Bul.St.Univ.Baia Mare
Seria B, Matematică-Informatică, Vol.VII, Nr.1-2,127-132

On the Efficiency of Lobacevschi-Graeffe's Method for Finding the Polynomial Zeros

by

Ioana CHIORESAN

Let $f(x) = 0$ be a nonlinear equation. The problem of determining its zeros is a very interesting and important one, existing many methods, iterative or not, see [2] - [1], to do this.

From this point of view, a special case is occupied by those nonlinear equations in which the function f is an algebraic polynomial one, with real coefficients, which can be solve by usually used methods, but also by special ones. So, in [4] is presented the Lobacevschi-Graeffe's method, which can be easy translated in an serial algorithmic form.

The aim of this paper is to derive the parallel algorithm for this method and to characterize it by speedup, respectively by the efficiency point of view.

To make it possible, we suppose that a MIMD system - an arbitrary number of processors with independent control and arbitrary large memory with unrestricted access is available.(see [5]). Also, one suppose that each processor is capable to perform any of the operations $+$, $-$, $*$, $/$. The time required for accessing data, storing results and communicating among processors is ignored.

As it is well known, one of the characteristic of a parallel method F_p is the speedup $S(F_p;n)$, i.e.

$$S(F_p;n) = \frac{CP(\tilde{F}_s)}{CP(F_p;n)}$$

where $\tilde{F}_s$ is the optimal serial method with regard to the complexity in a given class of such methods, (see [1]), $CP(F_p;n)$ is the complexity of the parallel method F_p and n is the number of processors.

Using the speedup $S(F_p;n)$, a new characteristic for a parallel method is defined, the efficiency:

$$e(F_p) = \frac{S(F_p;n)}{n}$$

Obviously ,for a given F_p , we have $1 \leq S(F_p;n) \leq n$, respectively $0 < e(F_p) \leq 1$.

As the optimal serial method F_s is not always accesible , the speedup is usually approximated by the value

$$s(F_p;n) = \frac{CP(F_s)}{CP(F_p;n)}$$

where F_s is an available serial method and F_p is a parallel version of F_s. It follows that

$$s(F_p;n) \geq S(F_p;n) .$$

Note. Let $\propto$ be an algorithm. Then

$CP(\propto) = CPI(\propto) + CPC(\propto)$, where $CPI(\propto)$ is the informational complexity of algorithm $\propto$ and $CPC(\propto)$ is the computational complexity of it. Because in this paper $CPI(\propto) = 0$ (see [1]) , by $CP(\propto)$ we understand only the $CPC(\propto)$.

Let f be an algebraic polynomial function of degree n with real simple zeros $x_1, x_2, \ldots, x_n$, i.e.

$$f(x) = a_0x^n + a_1x^{n-1} + \ldots + a_n \qquad (1)$$

All zeros are considered to be different , because if it is not so , by dividing f(x) with the least common multiple of f(x) and $f'(x)$ (determined with the Euclid's algorithm) , the assuming is accomplish.

Note. In our next considerations , we assume that $|x_i| \neq |x_j|$, $i \neq j$,it means that we don't refer to the cases $x_i = -x_j$ and $x_i = a + ib$, $x_j = a - ib$, for $i \neq j$.

First case. The zeros satisfie the condition

$$|x_1| >> |x_2| >> |x_3| >> \ldots >> |x_n| \qquad (2)$$

where ">>" means "very much greater than" . It follows that the zeros are well separated.

Relation (2) allowes us to write:

$$x_1 = \xi_1 x_1 \; , \; x_2 = \xi_1 x_2 \; , \; \ldots \; , \; x_n = \xi_{n-1} x_n \; , \qquad (3)$$

where $|\xi_i| < \xi$, $i=1,\ldots,n$, ξ less than unit.

Using the Viete's relations between zeros and coefficients and (3) , it is obtain

$$x_1 (1 + \xi_1) = -\frac{b_1}{b_0}$$

$$x_1 x_2 (1 + \xi_2) = \frac{b_2}{b_0}$$

$$\ldots\ldots\ldots\ldots\ldots$$

$$x_1 x_2 \ldots x_n (1 + \xi_n) = (-1)^n \frac{b_n}{b_0}$$

where from , neglecting $\xi_1, \xi_2, \ldots, \xi_n$ which are very small ,

$$x_1 \simeq -\frac{b_1}{b_0} \; , \; x_1 x_2 \simeq \frac{b_2}{b_0} \; , \; \ldots \; , \; x_1 x_2 \ldots x_n \simeq (-1)^n \frac{b_n}{b_0} \; ,$$

so the zeros $x_1, x_2, \ldots, x_n$ can be obtained by solving the following system

$$\begin{cases} b_0 x_1 + b_1 = 0 \\ b_1 x_2 + b_2 = 0 \\ \ldots\ldots\ldots\ldots \\ b_{n-1} x_n + b_n = 0 \end{cases} \qquad (4)$$

The second case.If the zeros $x_1, \ldots, x_n$ does not satisfies (2) , they can't be computed by the previous method. But there is a possibility to replace the equation (1) by a modified one , whose $y_1, \ldots, y_n$ zeros are , respectively , equal to $x_1^k, \ldots, x_n^k$. Obviously , if $k \gg 2$, (2) is accomplish for the new equation.

