Journal of Taiwan Normal University: Mathematics, Science & Technology
1997, Vol. 42 (New Version), 1-6

The Derivation of Two Parallel Zero-Finding Algorithms of Polynomials

Tai-Yih Tso

Department of Mathmatics
National Taiwan Normal University

In this paper we study the derivation of two famous algorithms for finding all zeros of a giving polynomial. These two algorithms which are the Weierstrass method and the Aberth method are highly suited for parallel computing. It is explained that both of the two algorithms can be arrived by the functional iteration analysis. We also show that the the Weierstrass method and the Aberth method can be derived by the fixed point iteration method and the Newton method, respectively, together with the implicit deflation scheme.

Keyword: parallel computing functional iteration analysis zeros-finding implicit deflation

Introduction

The problem of finding the zeros of a polynomial is a genuine nonlinear problem. It is a fundamental problem in mathematics but not solved perfectly. There are many numerical algorithms which have been developed for solving this problem. However, only few implementations are realizable and exploited. These algorithms may be classified as three main methods: determining one zero at a time, computing the eigenvalues of the companion matrix associated with the polynomial, and finding the all zeros simultaneously.

Classical algorithms determine one zero at a time and then use deflation step to remove the calculated zero from the polynomial for determining the further zeros. Since the calculated zero is an approximation of a zero, the error will be accumulated during the deflation step.

The power method which applies to the Frobenius companion matrix associated with the polynomial is equivalent to the Bernoulli method for finding the zeros of a polynomial. Instead of the power method, the most used method for computing directly the eigenvalues is the QR-method. Although the QR-method may be more stable, it need much more expensive computer floating arithmetic operation.

There also exist methods for finding simultaneously all the zeros of a giving polynomial. Two famous algorithms that allow to calculate the zeros in parallel computing [1] [2] are the Weierstrass method and the Aberth method. This paper is to study the derivation of these two parallel algorithms by functional iteration analysis and implicit deflation scheme.

The outline of this paper is as follows. Section 1 is this introduction. Section 2, we briefly review the two famous parallel algorithms for finding the zeros of a given polynomial. In section 3, we study the functional iteration analysis to construct the parallel algorithms. And then we show the derivation of the two algorithms with implicit deflation scheme in section 4. Finally, we have a conclusion in section 5.

Parallel zeros-finding algorithm

Two famous parallel algorithms for finding zeros of polynomials are reviewed in this section. They are Weierstrass method and Aberth methods. Let p(z) be a monic polynomial of degree $n \geqslant 3$ with complex coefficients:

$p(z)=\prod_{j=1}^{n}(z-z_j)=\sum_{k=0}^{n}c_k z^k$, where $c_n=1$.

An algorithm to derive simultaneously the approximate linear factors $(z-x_j)$ for the exact factors $(z-z_j)$, $j=1,2,\ldots,n$, is introduced by K. Weierstrass [3]. In proving the fundamental theorem by constructive method, Weierstrass proposed the algorithm

$$x_i^{(k+1)}=x_i^{(k)}-\frac{p(x_i^{(k)})}{\prod_{j=1, j\neq i}^{n}(x_i^{(k)}-x_j^{(k)})},\quad i=1,2,\ldots,n. \qquad (1)$$

where $^{(k)}$ denotes the iteration index. This iteration method for finding simultaneous the approximation of all zeros of a polynomial was rediscovered by E. Durand [4] and by I. O. Kerner [5] independently. K. Dochev [6] has also studied this iteration by applying the Newton method in the elementary symmetric functions of the zeros.

The Weierstrass method (1) is, in effect, the same as a Newton method. The order of convergence of Weierstrass method will be quadratic [7] [8] if the zeros of the polynomial are simple and the starting values $x_i^{(0)}$ are sufficiently close to the corresponding zeros.

A cubically convergent iteration method in the case of simple zeros was proposed by O. Aberth [9]. He described the idea of deriving the third order convergent method by a vector field of unit plus charges located at the n zeros $z_1,z_2,\ldots,z_n$. From the physical approach, Aberth proposed the algorithm:

$$x_i^{(k+1)}=x_i^{(k)}-\frac{p(x_i^{(k)})}{p'(x_i^{(k)})-p(x_i^{(k)})\sum_{j=1, j\neq i}^{n}\frac{1}{x_i^{(k)}-x_j^{(k)}}},\quad i=1,2,\ldots,n. \qquad (2)$$

Aberth also mentioned this algorithm can be arrived by a purely algebraic derivation. To illustrate the derivation, we let $w_i(z)$ be the product of linear factor

$w_i(z)=\prod_{j=1, j\neq i}^{n}(z-x_j)$,

and rational function $R_i(z)$ be the correction term of Weierstrass method (1)

$$R_i(z)=\frac{p(z)}{w_i(z)},\quad i=1,2,\ldots,n.$$

Differentiating the rational function $R_i(z)$ and applying the Newton method , we have

$$\frac{R_i(z)}{R_i'(z)}=\frac{p(z)}{p'(z)-p(z)\frac{w_i'(z)}{w_i(z)}}=\frac{p(z)}{p'(z)-p(z)\sum_{j=1, j\neq i}^{n}\frac{1}{z-x_j}},$$

$i=1,2,\ldots,n.$

Substituting x_i for z, we again obtain the Aberth algorithm (2).

