Application Research Based on Finite Difference Algorithm

Ruojing Tong [1,*], Xinyu Qian [2], Xinlan Xiang [3]

[1] College of Physical Science and Technology, Dalian University, Dalian, China, 116622

[2] College of Information Engineering, Dalian University, Dalian, China, 116622

[3] International College, Dalian University, Dalian, China, 116622

* Corresponding Author Email: 13164553314@163.com

Abstract. This paper discusses the importance of finite difference algorithms in the field of science and engineering and focuses on methods for both optimization of algorithm efficiency and numerical stability improvement. For algorithm efficiency, optimization methods such as parallel computing techniques, optimized mesh partitioning, selection of appropriate difference formats, adaptive mesh methods, use of high-performance computing platforms, and pre-processing and post-processing techniques are introduced. And for numerical stability, methods to improve the numerical stability of the algorithm are discussed. Through these optimization and improvement methods, the finite difference algorithm can respond to the challenges of complex problems more efficiently and reliably, and provide a more reliable and efficient numerical computation method for scientific research and engineering practice.

Keywords: Finite difference algorithm, Algorithm efficiency optimization, Numerical stability, Parallel computing, Adaptive grid.

1. Introduction

As an important numerical computation method, finite difference algorithms have been widely applied across various scientific and engineering disciplines to solve numerical approximate solutions for differential and partial differential equations. These applications range from geophysical exploration, electromagnetic field computation, fluid dynamics, to heat transfer, among others. However, as the complexity and scale of problems increase, the computational efficiency and numerical stability of finite difference algorithms have become significant challenges that need to be addressed to ensure reliable and accurate results. The issues of computational efficiency and numerical stability in finite difference algorithms are not new and have been recognized in the literature. For instance, researchers have noted that traditional finite difference methods may suffer from stability restrictions, such as the Courant-Friedrichs-Lewy (CFL) condition, which limits the time step size and thus affects the overall computational efficiency. Additionally, the need for high-resolution simulations in complex media, such as those encountered in geophysical exploration, often requires finer grids and smaller time steps, leading to increased computational costs. Furthermore, the numerical stability of finite difference algorithms is crucial to prevent the amplification of numerical errors, which can lead to divergent solutions or non-physical oscillations. The stability of these algorithms has been studied extensively, with various techniques proposed to enhance stability, such as the use of higher-order schemes, implicit methods, and artificial dissipation. However, these methods often come with trade-offs, such as increased computational complexity or potential impacts on the accuracy of the solution. In light of these challenges, the optimization of finite difference algorithms for improved efficiency and stability has been an active area of research. Strategies such as parallel computing, adaptive [1].

2. Basis of Finite Difference Algorithm

2.1. Overview of finite difference method

The finite difference method is a common numerical computation method for solving numerical approximate solutions of differential equations and partial differential equations. The method discretizes a continuous differential equation in space, divides the region into discrete grid points, and approximates the derivative term in the differential equation in difference form. The basic idea of the finite difference method is to approximate the derivatives by the difference of the function values on the grid, and by means of this discretization the differential equations are transformed into algebraic equations, which can be solved using a computer [2].

A common step is to divide the solution region into a finite number of grid points, usually including a uniform or non-uniform grid. The derivatives in the differential equations are approximated by discretization using forward, backward or central difference formulas. Appropriately treat the grid points at the boundary to satisfy the given boundary conditions. Solve the discretized system of algebraic equations using iterative or direct solution methods [3].

Post-processing of the numerical solution obtained from the solution, such as visualization, error analysis, etc. Finite difference methods are widely used in science and engineering fields, including but not limited to numerical simulation and computation of problems such as geophysical exploration, electromagnetic field computation, fluid mechanics, heat transfer, and so on. Its advantages include simple and easy implementation, wide range of applicability, and parallelizability, and thus it has been widely used in practical engineering and scientific computation.