This one , which is called "the transformed" , is obtained ,i.e.

$$h(y) = g(x) \cdot g(-x)$$

with $y = x^2$ and with zeros $y_i = x_i^2$ $(i=1,\ldots,n)$.

It is easy to prove that the coefficients c_i of polynomial function h are computed by means of the relations:

$$c_i = b_i^2 + \sum_{r=1}^{i} (-1)^r \cdot b_{i+r} \cdot b_{i-r} \; , \; i=0,\ldots,n \qquad (5)$$

(where $b_{i-r} = 0$ if $i - r < 0$ or $i + r > n$).

Applying repetitively (of j times) this method , the following equation is obtained:

$$p(z) = d_0 z^n + d_1 z^{n-1} + \ldots + d_n \quad , \qquad (6)$$

whose zeros are the numbers $z_i = -x_i^{-1}$. Constructing for polynomial (6) a system of form (4) , $z_1 , \ldots , z_n$ are to be found and then $x_1 , \ldots , x_n$.

Considering a serial algorithm for this method (the second case , which is more general) , we can compute its complexity. So , we count all the arithmetic operations which are involved.

To compute (5) , we make , firstly , $(-1)^n$, which requires (n-1) multiplications , the powers $(-1)^i$, i=0,...,n being stored.

Counting all the additions (suppose CP(+) = CP(-)) and multiplications in (5) , we obtain n(n+3) multiplications and n(n+1)/2 additions.

Further , the system (4) needs n division (the unary operation (-) being not take into account).So , if F_s denotes this serial algorithm , then we can write

$$CP(F_s) = \frac{n(n+1)}{2} CP(+) + n(n+3) CP(*) + n CP(/) \quad (7)$$

At the Felix C-256 computer , for instance , where CP(+) = CP(-) = 8.10 μ , CP(*) = 24.40 μ , CP(/) = 27.30 μ

$$CP_1(F_s) = n (28.45 n + 104.55) \mu \quad . \qquad (8)$$

Now , we consider that the method is executed on a MIMD system , so we have an arbitrary number of processors (say n+1) with independent control and arbitrary large memory with unrestricted acces. Hence , each processor can perform the arithmetic operations , independently or connected by the others.

To compute (5) , we consider that each of the n+1 processors computes one of the coefficients c_i , i = 0,...,n . Obviously , the processing time will be different for each c_i , the longest one being that for the last coefficient. Like in the serial algorithm , we assume that a single processor computes $(-1)^n$ by means of (n-1) multiplications. So , the operations involved in parallel computation of (5) are n additions and 3n multiplications. Further , if n processors compute , simultaneously , the system (4) , an equation each , the computation will be ended in a time equal with that necessary for a single division. So , if F_p denotes the parallel algorithm corresponding to the method just presented ,

$$CP(F_p) = n CP(+) + 3n CP(*) + CP(/) \quad (9)$$

which , at Felix C-256 means:

$$CP_1(F_p) = (81.30\ n + 27.30)\mu \qquad (10)$$

To measure the performance of the parallel algorithm , we can use the theoretical indices presented at the begining of this paper. So , taking into account also (7) and (9) ,

$$s(F_p;n+1) = \frac{(n(n+1)/2)\ CP(+) + n(n+3)\ CP(*) + n\ CP(/)}{n\ CP(+) + 3n\ CP(*) + CP(/)} > n \qquad (11)$$

and

$$e(F_1) = \frac{s(F_p;n+1)}{n+1} < 1 \qquad (12)$$

At Felix C-256 ,

$$s(F_p;n+1) = \frac{n(28.45\ n + 104.55)}{81.30\ n + 27.30}$$

and

$$e(F_p) = \frac{n(28.45\ n + 104.55)}{(n+1)(81.30\ n + 27.30)}$$

From (11) and (12) it is easy to observe that the time involved in performing the parallel method is almost n times less than that necessary for the serial method . If , instead of F , we'll try to find the best serial algorithm which solve our problem , the efficiency of corresponding parallel algorithm will be nearest the maximum value , i.e. the unit.

REFERENCES

1. Calude,C., Complexitatea calculului . Aspecte calitative, Ed.St. si Enciclopedica , Bucuresti , 1982

2. Coman,Gh.,Chiorean I., On the Efficiency of some Parallel simultaneous Methods for the Approximations of Zeros of Polynomials , Research Seminars , Seminar on Complexity of Algorithms , nr.10 ,1989

3. Chiorean,I.,A parallelization of the modified Bernoulli's method for evaluation of zeros of polynomials , Research Seminars , Seminar on Complexity of Algorithms , nr.10 , 1989

4. Rosculet M.,Balea P.,Moldoveanu S., Programarea si utilizarea masinilor de calcul si elemente de calcul numeric si informatica , Ed.Didactica si Pedagogica , Bucuresti , 1980

5. Modi,J.J., Parallel Algorithms and Matrix Computation , Clarendon Press , Oxford , 1988

6. Traub,J.F.(ed) Complexity of sequential and parallel algorithms , Acad.Press ,1973

UNIVERSITY OF CLUJ-NAPOCA

Faculty of Mathematics

3400 Cluj-Napoca