result for the order of convergence and the asymptotic error term. \square

Note that the theorem encompasses the convergence property (9) of the Secant sequence for $n = 0$.

The sequence $\{\psi_n\}_{n=0}^{\infty}$ is monotonically increasing and converges to 2 [2, 13]. As examples we have $\psi_0 = (1 + \sqrt{5})/2 \approx 1.6180$, $\psi_1 \approx 1.8393$, $\psi_2 \approx 1.9276$, and $\psi_3 \approx 1.9659$.

If we define

$$C_n = \frac{f^{(n)}(\alpha)}{n! f^{(1)}(\alpha)} \quad (56)$$

we have as examples for $K_{\infty n}$

$$\begin{aligned} K_{\infty 0} &= C_2, \\ K_{\infty 1} &= -C_3 + C_2^2, \\ K_{\infty 2} &= C_4 - 2C_2C_3 + C_2^3. \end{aligned} \quad (57)$$

7. Conclusions

We have devised an accelerated Secant algorithm which requires one function evaluation per iteration, does not require the evaluation of a derivative, and can achieve an order of convergence arbitrarily close to two. As such the algorithm is an alternative for the method of inverse interpolation [2] or for Sidi's method [13]. The accelerated algorithm is not much more complicated than the original Secant algorithm.

The algorithm is formulated in (42) but can also be formulated as follows. Start with initial estimates x_{-1}, \dots, x_n and generate the sequence as $x_{k+1} = S_n(x_k, \dots, x_{k-n})$ with S_n calculated as in, for example, (13) or (19). This formulation shows that the algorithm is a "one-point iteration function with memory" in the classification of Traub [2]. The form (42) seems easier to implement though and requires only two initial estimates.

It is possible that our approach can be applied to other algorithms than the Secant algorithm. We think in particular of algorithms generating a sequence $\{x_n\}$ with the property $(x_n - \alpha) = K_n \prod_{i=1}^m (x_{n-i} - \alpha)$.

We also note that the our algorithm appears to lend itself for parallelization, which is not obvious for algorithms based on polynomial fits. Multiple evaluations of f per iteration result in larger orders of convergence [2, 25–28]. It is not hard to see that our first-order algorithm can be modified into

a faster converging algorithm in case where two calculation cores are available. It remains to be investigated what the order of convergence is (assuming an order is defined) and whether or not this can be generalized to an arbitrary number of cores.

We have only studied the convergent behaviour of the subsequence $\{x_{p,n}\}_{p=n+1}^{\infty}$ of the n th order algorithm. One may wonder about the subsequences $\{x_{p,i}\}_{p=i+1}^{\infty}$ for $i < n$. A study of the first- and second-order accelerated versions of the algorithm [29] revealed that they converge with the same order as $\{x_{p,n}\}_{p=n+1}^{\infty}$ but with a different asymptotic error term. This is likely the case for all orders of the algorithm.

A different approach to judge the efficiency of the algorithm is to estimate the average computational cost of the algorithm by statistical means [30]. In this approach one averages the cost over a set of functions with a suitable probability measure. Although interesting, such a study is outside of the scope of the current article.

Appendices

A. The Approximant Directly Expressed in f

We prove that S_n takes the form (13). We have seen that it takes this form for $n = 0$ (11) and $n = 1$ (12) and we prove it for $n > 1$ by induction.

Suppose (13) is true for S_n . We calculate S_{n+1} from the recursive definition $S_{n+1} = \mathcal{N}/\mathcal{D}$ with

$$\mathcal{N} = S_n(x_1, \dots, x_{n+2})x_{n+3} - x_1 S_n(x_2, \dots, x_{n+3}), \quad (\text{A.1})$$

$$\begin{aligned} \mathcal{D} &= S_n(x_1, \dots, x_{n+2}) + x_{n+3} - x_1 \\ &\quad - S_n(x_2, \dots, x_{n+3}). \end{aligned} \quad (\text{A.2})$$

