

Design of High Order Equation of One Element System with Parallel Computation

Zhanlin Yu^{1,a}, Liying Wang^{1,b}, Chunqiang Li^{2,c}

¹School of Information Science and Technology, Bohai University, Jinzhou, 121013, China

²School of Information Management, Beijing Information Science and Technology University, Beijing, 100101, China

^a284189433@qq.com, ^b986558686@qq.com, ^c642510571@qq.com

Keywords: parallel algorithms design; data parallel design; high order equation of one element

Abstract. Parallel computing has obvious advantages in improving the computing speed, and it is widely used in large scale computing. In order to solve the problem that the speed of the traditional algorithm is affected by the initial value, this article designs a parallel computing system for solving the equation of high order. First of all, this paper describes the composition of the new algorithm, the process is described in detail; secondly, the parallel computing of the new algorithm is introduced in detail, and the architecture of parallel system is designed by using the hierarchical parallel model; finally, the iterative data is divided into blocks, and the data segmentation results and the processes are introduced. The design method of this paper can effectively improve the computing speed of processing massive data. The system has a strong applicability.

Introduction

The solution of one element higher order equation is more and more important in engineering calculation problems such as electronic system calculation [1], elastoplasticity problem [2] and circuit problem [3]. Its solution is usually iterative method, terminating in accuracy, and thus gets the approximate minimum. The classic method of Newton iteration method [4], choose a near optimal solution of the initial value can be fast convergence of the approximate solution is obtained, but for complex equation and transcendental equations, selection the initial value is very difficult, often adopt simplified system or method to reduce the size of the system, the calculation results can not meet the requirements of practical applications. In order to obtain an algorithm suitable for large-scale engineering computation, the parallel computation method is adopted to improve the computation speed. Parallel computing [5] is not a new technology. It has great influence in many fields such as computational simulation of scientific and engineering application, data mining of business application and transaction processing, including internal combustion engine design [6], wing design, disaster prediction and so on. Distributed computing [7], grid computing, high-performance computing [8] and other technologies [9] are related to parallel computing.

In order to make the algorithm have a better convergence rate, we can obtain all the real solutions of the univariate higher-order equations without selecting the initial values, and propose a new algorithm to solve the high order equation of one element with parallel computation. Parallel computing is a problem divided into several sub problems, and these sub problems in a parallel way at the same time, so as to improve the computing speed and solve the problem. In order to extend the scope of application, a parallel mode of function parallelism, time parallelism and data parallelism is used to parallelize the design of parallel system. Based on C++ programming language, the application of the implementation of process control as a special task design, the application of the implementation process more diverse, more flexible application design. So as to enhance the parallel computing model and the versatility of the system.

The composition of the new algorithm

Set the form of a high-order equation

$$H(x) = \sum_{i=0}^n a_i x^i = 0 \quad (1)$$

where $H(x)$ is a nonlinear mapping of R^n to R^n , $H(x): x \in D \subset R^n \rightarrow R^n$ is continuously differentiable over D , and the derivative of (1)

$$h(x) = \sum_{i=0}^n i a_i x^{i-1} = 0 \quad (2)$$

all of its real solutions are x_1, x_2, \dots, x_m . Therefore, the interval of all real solutions of the formula (1) can be determined, which is $(-\infty, x_1), (x_1, x_2), \dots, (x_m, +\infty)$. The interval is represented by an endpoint representation $X = [\underline{X}, \overline{X}]$, \underline{X} denotes the lower bound of the interval X , \overline{X} denotes the upper bound of the interval X . The interval width is defined as $\omega = \overline{X} - \underline{X}$, for any real number solution in the formula (1), $x \in [\underline{X}, \overline{X}]$.

