

Parallelized methods for solving polynomial equations

Rinela Kapçiu¹, Fatmir Hoxha², Eglantina Kalluçi³

¹(Computer Science Department/ "Aleksandër Moisiu" University of Durrës ALBANIA)

²(Applied Mathematics Department/ University of Tirana ALBANIA)

³(Applied Mathematics Department/ University of Tirana ALBANIA)

Abstract: The current microprocessors are concentrating on the multiprocessor or multi - core system architecture. The parallel algorithms are recently focusing on multi - core system to take full utilization of multiple processors available in the system. The design of parallel algorithm and performance measurement is the major issue on today's multi - core environment. Numerical problems arise in almost every branch of science which requires fast solution. In this paper we have presented parallel algorithms for computing the solution of system of non - linear equations and approximate the simple zeros of polynomial equations. The experimental results reveal that the performances of parallel algorithms are better than sequential. We implemented the parallel algorithms using multithreading features of OpenMP.

Keywords: Approximate roots, Börsch – Supan method, Durand-Kerner method, Parallelization, Simultaneous root finding methods

I. Introduction

One of the most important problems in solving nonlinear equations is the construction of the initial conditions which provide rapid convergence of numerical algorithm. In this paper we present two methods Durand – Kerner and Börsch – Supan, which have some initial new conditions to ensure convergence of methods for solving algebraic equations. The stated initial conditions are of practical importance since they are computationally verifiable, they depend only on the coefficients of a given polynomial, its degree n and initial approximations to polynomial roots. [1]

One of the main problems in the solution of equations of the form $f(z) = 0$ is the construction of the initial conditions which offer guaranteed convergence of numerical algorithms. These initial conditions include an initial approximation $z^{(0)}$ to the root of f with which starts the implementation of the algorithm to generate the sequence $\{z^{(m)}\}_{m=1,2,\dots}$ of approximations tends to the root of f . The study of a general problem of the construction of the initial conditions and the choice of initial approximation to ensure convergence is very difficult and generally cannot be solved in a satisfactory way, even for simple functions such as algebraic polynomials.

II. The Simultaneous Methods

In this paper we present two methods that provide improved conditions and fast convergence. These methods are: Durand – Kerner and Börsch – Supan. These conditions depend only on the coefficients of the given polynomial $P(z) = z^n + a_{n-1}z^{n-1} + \dots + a_1z + a_0$ of degree n and the vector of initial approximations $z^{(0)} = (z_1^{(0)}, \dots, z_n^{(0)})$. Most of iterative methods for the simultaneous determination of roots of a polynomial can be expressed as:

$$z_i^{(m+1)} = z_i^{(m)} - C(z_1^{(m)}, \dots, z_n^{(m)}) \quad (i \in I_n, m = 0, 1, \dots) \quad (1.1)$$

Where $z_1^{(m)}, \dots, z_n^{(m)}$ are distinct approximations to simple roots ζ_1, \dots, ζ_n respectively, obtained in the m -th step. The term $C_i^{(m)} = C_i(z_1^{(m)}, \dots, z_n^{(m)}) \quad (i \in I_n)$ will be called the *iterative correction term* or simply the *correction*. [1]

Let $\Lambda(\zeta_i)$ a close neighbourhood of the root $\zeta_i \quad (i \in I_n)$ and the function $(z_1, \dots, z_n) \rightarrow F_i(z_1, \dots, z_n)$ that satisfies the following conditions for each $i \in I_n$:

$$\begin{aligned} (1) & F_i(\zeta_1, \dots, \zeta_n) \neq 0, \\ (2) & F_i(z_1, \dots, z_n) \neq 0 \text{ for distinct approximations } z_i \in \Lambda(\zeta_i), \\ (3) & F_i(z_1, \dots, z_n) \text{ is continuous in } \square^n \end{aligned}$$