The calculation will show that they take the form

$$\mathcal{N} = \frac{F_1}{F_2} \sum_{i=1}^{n+3} \frac{x_i}{f_i} a_{n+1,i}(x_1, \dots, x_{n+3}) \quad (\text{A.3})$$

$$\mathcal{D} = \frac{F_1}{F_2} \sum_{i=1}^{n+3} \frac{1}{f_i} a_{n+1,i}(x_1, \dots, x_{n+3})$$

with

$$\begin{aligned} F_1 &= \frac{1}{(x_1 - x_{n+3}) \prod_{i=2}^{n+2} (x_1 - x_i) (x_i - x_{n+3})} \\ &\quad \times \sum_{j=2}^{n+2} \frac{(x_1 - x_j)(x_j - x_{n+3})}{f_j} a_{n+1,j}(x_1, \dots, x_{n+3}), \\ F_2 &= \sum_{i=1}^{n+2} \sum_{j=2}^{n+3} \frac{1}{f_i f_j} a_{n,i}(x_1, \dots, x_{n+2}) a_{n,j-1}(x_2, \dots, x_{n+3}). \end{aligned} \quad (\text{A.4})$$

Taking the ratio \mathcal{N}/\mathcal{D} completes the proof.

```

define imax(p)
  if p = -1 or p = 0
    return 0
  else if p ≤ n
    return p - 1
  else
    return n

```

ALGORITHM 1

Throughout this appendix we use the following identities:

$$\begin{aligned} &\left\{ \prod_{j=2}^{n+2} (x_j - x_{n+3}) \right\} a_{n,i}(x_1, \dots, x_{n+2}) \\ &= \frac{x_i - x_{n+3}}{x_1 - x_{n+3}} a_{n+1,i}(x_1, \dots, x_{n+3}) \end{aligned} \quad (\text{A.5})$$

which holds for $i = 1, \dots, n+2$, and

$$\begin{aligned} &\left\{ \prod_{j=2}^{n+2} (x_1 - x_j) \right\} a_{n,i-1}(x_2, \dots, x_{n+3}) \\ &= -\frac{x_1 - x_i}{x_1 - x_{n+3}} a_{n+1,i}(x_1, \dots, x_{n+3}) \end{aligned} \quad (\text{A.6})$$

which holds for $i = 2, \dots, n+3$. If we take $i = 1$ in (A.5) and $i = n+3$ in (A.6) we obtain in particular

$$\begin{aligned} &\left\{ \prod_{j=2}^{n+2} (x_j - x_{n+3}) \right\} a_{n,1}(x_1, \dots, x_{n+2}) \\ &= a_{n+1,1}(x_1, \dots, x_{n+3}), \end{aligned} \quad (\text{A.7})$$

$$\begin{aligned} &\left\{ \prod_{j=2}^{n+2} (x_1 - x_j) \right\} a_{n,n+2}(x_2, \dots, x_{n+3}) \\ &= -a_{n+1,n+3}(x_1, \dots, x_{n+3}). \end{aligned}$$

For a more compact notation we will write $a_{n,i}(x_1, \dots, x_{n+2}) \equiv a_{n,i;1,\dots,n+2}$ in the remainder of this appendix. We prove (A.3) for the numerator \mathcal{N} . The result for the denominator \mathcal{D} can be obtained following the same steps.

A.1. Numerator. We start with the numerator \mathcal{N} . Inserting (13) in (A.1) we obtain

$$\begin{aligned} \mathcal{N} &= S_n(x_1, \dots, x_{n+2})x_{n+3} - x_1 S_n(x_2, \dots, x_{n+3}) \\ &= \frac{1}{F_2} \left\{ x_{n+3} \sum_{i=1}^{n+2} \frac{x_i}{f_i} a_{n,i;1,\dots,n+2} \sum_{j=2}^{n+3} \frac{1}{f_j} a_{n,j-1;2,\dots,n+3} \right. \\ &\quad \left. - x_1 \sum_{i=2}^{n+3} \frac{x_i}{f_i} a_{n,i-1;2,\dots,n+3} \sum_{j=1}^{n+2} \frac{1}{f_j} a_{n,j;1,\dots,n+2} \right\} \end{aligned} \quad (\text{A.8})$$

```

define x(p, i)
  if p < -1 or i < 0 or i > imax(p)
    return error
  else if p = -1
    return x-1
  else if p = 0
    return x0
  else
    if i = 0
      x1 = x(p - 1, imax(p - 1))
      x2 = x(p - 2, imax(p - 2))
      return x1 - f(x1) * (x1 - x2) / (f(x1) - f(x2))
    else
      x1 = x(p - 1, i - 1)
      x2 = x(p - 1, imax(p - 1))
      x3 = x(p, i - 1)
      x4 = x(p - i - 2, imax(p - i - 2))
      return x3 + ((x2 - x3)(x1 - x3)) / (x1 + x2 - x3 - x4)