Commonly used methods for solving the real roots is the dichotomy, tangent method, etc. In this method, the endpoints of the interval are used as the initial value, and the approximate solution is searched. However, the convergence rate of this kind of method is very slow. In the process of solving the problem, it is often unable to meet the requirements of time and high efficiency. The golden section method is widely used in the optimization calculation because of its simple and fast convergence speed, and it is the basis of many optimization algorithms, so this paper quotes the golden section method for iteration. According to the golden section method, $|H(\underline{X})| > |H(\overline{X})|$, $x^* = \underline{X} + 0.382\omega$, $\underline{X} = x^*$. If $|H(\underline{X})| < |H(\overline{X})|$, $x^* = \underline{X} + 0.618\omega$, $\overline{X} = x^*$. This time interval can search reduced 0.382 times or 0.618 times, until it shrinks to a point, then can get a real solution. In order to realize data parallelism, the data will need to be divided according to the column. The j -th column has n data, then $\sum_{j=1}^k n_j = n$, for $j = 1, 2, \dots, k$, define sequence $\{x_j^t\}$, $\{z_j^t\}$, $z_j^t = x_j^{t+1} - x_j^t$, z_j^t is the correction value of the b -th column. $z^t = (z_1^t, z_2^t, \dots, z_k^t)^T$, t represents the iteration solution of the t -th time. $r_j^t = H_j(x^{t+1}) - H_j(x^t)$, r_j^t is the residual of the column j component of the equation. The algorithm for solving the equation of one element is shown in Fig. 1.

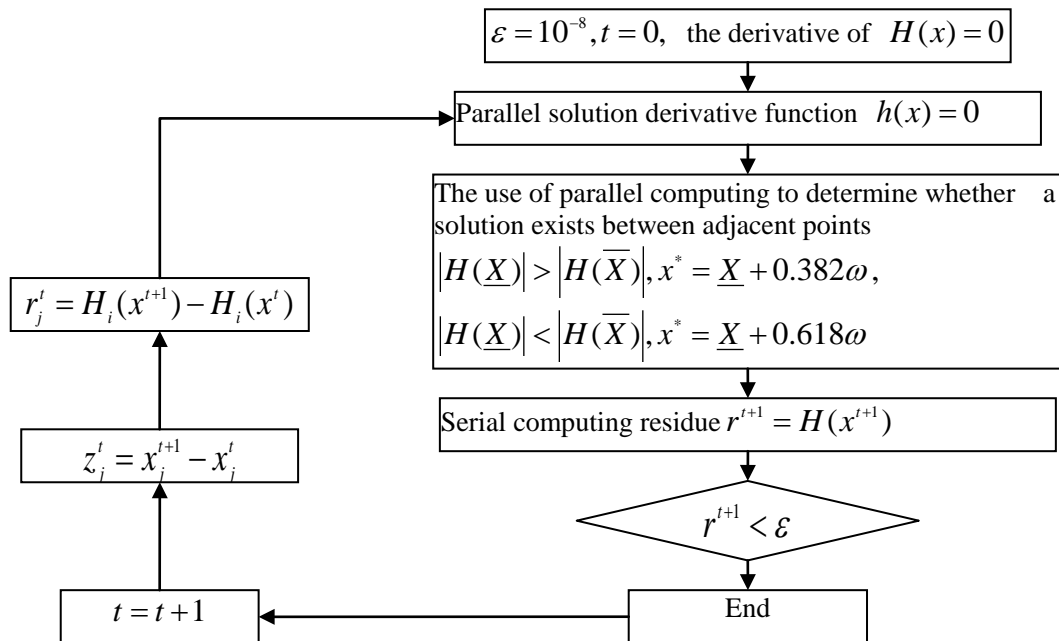


Fig. 1. parallel algorithms

From the algorithm block diagram can be seen, can not be used in parallel computing to a higher degree equation derivation and calculation parameters of $r^{t+1} = H(x^{t+1})$. The rest of the steps can be used in parallel computing. The residual r^{t+1} in the calculation is not dominant, this algorithm is close to the acceleration of the number of processes. The data communication time, from algorithm block diagram can see only in the calculation of residual type is needed when communication between processes, all the solutions of x^t and correction of e^t can be obtained by calculation, compared with the whole calculation part of the execution time can be ignored.

Parallel Architecture Design

The system is designed to ensure that in solving complex univariate polynomial equation in the process of data processing capacity is higher, and has applicability. Using data parallel [10], function in parallel [11] and time parallel [12] layered parallel mode allows the system to process multiple types of problems. Used for system programming C++ programming language, security flexibility in program design, iterative implementation of data aggregation and branch function. The model design is as follows. Firstly, the function of parallel placed at the top, make sure each function execution sequence and the effective region, to ensure a complex higher degree equation of parallel processing. In addition, for time parallelism, many of the functions that it contains are data dependent. From the process of concurrent execution of point of view, time is a subset of parallel parallel function therefore, it will be set in parallel. Data parallelism has sufficient independence to support data parallelism as long as the input data for each process is met. Data parallelism does not affect the other parallel. Finally the data under the bottom placed in parallel. The design of parallel layered architecture is shown in Fig. 2.