2.2. Numerical stability and convergence analysis of finite difference algorithm

As a common numerical computation method, the finite difference algorithm plays an important role in the process of solving the numerical approximate solutions of differential equations and partial differential equations. However, in order to ensure that the algorithm obtains accurate and reliable results, its numerical stability and convergence must be carefully analyzed and evaluated. In terms of numerical stability, the accumulation of numerical errors in the algorithm during the computation process needs to be considered. Specifically, it is desired that the numerical solution of the algorithm will not diverge or oscillate due to the accumulation of numerical errors. Therefore, stability analysis methods, such as using frequency domain analysis or energy methods, are often employed to assess the stability properties of the algorithm. The stability conditions of the algorithm can be determined by analyzing the eigenvalues of the difference format or by calculating stability bounds to ensure the stability of the numerical solution. In terms of convergence, it is desired that the numerical solution of the algorithm gradually approaches the true solution. In order to evaluate the convergence of the algorithm, two methods, grid convergence analysis and stepwise convergence analysis, are usually used. Grid convergence analysis observes the change in the numerical solution by gradually decreasing the grid spacing to determine the convergence of the algorithm under different grids. Step convergence analysis, on the other hand, observes the change in the numerical solution by gradually decreasing the time step or space step to determine the convergence of the algorithm at different step sizes. Through these analysis methods, the convergence order and convergence speed of the algorithm can be derived to assess the numerical convergence of the algorithm [4].

3. Application areas of finite difference algorithms

3.1. Finite difference algorithm in geophysical exploration

3.1.1 Seismic wave simulation

In the field of geophysical exploration, seismic wave simulation is a crucial task, which plays a key role in revealing the underground structure and detecting resources such as oil and gas. Seismic wave simulation is not only a simple simulation of the process of seismic wave propagation in the subsurface, but also an effective means to understand the internal structure of the earth and the nature of the subsurface medium. The process of seismic wave simulation can be viewed as abstracting the physical properties of the subsurface medium into a mathematical model, and simulating the propagation of seismic waves in the medium by means of a finite-difference algorithm. In this process, the numerical stability and convergence of the finite difference algorithm are particularly important. Numerical stability ensures that the algorithm will not produce unreasonable results due to the accumulation of numerical errors in the computation process, while convergence ensures that the numerical solution obtained by the algorithm can approximate the real physical phenomena. In order to realize the seismic wave simulation, it is first necessary to establish the velocity model of the subsurface medium, which is usually obtained by modeling the geological information obtained from seismic exploration and geophysical observation. Then, the subsurface medium is divided into a grid and the seismic fluctuation equation is solved numerically using a finite difference algorithm [5]. In this process, factors such as the interaction between seismic waves and the medium, and the nonlinear properties of the medium need to be considered. In seismic wave simulation, the selection of the finite difference algorithm and parameter settings have an important impact on the accuracy and reliability of the simulation results. Therefore, the numerical stability and convergence of the finite-difference algorithm need to be fully analyzed and evaluated when simulating seismic waves. Only on the basis of ensuring the stability and convergence of the algorithm can we get accurate and reliable seismic wave simulation results, which can provide strong support for earth science research and resource exploration.

3.1.2 Seismic data processing

Seismic data processing is an indispensable part of geophysical exploration, which is aimed at extracting useful information from seismic records and revealing underground structures and geological features. Firstly, seismic data processing usually involves pre-processing of the raw seismic record, which includes steps such as de-noising, removal of instrumental response, calibration and time-frequency domain conversion. These preprocessing operations are designed to improve the quality and accuracy of seismic data and lay the foundation for subsequent analysis. Second, seismic data processing also includes the analysis and interpretation of seismic waveforms [6]. This involves the analysis of the amplitude, frequency, phase, and other characteristics of seismic waveforms, as well as the inference and modeling of the structure of the subsurface medium. In this process, finite difference algorithms are often used to simulate the propagation of seismic waves through the subsurface, thus helping to interpret features in the seismic record. Finally, seismic data processing also includes the application of seismic imaging, inversion, and imaging techniques. By comprehensively analyzing and processing seismic data, these techniques can reveal the distribution, nature, and characteristics of subsurface structures, providing an important basis for geological exploration and resource development. The numerical stability and convergence of finite difference algorithms are also crucial when performing seismic data processing. Because seismic data processing often involves the processing of large-scale data and the simulation of complex underground media, it requires efficient and reliable numerical methods to realize.