```

ALGORITHM 2

which we write as

$$\begin{aligned}
& \frac{1}{F_2} \left\{ \sum_{i=1}^{n+2} \sum_{j=2}^{n+3} \frac{x_i x_{n+3}}{f_i f_j} a_{n,i;1,\dots,n+2} a_{n,j-1;2,\dots,n+3} \right. \\
& \quad \left. - \sum_{i=1}^{n+2} \sum_{j=2}^{n+3} \frac{x_1 x_j}{f_i f_j} a_{n,i;1,\dots,n+2} a_{n,j-1;2,\dots,n+3} \right\} \\
&= \frac{1}{F_2} \left\{ \sum_{i=1}^{n+2} \sum_{j=2}^{n+3} \frac{x_i x_{n+3} - x_1 x_j}{f_i f_j} a_{n,i;1,\dots,n+2} a_{n,j-1;2,\dots,n+3} \right\} \\
&\equiv \frac{T}{F_2}.
\end{aligned} \tag{A.9}$$

We write T in (A.9) as

$$\begin{aligned}
T &= \sum_{i=2}^{n+2} \frac{x_1 x_{n+3} - x_1 x_i}{f_1 f_i} a_{n,1;1,\dots,n+2} a_{n,i-1;2,\dots,n+3} \\
&+ \sum_{i=2}^{n+2} \frac{x_i x_{n+3} - x_1 x_{n+3}}{f_i f_{n+3}} a_{n,i;1,\dots,n+2} a_{n,n+2;2,\dots,n+3} \\
&+ \sum_{i=2}^{n+2} \sum_{j=2}^{n+2} \frac{x_i x_{n+3} - x_1 x_j}{f_i f_j} a_{n,i;1,\dots,n+2} a_{n,j-1;2,\dots,n+3} \\
&\equiv T_1 + T_2 + T_3
\end{aligned} \tag{A.10}$$

with

$$\begin{aligned}
T_1 &= -\frac{x_1}{f_1} a_{n,1;1,\dots,n+2} \sum_{j=2}^{n+2} \frac{x_j - x_{n+3}}{f_j} a_{n,j-1;2,\dots,n+3} \\
&= -\frac{1}{\prod_{k=2}^{n+2} (x_1 - x_k) (x_k - x_{n+3})}
\end{aligned}$$

$$\begin{aligned}
& \times \frac{x_1}{f_1} \left\{ \prod_{k=2}^{n+2} (x_k - x_{n+3}) \right\} a_{n,1;1,\dots,n+2} \\
& \times \sum_{j=2}^{n+2} \frac{x_j - x_{n+3}}{f_j} \left\{ \prod_{k=2}^{n+2} (x_1 - x_k) \right\} a_{n,j-1;2,\dots,n+3} \\
&= \frac{1}{(x_1 - x_{n+3}) \prod_{k=2}^{n+2} (x_1 - x_k) (x_k - x_{n+3})} \\
& \times \frac{x_1}{f_1} a_{n+1,1;1,\dots,n+3} \\
& \times \sum_{j=2}^{n+2} \frac{(x_1 - x_j) (x_j - x_{n+3})}{f_j} a_{n+1,j;1,\dots,n+3} \\
&= F_1 \frac{x_1}{f_1} a_{n+1,1;1,\dots,n+3} \\
T_2 &= -\frac{x_{n+3}}{f_{n+3}} a_{n,n+2;2,\dots,n+3} \sum_{j=2}^{n+2} \frac{x_1 - x_j}{f_j} a_{n,j;1,\dots,n+2} \\
&= -\frac{1}{\prod_{k=2}^{n+2} (x_1 - x_k) (x_k - x_{n+3})} \\
& \times \frac{x_{n+3}}{f_{n+3}} \left\{ \prod_{k=2}^{n+2} (x_1 - x_k) \right\} a_{n,n+2;2,\dots,n+3} \\
& \times \sum_{j=2}^{n+2} \frac{x_1 - x_j}{f_j} \left\{ \prod_{k=2}^{n+2} (x_k - x_{n+3}) \right\} a_{n,j;1,\dots,n+2} \\
&= \frac{1}{(x_1 - x_{n+3}) \prod_{k=2}^{n+2} (x_1 - x_k) (x_k - x_{n+3})} \\
& \times \frac{x_{n+3}}{f_{n+3}} a_{n+1,n+3;1,\dots,n+3}
\end{aligned}$$

TABLE 1: Example of the convergence properties of the first-order accelerated Secant algorithm. The two subsequences $\{x_{p,i}\}$ ($i = 0, 1$) are shown in the columns 2 and 3. The convergence properties for the subsequence $\{x_{p,1}\}$ are shown in the last column. The ratio $|x_{p,1}|/|x_{p-1,1}|^{\psi_1}$ is expected to converge to $|K_{\infty}|^{(\psi_1-1)/2} \approx 1.5857$; compare or confer Theorem 6.