Functional iteration analysis

In this section, we discuss the derivation for both of Weierstrass and Aberth methods by functional iteration analysis. We use the notation X as the vector of approximate zeros $X=(x_1,x_2,\ldots,x_n)$ and the exact zeros vector $Z=(z_1,z_2,\ldots,z_n)$.

The zeros-finding methods can be constructed as functional iteration form:

$X^{(k+1)}=F(X^{(k)})=(f_1(X^{(k)}),f_2(X^{(k)}),\ldots,f_n(X^{(k)}))$.

Taking the idea of Jacobi iteration technique, we have

$z_i^{(k+1)}=f_i(z_1^{(k)},z_2^{(k)},\ldots,z_n^{(k)}),i=1,2,\ldots,n,$

and the Gauss-Seidel-like version

$z_i^{(k+1)}=f_i(z_1^{(k+1)},\ldots,z_{i-1}^{(k+1)},z_i^{(k)}\ldots,z_n^{(k)}),i=1,2,\ldots,n.$

A monic polynomial p(z) with simple zeros z_1 $,z_2,\ldots,z_n$ can be written as

$p(z)=\prod_{j=1}^{n}(z-z_j)=(z-z_i)\prod_{j=1, j\neq i}^{n}(z-z_j)$,

then we get the fixed point relation

$$z_i=z-\frac{p(z)}{\prod_{j=1, j\neq i}^{n}(z-z_j)}.$$

This relation suggests a way to construct the functional iteration by substituting the zeros z_j by approximation x_j and putting $z=x_i$, we have

$$f_i(X)=x_i-\frac{p(x_i)}{\prod_{j=1, j\neq i}^{n}(x_i-x_j)},i=1,2,\ldots,n.$$

Applying the Gauss-Seidel-like version, we obtain the algorithm

$$x_i^{(k+1)}=x_i^{(k)}-\frac{p(x_i^{(k)})}{\prod_{j=1}^{i-1}(x_i^{(k)}-x_j^{(k+1)})\prod_{j=i+1}^{n}(x_i^{(k)}-x_j^{(k)})},i=1,2,...,n$$

In the Jacobi-like version, we have the Weierstrass algorithm (1).

In order to derive the Aberth method by functional iteration, we consider a vector valued function of n variables F: $R^n \rightarrow R^n$ defined by $F(X)=(f_1(X),f_2(X),...f_n(X))$, where

$$f_i(X)=\frac{p(x_i)}{\prod_{j=1, j\neq i}^{n}(x_i-x_j)},\ i=1,2,...,n.$$

In the following theorem, we give the derivation for the Aberth algorithm by the Jacobi-Newton construction.

Theorem 1:

(1) Let $Z=(z_1,z_2,...,z_n)$. Then $F(Z)=0$ if only if $p(z_j)=0$, $j=1,...,n$.

(2) The Aberth method can be derived by applying the Newton method to $f_j(Z)=0$, $j=1,...,n$.

Proof:

It is easy to prove (1) using the definition of the function F.

Now we consider the Newton method applying to $f_i(Z)=0$, that is,

$$x_i^{(k+1)}=x_i^{(k)}-\frac{f_i(x_i^{(k)})}{f_i'(x_i^{(k)})},\ i=1,...,n.$$

Since $f_i'(x_i^{(k)})$

$$=\frac{1}{\prod_{j=1, j\neq i}^{n}x_i^{(k)}-x_j^{(k)}}\left[p'(x_i^{(k)})-p(x_i^{(k)})\sum_{j=1, j\neq i}^{n}\frac{1}{x_i^{(k)}-x_j^{(k)}}\right]$$

, we have

$$x_i^{(k+1)}=x_i^{(k)}-\frac{p(x_i^{(k)})}{p'(x_i^{(k)})-p(x_i^{(k)})\sum_{j=1, j\neq i}^{n}\frac{1}{x_i^{(k)}-x_j^{(k)}}},\ i=1,...,n.$$

We get the Aberth algorithm again. This completes the proof.