3.2. Finite Difference Algorithms in Electromagnetic Field Calculations

Electromagnetic field calculation is one of the important methods to study the propagation and interaction of electromagnetic waves in different media. A mathematical model of the electromagnetic field needs to be established according to the requirements of the specific problem, taking into account the electromagnetic properties of the medium, boundary conditions and other factors. This step usually involves a system of Maxwell's equations or other appropriate electromagnetic theory. The solution region is divided into discrete grid points that are uniformly or in homogeneously distributed in space. Typically, the grid in 3D space is usually a cubic grid in a Cartesian coordinate system, but other forms of grids can be used as needed. The partial differential equations in the EMF model are discretized using a difference formulation. For the electric and magnetic fields, the approximation can be done by using center difference, forward difference, or backward difference, respectively. This step transforms the differential equations into difference equations, making the problem into solving a system of algebraic equations. Appropriate treatment is applied at the boundaries of the model to satisfy the given boundary conditions [7]. The boundary conditions usually include the values of the electric and magnetic fields at the boundary or the derivative values. Solve the discretized system of difference equations using numerical iterative methods. Common solution methods include iterative methods, direct methods, etc., of which iterative methods such as the iterative solution method and the conjugate gradient method are often applied to the solution of large and complex problems.

Post-processing of the electromagnetic field data obtained from the solution includes visualization, data analysis, and validation of simulation results. These steps help to analyze and interpret the results of electromagnetic field calculations to draw conclusions about the corresponding problems.

3.3. Finite Difference Algorithms in Fluid Dynamics

3.3.1 Fluid dynamics simulation

Fluid dynamics simulation is one of the important methods to study fluid motion and corresponding physical phenomena, which is widely used in engineering, meteorology, earth science and other fields. A mathematical model of the fluid needs to be established according to the requirements of the specific problem, taking into account factors such as the nature of the fluid, boundary conditions and external force fields. Common fluid models include systems of fluid dynamics equations such as the Navier-Stokes equations. The solution region is divided into discrete grid points and the parameter fields of the fluid, such as velocity, pressure, temperature, etc., are established at the grid points. Typically, the grid can be structured or unstructured, and the selection of an appropriate grid structure can effectively affect the accuracy and efficiency of the simulation results. The fluid dynamics equations are discretized using the difference format. As shown in Figure 1. For the Navier-Stokes equations, the partial differential equations can be approximated by using the center difference and the windward difference to transform the partial differential equations into a system of algebraic equations. Time-stepping methods are used to solve the fluid model by discretizing the time continuity equations into a series of algebraic equations over time steps. Commonly used time-stepping methods include Euler implicit methods, Runge-Kutta methods, etc [8]. The boundary conditions of the fluid model are appropriately treated to ensure that the fluid parameters at the boundary satisfy the given conditions, such as velocity, pressure, and temperature. Numerical iterative methods are used to solve the discretized set of algebraic equations to obtain the numerical solution of the flow field at each time step. Common solution methods include the iterative method, conjugate gradient method, and so on. The flow field data obtained from the solution are post-processed, including visualization, data analysis, and extraction of flow field properties. These post-processing steps help to deeply understand the fluid motion laws and the structural characteristics of the flow field.