P	$x_{p,0}$	$x_{p,1}$	$\frac{ x_{p,1} }{ x_{p-1,1} ^{\psi_1}}$
-1	-0.1		
0	0.1		
1	1.9900×10^{-2}		
2	-4.8788×10^{-3}	-6.3502×10^{-4}	
3	2.6028×10^{-5}	-4.6934×10^{-6}	3.5652
4	-5.9550×10^{-9}	1.8383×10^{-10}	1.1617
5	1.7255×10^{-15}	1.6419×10^{-18}	1.3240
6	-6.0365×10^{-28}	-4.2497×10^{-33}	2.1850
7	1.3955×10^{-50}	-3.8480×10^{-60}	1.3365
8	-3.2706×10^{-92}	8.0549×10^{-110}	1.5376
9	6.1990×10^{-169}	3.9516×10^{-201}	1.7857
10	-6.3660×10^{-310}	-3.6744×10^{-369}	1.4594
11	2.9040×10^{-569}	-3.5087×10^{-678}	1.5937
12	$-2.5785 \times 10^{-1046}$	1.5284×10^{-1246}	1.6519
13	1.0725×10^{-1923}	5.9115×10^{-2292}	1.5280
14	$-1.8070 \times 10^{-3537}$	$-9.5104 \times 10^{-4215}$	1.5998
15	1.1244×10^{-6505}	$-2.5778 \times 10^{-7751}$	1.6060
16	$-4.9032 \times 10^{-11965}$	$4.3478 \times 10^{-14256}$	1.5613
17	$2.2415 \times 10^{-22006}$	$3.1977 \times 10^{-26220}$	1.5954
18	$-2.7806 \times 10^{-40475}$	$-1.0752 \times 10^{-48225}$	1.5910
19	$6.8762 \times 10^{-74445}$	$-4.4844 \times 10^{-88700}$	1.5761
20	$-9.6430 \times 10^{-136925}$	$4.6253 \times 10^{-163144}$	1.5910
21	$4.1483 \times 10^{-251843}$	$6.6903 \times 10^{-300068}$	1.5865
22	$-6.1889 \times 10^{-463211}$	$-4.1630 \times 10^{-551910}$	1.5821
23	$5.5704 \times 10^{-851977}$	$-3.8647 \times 10^{-1015120}$	1.5882
24	$-3.2178 \times 10^{-1567029}$	$3.2292 \times 10^{-1867096}$	1.5855
25	$2.4959 \times 10^{-2882215}$	$1.5586 \times 10^{-3434124}$	1.5845
26	$-1.0066 \times 10^{-5301219}$	$-5.8353 \times 10^{-6316339}$	1.5868

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

$$\begin{aligned}
 & \times \sum_{j=2}^{n+2} \frac{(x_1 - x_j)(x_j - x_{n+3})}{f_j} a_{n+1,j;1,\dots,n+3} && \times \left\{ \prod_{k=2}^{n+2} (x_1 - x_k) \right\} a_{n,j-1;2,\dots,n+3} \\
 & = F_1 \frac{x_{n+3}}{f_{n+3}} a_{n+1,n+3;1,\dots,n+3}, && = -\frac{1}{(x_1 - x_{n+3})^2 \prod_{k=2}^{n+2} (x_1 - x_k) (x_k - x_{n+3})} \\
 & && \times \sum_{i=2}^{n+2} \sum_{j=2}^{n+2} \frac{T_{ij}}{f_i f_j} a_{n,i;1,\dots,n+3} a_{n,j;1,\dots,n+3} \\
 T_3 & = \sum_{i=2}^{n+2} \sum_{j=2}^{n+2} \frac{x_i x_{n+3} - x_1 x_j}{f_i f_j} a_{n,i;1,\dots,n+2} a_{n,j-1;2,\dots,n+3} && (A.11) \\
 & = \frac{1}{\prod_{k=2}^{n+2} (x_1 - x_k) (x_k - x_{n+3})} && (A.12) \\
 & \times \sum_{i=2}^{n+2} \sum_{j=2}^{n+2} \frac{x_i x_{n+3} - x_1 x_j}{f_i f_j} && \\
 & \times \left\{ \prod_{k=2}^{n+2} (x_k - x_{n+3}) \right\} a_{n,i;1,\dots,n+2} && \\
 & \text{with} && \\
 & T_{ij} = (x_i x_{n+3} - x_1 x_j) (x_i - x_{n+3}) (x_1 - x_j). && (A.13) \\
 & \text{We can replace } T_{ij} \text{ by } (T_{ij} + T_{ji})/2 \text{ under the double sum:} && \\
 & T_{ij} \longrightarrow \frac{1}{2} (T_{ij} + T_{ji}) &&
 \end{aligned}$$

