An Adaptive Algorithm Finding Multiple Roots of Polynomials

Wei Zhu[1,2,*], Zhe-zhao Zeng[1,*], and Dong-mei Lin[1]

[1] College of Electrical & Information Engineering, Changsha University of Science &Technology, Changsha, Hunan 410076, China
[2] College of Electrical & Information Engineering, Hunan University, Changsha, Hunan 410082, China
hncs6699@yahoo.com.cn

Abstract. An adaptive algorithm is proposed to find multiple roots of polynomials which were not well solved by the other methods. Its convergence was presented and proved. The computation is carried out by simple steepest descent rule with adaptive variable learning rate. The specific examples showed that the proposed method can find the multiple roots of polynomials at a very rapid convergence and very high accuracy with less computation.

Keywords: Adaptive Algorithm, Multiple Real or Complex Roots, Variable Learning Rate.

1 Introduction

Finding rapidly and accurately the roots of polynomials is an important problem in various areas of control and communication systems engineering, signal processing and in many other areas of science and technology. The problem of finding the zeros of a polynomial has fascinated mathematicians for centuries, and the literature is full of ingenious methods, analyses of these methods, and discussions of their merits [1-3]. Over the last decades, there exist a large number of different methods for finding all polynomial roots either iteratively or simultaneously. Most of them yield accurate results only for small degree or can treat only special polynomials, e.g., polynomials with simple real or complex roots [4].

So far, some better modified methods of finding roots of polynomials cover mainly the Jenkins/Traub method [5], the Markus/Frenzel method [4], the Laguerre method [6], the Routh method [7], the Truong, Jeng and Reed method [8], the Fedorenko method [9], the Halley method [10], and some modified Newton's methods [11-13], etc. Although the Laguerre method is faster convergent than all other methods mentioned above, it has more computation. Among other methods, some have low accuracy, and some have more computation, especially to say is the modified Newton's methods must have a good initial value near solution. Furthermore, it is very difficult for the all methods mentioned above to find multiple real or complex roots of polynomials.

---

* Corresponding authors.

F. Sun et al. (Eds.): ISNN 2008, Part II, LNCS 5264, pp. 674–681, 2008.

In order to solve the problems above, we propose an algorithm finding multiple real or complex roots of polynomials with adaptive variable learning rate. The approach can find multiple roots of polynomials with less computation, high accuracy and rapid convergence.

2 The Algorithm Finding Multiple Zeros of Polynomials

2.1 The Algorithm Description

We start by defining our typical polynomial of degree n as

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_0 \tag{1a}$$

$$= a_n (x - p_1)^{m_1} (x - p_2)^{m_2} \cdots (x - p_l)^{m_l} \tag{1b}$$

Where, $\sum_{j=1}^{l} m_j = n$, and $1 < m_i < n\,(i = 1,2,\cdots,l)$.

Here we are given the coefficients, a_i ($a_n \neq 0$), and wish to find the multiple real or complex zeros: p_i . Usually, in science and engineering applications, the coefficients will all be real, and then the zeros will either be real or else occur in conjugate-complex pairs.

Let us then assume for the moment that all the p_i is real or complex and distinct, and numbered so that

$$\mathrm{Re}(p_1) < \mathrm{Re}(p_2) < \cdots \mathrm{Re}(p_l) \tag{2}$$

Also we will assume that we have made some real-valued or complex-valued guess, p_k , possibly quite crude, for one of the zeros, and that

$$\mathrm{Re}(p_m) < \mathrm{Re}(p_k) < \mathrm{Re}(p_{m+1}) \tag{3}$$

It is well known that the multiple root p_i is also the root of polynomial $f^{(m_i-1)}(x)$ while p_i is the multiple m_i root of function $f(x)$. The principal feature of the algorithm proposed is to make $f^{(m_i-1)}(x)$ satisfy

$$f^{(m_i-1)}(x) = 0$$

by training the weighting variable x . The algorithm is as follows:

Given an arbitrary initial approximation weighting coefficient x_k , real or complex, an error function can be obtained:

$$e(k) = 0 - f^{(m_i-1)}(x_k) = -f^{(m_i-1)}(x_k) \tag{4}$$