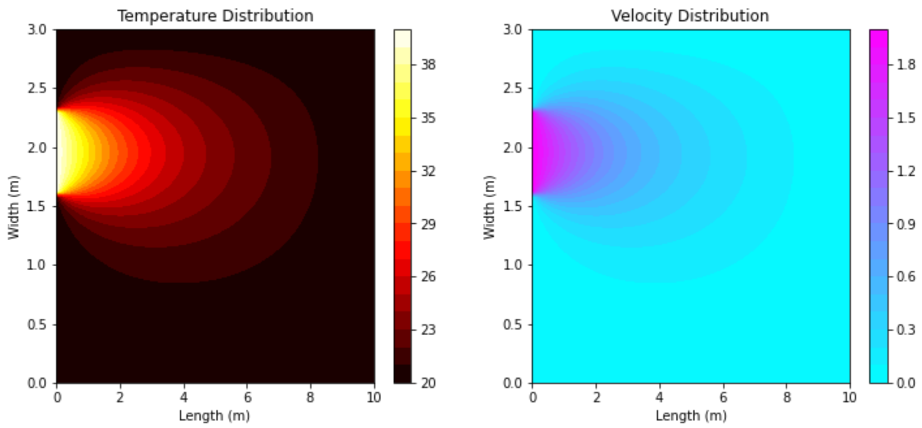

Figure 1. Example diagram

3.3.2 Heat Transfer Simulation

Heat conduction simulation is an important method to study the temperature distribution and heat conduction process inside an object, which requires the establishment of a mathematical model according to the characteristics of the heat conduction problem, taking into account the thermal properties of the material, boundary conditions and heat sources. Common heat conduction models include heat conduction equation, energy conservation equation and so on. The solution region is divided into discrete grid points and the temperature field is established at the grid points. Typically, the grid can be structured or unstructured, and the selection of an appropriate grid structure can affect the accuracy and efficiency of the simulation results. The heat conduction equation is discretized using a difference format. The partial differential equations can be approximated by using central difference, forward difference or backward difference to transform them into a system of difference equations. Appropriate treatment is applied at the boundaries of the model to ensure that the temperatures at the boundaries satisfy the given boundary conditions. The boundary conditions usually include conditions such as fixed temperature, heat flow or convective heat transfer [9]. Time-stepping methods are used to solve heat transfer models by discretizing the time continuity equations into a series of algebraic equations over time steps. Commonly used time-stepping methods include explicit Eulerian methods, implicit Eulerian methods, etc. Numerical iterative methods are used to solve the discretized set of algebraic equations to obtain the numerical solution of the temperature field at each time step. Common solution methods include the iterative method, conjugate gradient method, etc. As shown in Figure 2. The temperature field data obtained from the solution are post-processed, including visualization, data analysis, and extraction of temperature distribution. These post-processing steps help to gain a deeper understanding of the temperature distribution and heat transfer process inside the object. The finite difference algorithm plays an important role in heat conduction simulation and provides an efficient and reliable numerical calculation method for studying the temperature distribution and heat conduction process inside the object.