TABLE 2: Example of the convergence properties of the second-order accelerated Secant algorithm. The three subsequences $\{x_{p,i}\}$ ($i = 0, 1, 2$) are shown in the columns 2, 3 and 4. The convergence properties for the subsequence $\{x_{p,2}\}$ are shown in the last column. The ratio $|x_{p,2}|/|x_{p-1,2}|^{\psi_2}$ is expected to converge to $|K_{\text{co2}}|^{(\psi_2-1)/3} \approx 1.6448$; compare or confer Theorem 6.

p	$f(x) = \frac{x(x^2 + x - 1)}{x + 1}$			
	$x_{p,0}$	$x_{p,1}$	$x_{p,2}$	$\frac{ x_{p,2} }{ x_{p-1,2} ^{\psi_2}}$
-1	-0.1			
0	0.1			
1	1.9900×10^{-2}			
2	-4.8788×10^{-3}	-6.3502×10^{-4}		
3	2.6028×10^{-5}	-4.6934×10^{-6}	-6.6938×10^{-7}	
4	-8.4933×10^{-10}	2.6218×10^{-11}	-5.1981×10^{-12}	4.1423
5	-6.9591×10^{-18}	-6.6217×10^{-21}	2.2686×10^{-22}	1.2784
6	2.3585×10^{-33}	2.3681×10^{-39}	2.5037×10^{-42}	1.3159
7	-1.1360×10^{-63}	-8.8573×10^{-75}	-9.8816×10^{-81}	1.5280
8	4.9481×10^{-122}	-1.6838×10^{-143}	-1.4587×10^{-154}	2.3928
9	-2.8829×10^{-234}	1.0827×10^{-275}	-4.0936×10^{-297}	1.3822
10	-1.1943×10^{-450}	-1.7702×10^{-530}	7.3869×10^{-572}	1.4945
11	6.0478×10^{-868}	1.3233×10^{-1021}	2.1794×10^{-1101}	1.6974
12	$-3.2198 \times 10^{-1672}$	$-1.9771 \times 10^{-1968}$	$-4.8068 \times 10^{-2122}$	1.8855
13	2.0952×10^{-3222}	$-2.3216 \times 10^{-3793}$	$-1.5839 \times 10^{-4089}$	1.4857
14	$-1.5227 \times 10^{-6210}$	4.9781×10^{-7311}	$-6.1288 \times 10^{-7882}$	1.5969
15	$-1.9415 \times 10^{-11970}$	$-1.3999 \times 10^{-14091}$	$5.0849 \times 10^{-15192}$	1.7063
16	$6.2328 \times 10^{-23073}$	$1.4809 \times 10^{-27161}$	$1.1864 \times 10^{-29282}$	1.7153
17	$-1.2065 \times 10^{-44473}$	$-1.1092 \times 10^{-52354}$	$-2.9281 \times 10^{-56443}$	1.5606
18	$6.9477 \times 10^{-85725}$	$-5.2992 \times 10^{-100916}$	$-5.4130 \times 10^{-108797}$	1.6392
19	$-3.1699 \times 10^{-165239}$	$5.6411 \times 10^{-194521}$	$-4.7807 \times 10^{-209712}$	1.6827
20	$-5.1756 \times 10^{-318508}$	$-2.2732 \times 10^{-374950}$	$4.4948 \times 10^{-404232}$	1.6594
21	$4.2977 \times 10^{-613943}$	$3.4895 \times 10^{-722739}$	$1.7029 \times 10^{-779181}$	1.6052
22	$-1.5309 \times 10^{-1183412}$	$-1.0978 \times 10^{-1393123}$	$-9.9039 \times 10^{-1501920}$	1.6511
23	$3.3731 \times 10^{-2281100}$	$-2.2742 \times 10^{-2685331}$	$-1.8121 \times 10^{-2895042}$	1.6631