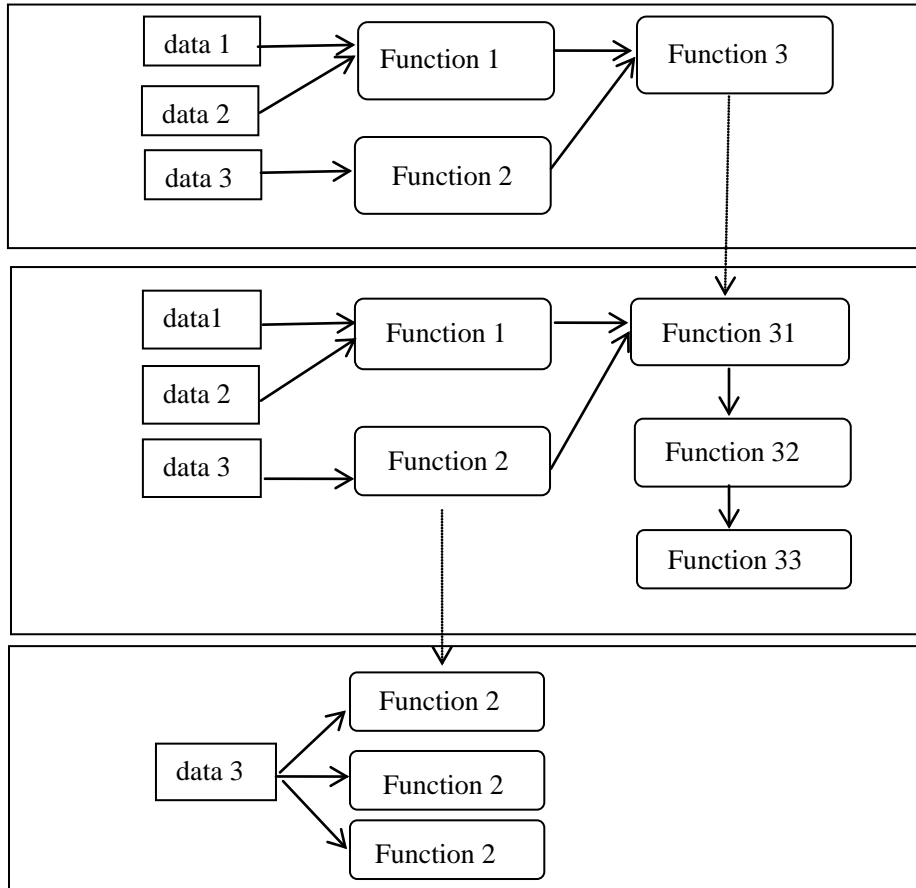


Fig. 2. Parallel layered architecture

Data Parallel Design

In order to realize the data parallel, the data will be divided into columns. A total of n processes for parallel computing at the same time. There are s data for iteration, which are stored in two-dimensional array, and then $s = kn$. According to the result of data segmentation, it is known that the data distribution in n different processes are as follows: Process 0 is $A(n, 1:k)$, Process 1 is $A(n, k+1:2k)$, and so on. In the iterative process, the computation of the new boundary points requires the data of the other blocks of the neighboring boundary. Therefore, in order to store data obtained from adjacent data blocks, each data block on both sides of the increase in an array. Then the communication and computation process is changed into, first on the univariate Polynomial Equation Derivation all point, interval of two adjacent stagnation point determined is the original equation. Then through the real solution before iteration, the method of golden section iterative process, each process from adjacent data blocks, and each process adjacent to the process of data blocks. Taking $n = 4$ as an example. The data segmentation results and its corresponding process are shown in Fig. 3.