Applying the Newton method to the same function F(Z) in the Gauss-Seidel-like version, we have the algorithm

$$x_i^{(k+1)}=x_i^{(k)}-\frac{p(x_i^{(k)})}{p'(x_i^{(k)})-p(x_i^{(k)})\left(\sum_{j=1}^{i-1}\frac{1}{x_i^{(k)}-x_j^{(k)}}+\sum_{j=i+1}^{n}\frac{1}{x_i^{(k)}-x_j^{(k)}}\right)}$$

Implicit deflation Scheme

Most of the efficient algorithms for solving nonlinear equations may be used in finding zeros of polynomials. For a given polynomial, the zeros can be found using some standard methods, for example, Bernoulli's method, Newton's method, and Laguerre's method. These methods determine one zero at a time. Once a zero has been found, the corresponding linear factor has to be divided out for determining the next zero. This deflating process makes the error accumulation since the roundoff-error and each zero found is just a approximation of the exact zero. One way to avoid the error accumulation is that each zero is used in turn as initial value for the algorithm to the full undeflated polynomial after all of approximate zeros have been found by the deflation process. But this method slows down the convergence. We may use implicit deflation technique to speed up convergence and improve accuracy.

If an approximate zero x_1 of p(z) has been determined by a zero-finding method, we wish to find an approximation x_m to the next zero of p(z) by dividing out the factor $z-x_1$. The basic idea of implicit deflation method is that at the first a relative function f(z) satisfying $f(z)(z-x_i)=p(z)$ is produced implicitly, and then the same zero-finding method applies to f(z) for determining the further zero of p(z).

For a given polynomial p(z) of degree n with simple zeros $z_1,z_2,...,z_n$, we consider two algorithms as discrete dynamic systems:

$$x^{(k+1)}=g(x^{(k)})\equiv x^{(k)}-p(x^{(k)}),\ k=0,1,..., \qquad (3)$$

$$x^{(k+1)}=h(x^{(k)})\equiv x^{(k)}-\frac{p(x^{(k)})}{p'(x^{(k)})},\ k=0,1,..., \qquad (4)$$

The rate of convergence of the system (3) with fixed point iteration operator $g(z)=z-p(z)$ is usually linear and the fixed point of g(z) is the zero of the given polynomial p(z). The system (4) with Newton iteration operator $h(z)=z-(p(z)/p'(z))$, as we know, is quadratic convergent and the zero of the polynomial p(z) is also the fixed point of the Newton iteration operator h(z).

Once a zero x_1 of the polynomial p(z) is determined using algorithm (1), we want to find the next zero x_m with implicit deflation, that is, to find the zero of $p_m(z) = p(z)/(z-x_1)$. Then, we have $p_m(x_m) = p(x_m)/(x_m-x_1)$. This give a algorithm

$$x_m^{(k+1)} = x_m^{(k)} - p(x_m^{(k)})/(x_m^{(k)} - x_1^{(k)})$$

for determining the next zero x_m of p(z) by fixed point iteration algorithm (3) with implicit deflation after a zero x_1 has been found.

Now suppose $x_1, x_2, \ldots, x_{i-1}, x_{i+}, \ldots, x_n$, for i=1,2,... , n, has been determined, the implicit deflation method is applied to $p_i(z) = p(z)/\prod_{j=1, j\neq i}^{n}(z-x_j)$, we have

$p_i(x_i) = p(x_i)/\prod_{j=1, j\neq i}^{n}(x_i-x_j)$. This give the Weierstrass algorithm again. It is interesting and useful observation and then we have following results.

Theorem 2:

(1) The Weierstrass method can be derived by the fixed point iteration algorithm with implicit deflation and its order of convergence is increased by one.

(2) The Aberth method can be derived by the Newton method with implicit deflation and its order of convergence is increased by one.

Proof:

(1) is already done.

We now prove (2) by a recurrence relation.

In the following discussion, we fixed an index i, $1 \leq i \leq n$. Let the relative function f(z) satisfy

$$p(z) = \prod_{m=1, m\neq i}^{n}(z-x_m)\, f(z), \text{ and}$$

$$f_j(z) = \prod_{m=1, m\neq i}^{n}(z-x_m)\, f(z),\ j=1,\ldots,i-1,i+1,\ldots,n.$$

Thus, $f_1(z) = p(z)$, $f_{i-1}(z) = (z-x_{i-1})f_{i+1}(z)$, and

$f_j(z) = (z-x_j)f_{j+1}(z)$, $j = 1,\ldots,i-2,i+1,\ldots,n$.