$$\begin{aligned}
&= -\frac{1}{2}(x_1 - x_{n+3}) \\
&\cdot \left\{ x_i(x_1 - x_j)(x_j - x_{n+3}) + x_j(x_1 - x_i)(x_i - x_{n+3}) \right\}. \\
&\quad (A.14)
\end{aligned}
\quad \times \left\{ \sum_{j=2}^{n+2} \frac{(x_1 - x_j)(x_j - x_{n+3})}{f_j} a_{n,j;1,\dots,n+3} \right\}$$

$$= F_1 \sum_{i=2}^{n+2} \frac{x_i}{f_i} a_{n,i;1,\dots,n+3}. \quad (A.16)$$

The two terms will give the same result under the double sum; hence we can stay with the first and substitute

$$T_{ij} \longrightarrow -(x_1 - x_{n+3}) x_i (x_1 - x_j) (x_j - x_{n+3}) \quad (A.15)$$

in (A.12):

$$\begin{aligned}
T_3 &= \frac{1}{(x_1 - x_{n+3}) \prod_{k=2}^{n+2} (x_1 - x_k) (x_k - x_{n+3})} \\
&\times \left\{ \sum_{i=2}^{n+2} \frac{x_i}{f_i} a_{n,i;1,\dots,n+3} \right\}
\end{aligned}$$

Adding T_1 , T_2 , and T_3 we obtain

$$T = F_1 \sum_{i=1}^{n+3} \frac{x_i}{f_i} a_{n,i;1,\dots,n+3} \quad (A.17)$$

substituting (A.17) in (A.9) results in the expression for the numerator \mathcal{N} in (A.3).

B. Pseudocode

The n th order accelerated Secant algorithm looks as follows in pseudocode. The algorithm calculates the approximant $x_{p,i}$ of the root of a function f . The initial estimates are x_{-1} and x_0 .

We define $i_{\max,p}$ as in (41) (see Algorithm 1), and we define $x_{p,i}$ as in (42) (see Algorithm 2).

C. Numerical Example

A numerical example for the accelerated Secant method is given in Tables 1 and 2. Both tables have been calculated with Algorithms 1 and 2.

Table 1 shows how the root α of a function f is approximated by the first-order accelerated Secant algorithm. Table 2 does the same for the second-order accelerated Secant algorithm. Both tables use the function

$$\begin{aligned} f(x) &= \frac{x(x^2 + x - 1)}{x + 1} = \frac{1}{x + 1} + x^2 - 1 \\ &= -x + 2x^2 - x^3 + x^4 \\ &\quad - x^5 + x^6 - \dots, \quad |x| < 1 \end{aligned} \quad (\text{C.1})$$

which has $\alpha = 0$, $f^{(1)}(0) = -1$, $C_2 = -2$, and $C_n = (-1)^{n+1}$ for $n = 3, 4, 5, \dots$, $K_{\infty 1} = 3$, and $K_{\infty 2} = -5$.

Conflict of Interests

The author declares that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

The author thanks Dr. A. Hogervorst for his kind interest and encouragement. All numerical results have been obtained with the help of the open-source program Sage (<http://www.sagemath.org/>). Some of the listed references have been brought to my attention by the referees. Taylor and Francis publishers kindly provided a copy of [12].