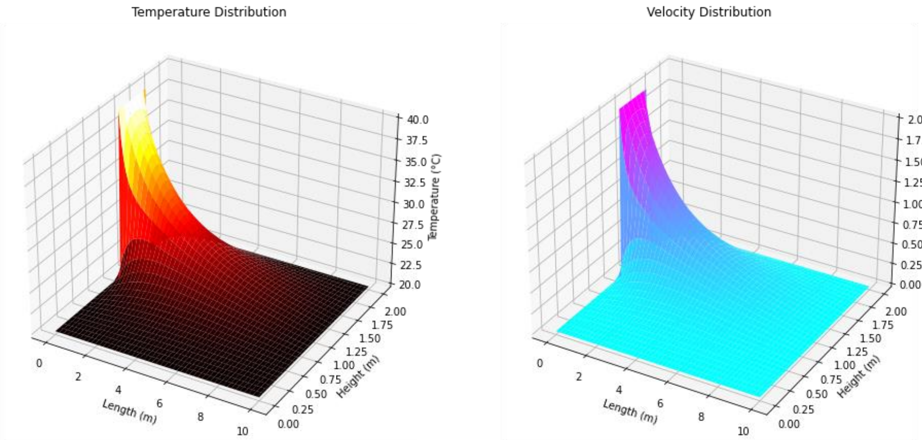

Figure 2. Example diagram

4. Optimization and Improvement of Finite Difference Algorithms

4.1. Algorithm efficiency optimization method

As a commonly used numerical computation method, the finite difference algorithm often needs to face large-scale and complex problems in practical applications, so the efficiency optimization of the algorithm is particularly important. Using parallel computing techniques, such as MPI (Message Passing Interface) and OpenMP, the problem is decomposed into multiple sub-problems and multiple processors or computing nodes are utilized to handle these sub-problems simultaneously. The computational speed and efficiency of the algorithm can be significantly improved by parallel computing techniques. Reasonable selection of grid division methods, including structured and unstructured division of the grid, and the way of arranging grid points, can reduce the computational complexity and storage requirements of the algorithm, thus improving the efficiency of the algorithm. According to the characteristics of the problem and the requirements of the solution accuracy, choose the appropriate difference format. Sometimes, it may be more efficient to use the first-order difference format without solving the higher-order difference equations. Through the adaptive mesh technique, the mesh division and the difference step size are dynamically adjusted according to the local characteristics of the solution and the error distribution. This can minimize the amount of computation and improve the efficiency of the algorithm while ensuring the accuracy of the solution. The computational speed and efficiency of the finite difference algorithm can be further improved by utilizing high-performance computing platforms, such as GPU (graphics processing unit) gas pedals. Parallel computing and acceleration of the computational process is achieved by making full use of hardware resources. Through the pre-processing and post-processing of the problem, such as data compression and dimensionality reduction processing, the amount of computation and storage requirements can be reduced and the efficiency of the algorithm can be improved. Reasonable application of parallel computing technology, optimization of grid division, selection of appropriate difference formats, use of adaptive grid methods, use of high-performance computing platforms and pre-processing and post-processing techniques can effectively improve the computational efficiency of finite difference algorithms and provide faster and more reliable numerical computation methods for solving practical problems.

4.2. Algorithm numerical stability improvement

In the application of the finite difference algorithm, choosing the appropriate grid size and grid division has an important impact on the numerical stability. Generally speaking, when the characteristic scale of the problem varies greatly, non-uniform grid division should be used to ensure a higher grid density in important regions, so as to improve the accuracy and stability of the numerical solution. The stability of the numerical solution is ensured by employing appropriate time-step control strategies, such as the CFL (Courant-Friedrichs-Lewy) condition, which limits the time-steps to ensure that the information does not propagate too fast between neighboring time-steps, thus avoiding instability in the numerical solution. In some cases, high-frequency oscillations and instability of the numerical solution are suppressed by introducing artificial damping or numerical dissipation terms. This technique can improve the numerical stability of the algorithm to a certain extent, but it can also have an impact on the accuracy of the numerical solution, so it needs to be chosen carefully. Implicit difference methods provide better numerical stability compared to explicit methods, especially when dealing with problems with a high degree of nonlinearity or coupling. Therefore, in some cases, the numerical stability and convergence of the algorithm can be improved by using implicit methods [10].

The accuracy and stability of the numerical solution can be improved by dynamically adjusting the mesh division and the difference step size according to the local characteristics of the solution and the distribution of the error by means of the adaptive mesh method. Error analysis and convergence test are performed to assess the accuracy and stability of the numerical solution. In this way, the instability and source of error of the numerical solution can be found in time, and corresponding measures can be taken to improve it.

5. Conclusion

This paper explores the application of finite difference algorithms in the field of science and engineering, and discusses both optimizing the efficiency of the algorithm and improving the numerical stability. As a common numerical computation method, finite difference algorithms play an important role in solving numerical approximate solutions of differential and partial differential equations. The computational efficiency of the algorithm can be effectively improved by means of parallel computing techniques, optimized mesh division, selection of appropriate difference formats, adaptive mesh methods, use of high-performance computing platforms, and pre-processing and post-processing techniques. At the same time, by improving the numerical stability of the algorithm, it can ensure that the algorithm will not produce unreasonable results due to the accumulation of numerical errors during the calculation process, thus improving the reliability and accuracy of the algorithm.

In future research, more efficient and reliable finite difference algorithms can be further explored and developed, and applied to a wider range of fields, such as geophysical exploration, electromagnetic field computation, fluid dynamics and heat transfer. By continuously optimizing and improving the algorithms, the challenges of complex problems can be better met and more reliable and efficient numerical computation methods can be provided for scientific research and engineering practice.