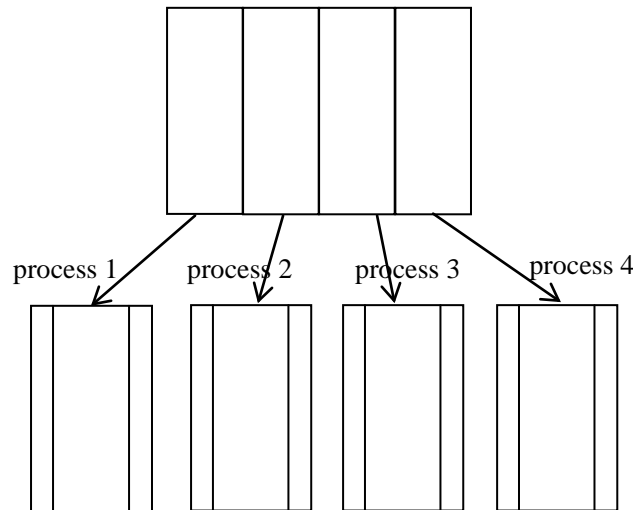


Fig. 3. Data segmentation results and its corresponding process

Its parallel algorithm is as follows.

```

build A[s][s],B[s][s],C[N]
build process[n]
do K=1,step
  do n=1,step
    process[n]=A[n]+(A[n]-A[n+1])/(f(A[n])/f(A[n+1]-1))
  end do
  do l=1,N
    C[N]=process[n]
  end do
end do

```

Conclusion

This article adopts the golden method to iterate, based on Parallel computing system architecture, purpose is to solve the traditional method to solve the problems in the high degree univariate polynomial equation, in order to improve the computing speed. Newton's method for solving a polynomial equation of the traditional element, need to calculate the Jacobian matrix. For complex practical problems, it will produce a large amount of computation, and it is difficult to use parallel computing. The parallel computing system of a high order equation in this paper, the structure is clear, which can be used to realize parallel computation and improve the computing speed. This parallel computing system has the universal applicability, which can be applied to large-scale high performance computing.

References

- [1] Caleb J.Ball, Peter M.W.Gill, "a software package for electronic structure calculations on one-dimensional systems," *Molecular Physics*, vol. 113, no. 13, pp. 1843-1857, 2015.
- [2] P.D.Howell, H.Ockendon, J.R.Ockendon, "An asymptotic model for elastoplasticity," *Journal of Engineering Mathematics*, vol. 84, no. 1, pp. 57-71, 2014.
- [3] W. H. Ruan, Q. S. Hu, "A 14.5Gb/s word alignment circuit in 0.18 μ m CMOS technology for high-speed SerDes," *High Technology Letters*, vol. 20, no. 3, pp. 328-332, 2014.
- [4] Thukral R, "New Modification of Newton Method with Third-Order Convergence for Solving Nonlinear Equations of Type $f(0)=0$," *American Journal of Computational and Applied Mathematics*, vol. 6, no. 1, pp. 5, 2016.

- [5] W. J. Li, Y. Z. Bi, X. F. Zhu, etc. "Hybrid Swarm Intelligent Parallel Algorithm Research Based on Multi-core Clusters," *Microprocessors and Microsystems*, 2016.
- [6] G. Mohamed, A. A. Sofiane, L. Nicolas, "An Adaptive Integral Sliding Mode Control Design for Internal Combustion Engine Air Path," *IFAC Proceedings Volumes*, vol. 46, no. 25, pp. 46-49, 2013.
- [7] Z. J. Liu, A. X. Zhu, C. Z. Qin, etc, "Progress on Parallel Studies of the Distributed Hydrological Models," *Progress in Geography*, vol. 32, no. 4, pp. 538-547, 2013.
- [8] I. Masatomo, U. Nobuyuki, S. Ryohei. "Shrinking sphere: A parallel algorithm for computing the thickness of 3D objects," *Computer-Aided Design and Applications*, vol. 13, no. 2, pp. 199-207, 2016.
- [9] A. Alisa. A. David, "Activating the Invisible: Data Processing and Parallel Computing in Architectural Design," *Intelligent Buildings International*, vol. 8, no. 2, pp. 106-117, 2016.
- [10] H. K. Kashif, Q. Kalim, A. Mostafa, "An Efficient Grid Scheduling Strategy for Data Parallel Applications," *The Journal of Supercomputing*, vol. 68, no. 3, pp. 1487-1502, 2014.
- [11] G. B. Guo, L. Lin "Parallel Bootstrap and Optimal Subsample Lengths in Smooth Function Models," *Communications in Statistics - Simulation and Computation*, vol. 45, no. 6, pp. 2208-2231, 2016.
- [12] N. M. Yao, J. Z. Chen, S. B. Cai "A Non-partitioning File Assignment Scheme with Approximating Average Waiting Time in Parallel I/O System," *Journal of Software*, vol. 8, no. 2, pp. 302-309, 2013.