References

- [1] D. E. Muller, "A method for solving algebraic equations using an automatic computer," *Mathematical Tables and Other Aids to Computation*, vol. 10, pp. 208–215, 1956.
- [2] J. Traub, *Iterative Methods for the Solution of Equations*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1964.
- [3] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *The Computer Journal*, vol. 14, pp. 422–425, 1971.
- [4] R. L. Burden, J. D. Faires, and A. C. Reynolds, *Numerical Analysis*, Weber and Schmidt, 2nd edition, 1981.
- [5] J. Stoer and R. Bulirsch, *Introduction to Numerical Analysis*, Springer, 3rd edition, 2002.
- [6] A. Sidi, "Unified treatment of regula falsi, Newton-Raphson, secant, and Steffensen methods for nonlinear equations," *Journal of Online Mathematics and its Applications*, vol. 6, pp. 1–13, 2006.
- [7] J. F. Epperson, *An Introduction to Numerical Methods and Analysis*, Wiley-Interscience, 2007.
- [8] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, 3rd edition, 2007.
- [9] A. Iliev and N. Kyurkchiev, *Nontrivial Methods in Numerical Analysis: Selected Topics in Numerical Analysis*, LAP LAMBERT Academic Publishing, 2010.
- [10] M. Raydan, "Exact order of convergence of the secant method," *Journal of Optimization Theory and Applications*, vol. 78, no. 3, pp. 541–551, 1993.
- [11] G. E. Alefeld and F. A. Potra, "Some efficient methods for enclosing simple zeros of nonlinear equations," *BIT Numerical Mathematics*, vol. 32, no. 2, pp. 334–344, 1992.
- [12] R. J. Schmidt, "On the numerical solution of linear simultaneous equations by an iterative method," *Philosophical Magazine*, vol. 32, pp. 369–383, 1941.
- [13] A. Sidi, "Generalization of the secant method for nonlinear equations," *Applied Mathematics E-Notes*, vol. 8, pp. 115–123, 2008.
- [14] R. W. Hakala, "Acceleration of the convergence of the secant method," *Proceedings of the Oklahoma Academy of Science*, vol. 48, pp. 127–128, 1968.
- [15] W. Han and F. A. Potra, "Convergence acceleration for some root finding methods," *Computing Supplementum*, vol. 9, pp. 67–78, 1993.
- [16] C. Brezinski and M. R. Zaglia, "Generalizations of Aitken's process for accelerating the convergence of sequences," *Computational and Applied Mathematics*, vol. 26, no. 2, pp. 171–189, 2007.
- [17] D. Shanks, "Non-linear transformations of divergent and slowly convergent sequences," *Journal of Mathematics and Physics*, vol. 34, pp. 1–42, 1955.
- [18] F. B. Hildebrand, *Introduction to Numerical Analysis*, Dover, 2nd edition, 1987.
- [19] A. Ralston and P. Rabinowitz, *A First Course in Numerical Analysis*, Dover, 2nd edition, 2011.
- [20] L. Milne-Thomson, *The Calculus of Finite Differences*, AMS Chelsea Publishing, 2nd edition, 2000.
- [21] P. Dez, "A note on the convergence of the secant method for simple and multiple roots," *Applied Mathematics Letters*, vol. 16, no. 8, pp. 1211–1215, 2003.
- [22] J. Steffensen, "Note on divided differences," *Kgl Danske Videnskabernes Selskab: Matematisk-Fysiske Meddelelser*, vol. 17, pp. 1–12, 1939.
- [23] T. Popoviciu, "Introduction à la théorie des différences divisées," *Bulletin de la Société Roumaine de Sciences*, vol. 42, pp. 65–78, 1940.
- [24] M. S. Floater and T. Lyche, "Divided differences of inverse functions and partitions of a convex polygon," *Mathematics of Computation*, vol. 77, no. 264, pp. 2295–2308, 2008.
- [25] S. Amat and S. Busquier, "On a higher order secant method," *Applied Mathematics and Computation*, vol. 141, no. 2-3, pp. 321–329, 2003.
- [26] A. Cordero, J. L. Hueso, E. Martínez, and J. R. Torregrosa, "A family of derivative-free methods with high order of convergence and its application to nonsmooth equations," *Abstract and Applied Analysis*, vol. 2012, Article ID 836901, 15 pages, 2012.
- [27] B. Ignatova, N. Kyurkchiev, and A. Iliev, "Multipoint algorithms arising from optimal in the sense of Kung- Traub iterative procedures for numerical solution of nonlinear equations," *General Mathematics Notes*, vol. 6, no. 2, pp. 45–79, 2011.
- [28] N. Anderson and A. Bjorck, "A new high order method of regula falsi type for computing a root of an equation," *BIT*, vol. 13, pp. 253–264, 1973.

- [29] M. Nijmeijer, unpublished, 2013.
- [30] E. Novak, K. Ritter, and H. Wozniakowski, "Average-case optimality of a hybrid secant-bisection method," *Mathematics of Computation*, vol. 64, no. 212, pp. 1517–1539, 1995.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