In order to get the recurrence relation $f_j(x_i)/f_j'(x_i)$, we consider

$$f_j(z) = (z-x_j)f_{j+1}(z) = (z-x_j)[(z-x_i)^2 q(z) + (z-x_i)f_{j+1}'(x_i) + f_{j+1}(x_i)]$$

Replacing $z-x_j$ by $(z-x_i)+(x_i-x_j)$, we have

$$f_j(z) = (z-x_j)^2 Q(z) + [(x_i-x_j)f_{j+1}'(x_i) + f_{j+1}(x_i)] + (x_i-x_j)f_{j+1}(x_i).$$

Substituting z by x_i, we get

$$f_{i+1}(x_i) = \frac{f_{i-1}(x_i)}{x_i-x_{i-1}},\quad f_{j+1}(x_i) = \frac{f_j(x_i)}{x_i-x_j},$$

$j = 1,\ldots,i-2,i+1,\ldots,n$, and

$$f_{i+1}'(x_i) = \frac{(x_i-x_{i-1})f_{i-1}'(x_i) - f_{i-1}(x_i)}{(x_i-x_{i-1})^2}.$$

$$f_{j+1}'(x_i) = \frac{(x_i-x_j)f_j'(x_i) - f_j(x_i)}{(x_i-x_j)^2}.\ j = 1,\ldots,i-2, i = 1,\ldots,n.$$

It follows that

$$\begin{aligned}\frac{f'(x_i)}{f(x_i)} &= \frac{(x_i-x_n)f_n' - f_n(x_i)}{(x_i-x_n)^2} \quad \frac{x_i-x_n}{f_n(x_i)} \\ &= \frac{f_n'(x_i)}{f_n(x_i)} - \frac{1}{x_i-x_n} \\ &= \frac{f_{n-1}'(x_i)}{f_{n-1}(x_i)} - \frac{1}{x_i-x_n} - \frac{1}{x_i-x_{n-1}} \\ &= \cdots = \frac{f_1'(x_i)}{f_1(x_i)} - \sum_{j=1, j\neq i}^{n}\frac{1}{x_i-x_j}.\end{aligned}$$

Therefore, the algorithm of the Newton method with implicit deflation is

$$\begin{aligned}x_i^{(k+1)} &= x_i^{(k)} - \frac{f(x_i^{(k)})}{f'(x_i^{(k)})} \\ &= x_i^{(k)} - \frac{f_1(x_i^{(k)})}{f_1'(x_i^{(k)}) - f_1(x_i^{(k)})\sum_{j=1, j\neq i}^{n}\frac{1}{x_i^{(k)}-x_j^{(k)}}}\end{aligned}$$

This algorithm is just the Aberth algorithm since $f_1(z) = p(z)$.

Conclusion

Two famous parallel zero-finding algorithms of polynomials are explored. It has been shown that the Weierstrass algorithm and the Aberth algorithm can be derived from the functional iteration analysis in Jacobi-like version. We also observe that these two algorithms can be arrived by performing the implicit deflation scheme.

Reference

(1)Cosnard, M. & Fraigniaud, P. (1990). Finding the roots of a polynomial on an MIMD multicomputer. *Parallel Computing.* 15, 75-85.

(2)Cosnard, M. & Fraigniaud, P. (1994). Analysis of asynchronous polynomial root finding methods on a distributed memory multicomputer. *IEEE Transactions on parallel and distributed system. 5,* 639-648.

(3)Green, M. W., Korsak, A. J. & Pease, M. C. (1976). Simultaneous iteration towards all roots of a complex polynomial. *SIAM Rev. 18,* 501-502.

(4)Durand, E. (1960). Solution numériques des équations algebraiques, Tom. I: Équations du Type F(x)=0; *Racines d'un Pôlynome.* Masson, Paris.

(5)Kerner, I. O. (1966). Ein Gesamtschrittverfahren zur Berechnung der Nullstellen von Polynomen. *Numer. Math. 8,* 290-294.

(6)Dochev, K. (1962). Modified Newton method for the simultaneous approximate calculation of all roots of a given algebraic equation (in Bulgarian). *Mat. Spis. B'lgar Akad. Nauk 5,* 136-139.

(7)Pasquini, L. & Trigiante, D. (1985). A globally convergent method for simultaneously finding polynomial roots. *Mathematics of computation.* 44, 135-149.

(8)Kjurkchiev, N. (1993). Some remarks on Weierstrass root-finding method. C.R. *Acad. Bulgare Sci. 46,* 17-20.

(9)Aberth, O. (1973). Iteration methods for finding all zeros of a polynomial simultaneously. *Mathematics of computation. 27,* 339-344.

收稿日期：86年 7月 1日
修正日期：86年 10月 23日
接受日期：86年 10月 23日

解多項式零位之平行演算法

左台益

國立臺灣師範大學數學系

本文研究適合平行計算之二種演算法 Weierstrass 法及 Aberth 法以求解多項式之零位。我們說明由函數疊代分析可以導出此二種演算法。同時也驗證 Weierstrass 法可由不動點疊代法結合隱式除法計算導出，而牛頓法結合隱式除法可計算出 Aberth 法。

關鍵詞：平行計算　函數疊代分析　解零位　隱式除法