6. Future Research Directions for Finite Difference Algorithms

The future of finite difference algorithms lies in addressing the challenges outlined in the previous sections and pushing the boundaries of computational science and engineering. Adaptive mesh refinement (AMR) is a technique that has shown promise in improving the efficiency and accuracy of finite difference algorithms. By dynamically adjusting the grid resolution based on the local solution features and error estimates, AMR can focus computational resources where they are most needed, reducing the overall computational cost while maintaining high solution fidelity. Future research could focus on developing more sophisticated AMR strategies that are capable of handling problems with multiple scales and complex geometries. Machine learning techniques can be employed to optimize the parameters of finite difference algorithms, such as the choice of time step or spatial discretization, potentially leading to more efficient and accurate simulations. Additionally, AI can be used to identify patterns in large datasets generated by these simulations, aiding in the interpretation and prediction of complex physical phenomena. The continued development of high-performance computing (HPC) platforms, including GPUs and specialized accelerators, offers unprecedented opportunities for finite difference algorithms. Research in this area should focus on optimizing the implementation of finite difference algorithms on these platforms, leveraging their parallel processing capabilities to solve larger and more complex problems. This includes the development of novel parallel algorithms and the adaptation of existing ones to take full advantage of the hardware's performance characteristics. Finite difference algorithms have been traditionally applied within the domains of physics and engineering. However, there is a growing interest in exploring their potential in other fields, such as biology, finance, and social sciences. Future research could investigate how these algorithms can be adapted and applied to model complex systems in these areas, potentially leading to new insights and discoveries. As finite difference algorithms become more sophisticated, it is essential to ensure that they remain transparent and explainable. Research should focus on developing methods to validate and verify the algorithms, providing users with confidence in the results and the ability to understand the underlying processes. This is particularly important in fields where the consequences of errors can be significant, such as climate modeling or structural engineering. With the increasing reliance on finite difference algorithms for critical applications, it is imperative to consider the ethical implications of their development and use. Future research should address issues such as data privacy, algorithmic bias, and the responsible use of computational resources. This includes the development of guidelines and best practices for the ethical design and deployment of these algorithms.

References

[1] Yang Chen, Gongxue Zhang & Yao Wu. (2023). Finite difference algorithm for static characterization of multileaf dynamic gas plain bearings. Lubrication and Sealing (10), 157-164.

[2] Yang, Bohan. (2023). Finite difference algorithms for linear elasticity problems on staggered grids. Journal of Computational Mathematics for Higher Education (01), 1-15.

[3] Xiaosen Kang, Hongjian Liao, Qiangbing Huang, Zuochen Wang & Zongyuan Ma. (2023). Finite difference algorithm and its numerical validation for double-shear uniform boundary surface plasticity model. Journal of Rock Mechanics and Engineering (01), 65-75.

[4] Tang, D. S.. (2022). Unconditionally optimal point-by-point error estimation for finite difference algorithms for two-dimensional complex-valued Ginzburg-Landau equations M.S. Dissertation, Nanjing University of Information Engineering).

[5] Zhang, Zhiyuan. (2022). Extrapolated linearized finite difference algorithm for Rosenau-RLW equations M.S. Dissertation, Xihua University).

[6] Zang, N. (2022). Research on Docking Grid and Nested Grid Finite Difference Algorithms for Seismic Wave Simulation in Complex Terrain Ph.D. Dissertation, University of Science and Technology of China).

[7] Li, Xuan Yu. (2021). Runoff simulation based on improved finite difference algorithm ELM model. Soil and Water Conservation Applied Technology (03), 18-20.

[8] Jijun Li & Zhuan Xiao. (2020). Application of finite difference algorithm in teaching guided wave pattern visualization. Journal of Heze College (05), 114-120.

[9] H.B. Chen. (2020). Parallel Optimization of Finite Difference Algorithms Based on Heterogeneous Multi-Core Processors Master's Degree Thesis, JNU).

[10] Zhili Dai, Zhongfu Xiang & Hui Zhou. (2020). Matrixed finite difference algorithm for acoustic wave equation simulation efficiency analysis. Journal of Chongqing Jiaotong University (Natural Science Edition) (01), 67-71.