If the correction term of iterative method 1.1 has the form

$$C_i(z_1, \dots, z_n) = \frac{P(z_i)}{F_i(z_1, \dots, z_n)} \quad (i \in I_n) \tag{1.2}$$

for which conditions (1) – (3) hold and $z_1^{(0)}, \dots, z_n^{(0)}$ are the initial approximations to the polynomial roots, in [5] proved that this method is convergent if there is a real number $\beta \in (0, 1)$ such that satisfies the following inequalities::

- (i) $|C_i^{(m+1)}| \leq \beta |C_i^{(m)}| \quad (m = 0, 1, \dots).$
- (ii) $|z_i^{(0)} - z_j^{(0)}| > g(\beta) (|C_i^{(0)} + C_j^{(0)}|) \quad (i \neq j, \quad i, j \in I_n)$

III. The Durand – Kerner Method

One of the most useful simultaneous methods for solving a polynomial is the Durand - Kerner (Weierstrass) method expressed as follows:

$$z_i^{(m+1)} = z_i^{(m)} - W_i^{(m)} \quad (i \in I_n, m = 0, 1, \dots) \tag{1.3}$$

where

$$W_i^{(m)} = \frac{P(z_i^{(m)})}{\prod_{\substack{j=1 \\ j \neq i}}^n (z_i^{(m)} - z_j^{(m)})} \quad (i \in I_n, m = 0, 1, \dots)$$

In this case the correction term is equal to Weierstrass's correction

$$C_i = W_i = \frac{P(z_i)}{F_i(z_1, \dots, z_n)} \text{ ku } F_i(z_1, \dots, z_n) = \prod_{\substack{j=1 \\ j \neq i}}^n (z_i - z_j) \quad (i \in I_n).$$

The Durand – Kerner algorithm

1. Compute initial values $\{z_0; \dots; z_{n-1}\}$
2. Let $m=1$;
3. do
4. $\Delta z_{\max} = 0$;
5. for $j = 0, \dots, n-1$
6. $z_j^{m-1} = z_j^m$;
7. $z_j^{m-1} = H_i(z^{m-1})$;
8. Set $\Delta z_{\max} = \frac{|z_j^m - z_j^{m-1}|}{z_j^m}$;
9. $k=k+1$;
10. while $z_{\max} > \varepsilon$

IV. The Börsch- Supan Method

The Börsch – Supan method is a simultaneous method, which is determined by the iterative formula

$$z_i^{(m+1)} = z_i^{(m)} - \frac{W_i^{(m)}}{1 + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{W_j^{(m)}}{(z_i^{(m)} - z_j^{(m)})}} \quad (i \in I_n, m = 0, 1, \dots) \tag{1.4}$$

where $W_i^{(m)}$ is given by (1.2). This formula has term correction:

$$C_i(z_1, \dots, z_n) = \frac{P(z_i)}{F_i(z_1, \dots, z_n)} \quad (i \in I_n)$$

where

$$F_i(z_1, \dots, z_n) = \left(1 + \sum_{\substack{j \neq i \\ j=1}}^n \frac{W_j}{z_i - z_j} \right) \prod_{j \neq i} (z_i - z_j) \quad (i \in I_n)$$

The Börsch – Supan algorithm

1. Compute initial values $\{z_0; \dots; z_{n-1}\}$
2. Let $m=1$;
3. *do*
4. $\Delta z_{\max} = 0$;
5. *for* $j = 0, \dots, n-1$
6. $z_j^{m-1} = z_j^m$;
7. $z_j^{m-1} = H_i(z^{m-1})$;
8. Set $\Delta z_{\max} = \frac{|z_j^m - z_j^{m-1}|}{z_j^m}$;
9. $z_i^{(m+1)} = z_i^{(m)} - \frac{W_i^{(m)}}{1 + \sum_{\substack{j=1 \\ j \neq i}}^n \frac{W_j^{(m)}}{(z_i^{(m)} - z_j^{(m)})}}$;
10. $k=k+1$;
11. *while* $z_{\max} > \varepsilon$

V. The Wilkinson’s Polynomial

In numerical analysis, Wilkinson polynomial is a specific polynomial which was used by James H. Wilkinson 1963 to show the difficulty when finding the root of a polynomial. The Wilkinson polynomial

$$w(x) = \prod_{i=1}^{20} (x - i) = (x - 1)(x - 2) \dots (x - 20)$$

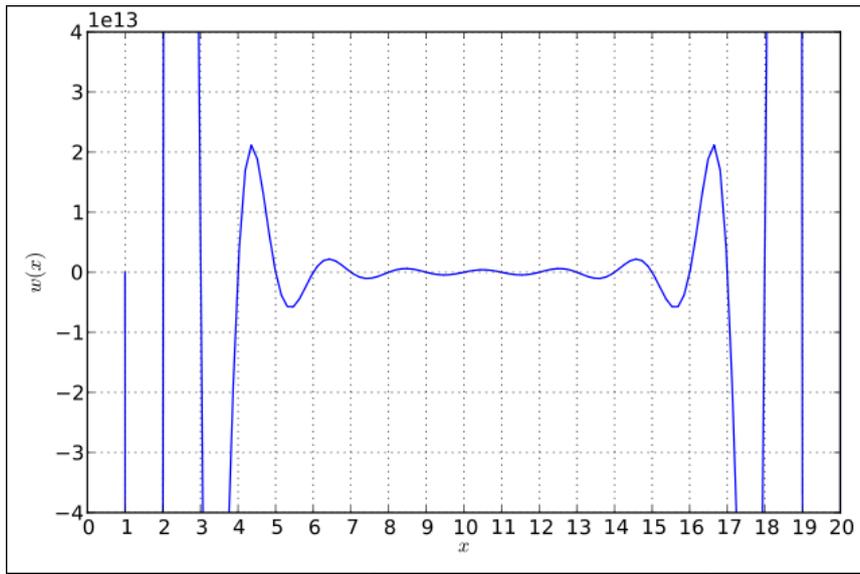
has 20 root, localized in points $x = 1, 2, \dots, 20$ respectively. These roots are separated from each - other, but they are still ill - conditioned.

Expanding the polynomial we have

$$\begin{aligned} w(x) = & x^{20} - 210x^{19} + 20615x^{18} - 1256850x^{17} + 53327946x^{16} \\ & - 1672280820x^{15} + 40171771630x^{14} - 756111184500x^{13} \\ & + 11310276995381x^{12} - 135585182899530x^{11} \\ & + 1307535010540395x^{10} - 10142299865511450x^9 \\ & + 63030812099294896x^8 - 311333643161390640x^7 \\ & + 1206647803780373360x^6 - 3599979517947607200x^5 \\ & + 8037811822645051776x^4 - 12870931245150988800x^3 \\ & + 13803759753640704000x^2 - 8752948036761600000x \\ & + 2432902008176640000. \end{aligned}$$

If the coefficient of x^{19} is decreased from -210 by 2^{-23} to -210.0000001192, then the polynomial value $w(20)$ decreases from 0 to $2^{-23}20^{19}=-6.25 \times 10^{17}$, and the root at $x=20$ grows to $x \approx 20.8$. The roots at $x=18$ and $x=19$ transform into a double root at $x \approx 18.62$ which turns into a pair of complex conjugate roots at $x \approx 19.5 \pm 1.9i$ as the perturbation increases further. The 20 roots become

1.00000	2.00000	3.00000	4.00000	5.00000
6.00001	6.99970	8.00727	8.91725	20.84691
10.09527 ± 0.64350i	11.79363 ± 1.65233i	13.99236 ± 2.51883i	16.73074 ± 2.81262i	19.50244 ± 1.94033i



The plot 1.1 The plot of Wilkinson's polynomial

Hereinafter we give the results taken from testing Durand - Kerner and Börsch - Supan algorithms in C++ programming language and platform OpenMP.

Table 1 - 1 Execution time of 2 methods in C++ programming language

The polynomials	Durand – Kerner	Börsch – Supan
The simple polynomial	0.54s	0.75s
The Wilkilson's polynomial	8.00s	9.3s

Table 1 – 2 Execution time of 2 methods in OpenMP platform

The polynomials	Durand – Kerner	Börsch – Supan
The simple polynomial	0.25s	0.4s
The Wilkilson's polynomial	5.3s	5.7s

After these tests, we see that the OpenMP platform is quite suitable for the execution of these methods because the time consumed is smaller compared with the sequential time.

VI. Conclusions

After we made some necessary tests we concluded that:

By testing these two simultaneous methods we see that the OpenMP platform is more qualitative than the sequential execution. As seen on the platform OpenMP implemented in our algorithms, their performance increases and this happens in the same drive hardware, with the same parameters, just exploiting parallelism and increasing the use of all potential multithread processor.

OpenMP is well adapted to intensive computing. We parallelized the Durand – Kerner algorithm and Börsch – Supan algorithm for polynomial roots - finding and we obtained encouraging results. Indeed, the experimental study confirms that our program determines the same roots than the sequential version for high

degrees. The contribution of the parallel solution allows us to accelerate the execution time and to study even more important degrees of polynomial.

References

- [1]. M.S. Petkovic^{a,*}, Đ. Herceg, Point estimation of simultaneous methods for solving polynomial equations: a survey, *J. Computational and Applied Mathematics* 136 (2001) 283–307
- [2]. P.Batra, Improvement of a convergence condition for the Durand–Kerner iteration, *J.Comput.Appl.Math.* 96 (1998) 117–125.
- [3]. M.S. Petkovic, On initial conditions for the convergence of simultaneous root - finding methods, *Computing* 57 (1996) 163–177.
- [4]. M.S. Petkovic, C.Carstensen, M.Trajkovic, Weierstrass’ formula and zero - finding methods, *Numer.Math.*69 (1995) 353–372.
- [5]. M.S. Petkovic ".Herceg, S.Ilic, Safe convergence of simultaneous methods for polynomial zeros, *Numer. Algorithms* 17 (1998) 313–331.
- [6]. A.W.M. Nourain, An iteration formula for the simultaneous determination of the zeroes of a polynomial, *J. Comput. Appl.Math.*4 (1975) 251–254.
- [7]. M.S. Petkovic^{a,*}, Đ. Herceg, Point estimation and safe convergence of root - finding simultaneous methods, *Sci.Rev.* 21–22 (1996) 117–130.
- [8]. W. Börsch – Supan, A posteriori error bounds for the zeros of polynomials, *Numer.Math.*5 (1963) 380–398.
- [9]. C. Carstensen, On quadratic – like convergence of the means for two methods for simultaneous root – finding of polynomials, *BIT* 33 (1993) 64–73.
- [10]. F. L. Lucio On the convergence of a parallel algorithm for finding polynomial zeros, 1997.
- [11]. M. Ben et al, A fast parallel algorithm for determining all roots of a polynomial with real roots, *SIAM J. Comp.* 11, 1988.
- [12]. D. Bini, L. Germignani, On the complexity of polynomial zeros, *SIAM, J. Comp.* 21, 1992.
- [13]. T. Auckenthaler, et al, Parallel solution of partial symmetric eigenvalue problems from electronic structure calculations, *Parallel Comp.* 2011.
- [14]. E. R.Alcalde, Parallel implementation of Davidson-type methods for large-scale eigenvalue problem, 2012.
- [15]. C. Caretensen and M. S. Petkovic, On iterative methods without derivatives for the simultaneous determination of polynomial zeros, *J. Comp. & Appl. Math.*, **45**, pp 251 – 266, 1993 E238
- [16]. Z. Bai, Demmel, J. Dongarra, A. Ruhe, H. Van der Vorst, *Templates for the solution of algebraic eigenvalue problems: a practical guide*, SIAM, Philadelphia, 2000. E238