Frontier Interdisciplinary Strategy

Cutting-edge Cross-disciplinary Research on Mathematical Software

Cutting-edge Cross-disciplinary Research on Mathematical Software Research Team

Funded by National Natural Science Foundation of China and Chinese Academy of Sciences

2024

Content

Introduction

With the accelerated development of manufacturing in sectors such as aviation, aerospace, shipbuilding, nuclear industry, and military industry in China, as well as the ongoing industrialization process, the need for domestically controllable critical industrial software has become urgent. This is especially significant in the context of building a new domestic-international dual circulation development pattern. It has become one of the "bottleneck" issues. For example, in May 2020, the U.S. Department of Commerce expanded its export control "Entity List," banning 13 Chinese universities, including the National University of Defense Technology and Northwestern Polytechnical University, from using the computer simulation software MATLAB. In August 2022, the U.S. implemented export controls on integrated circuit EDA toolchains for design, physical simulation, and manufacturing processes, citing "national security" concerns. Many major industrial software packages rely on mathematical software, so research on the cutting-edge intersections of mathematical software is of great significance. This report summarizes and reviews the current status, development trends, and the research foundations and conditions in China, analyzes potential new growth points in this field, and offers recommendations for the development of mathematical software in China.

Chapter 1: Basic Mathematical Software

Basic mathematical software focuses on fundamental operations and functions in scientific computing, covering areas such as matrix operations, numerical algebra, numerical analysis, differential equations, and data analysis. It provides general computational interfaces and programming platforms, serving as the cornerstone of various mathematical algorithms and professional software.

1.1 General Mathematical Software

General mathematical software integrates common mathematical fields and various basic operations. Such software typically offers a visual user interface, a programming environment, a wealth of built-in functions, and flexible expansion interfaces. Due to its wide applicability, high efficiency, and ease of use, it is favored by most scientists and engineers. By using general mathematical software, theoretical results can quickly be converted into practical applications, ultimately leading to products and solutions. Well-known international general mathematical software includes MATLAB, Maple, Mathematica, Octave, SciLab, Julia, and SageMath.

MATLAB, developed by MathWorks in the U.S., is a general-purpose numerical computing software mainly used in fields such as numerical computation, data analysis, signal processing, and image processing. MATLAB uses the M language for programming, an interpreted high-level language similar to R and Python. It provides a wide range of toolboxes and application programming interfaces (APIs), including the Statistics Toolbox, Signal Processing Toolbox, Image Processing Toolbox, Control System Toolbox, Optimization Toolbox, and more.

The advantage of MATLAB lies in its powerful matrix computation capabilities. Its core is written in C/C++, supporting multiple data types, including numerical, character, logical, struct, cell, and function handle types. It also supports object-oriented programming and functional programming. MATLAB excels in data processing and supports various data import/export formats, such as MAT files, text files, Excel files, and image files. Additionally, MATLAB includes thousands of toolboxes for various practical fields.

In terms of user experience, MATLAB has a user-friendly graphical interface that allows for data analysis, plotting, and simulation via the visual interface, command line, or by writing custom functions or scripts to perform complex calculations.

In the market, MATLAB occupies a significant share of the software market, widely used in education, research, engineering, and business sectors.

1.2 Low-level Mathematical Libraries

Low-level mathematical libraries focus on the high-performance implementation of mathematical functions and operations but generally lack integrated development environments and graphical interfaces.

Numerical linear algebra software packages are one of the most basic types of low-level mathematical libraries. The four basic problems in numerical linear algebra are linear systems of equations, least squares problems, eigenvalue decompositions, and singular value decompositions.

The performance of numerical algebra software packages is a key metric, and it must consider different hardware platforms. In addition to single-core, multi-core, many-core, distributed, GPU, and heterogeneous environments must also be considered, with algorithm implementations incorporating techniques such as block operations and mixed precision. Popular programming languages for implementing numerical algebra software packages include assembly, C/C++, FORTRAN, and recently RUST, Numba, and Julia, which are easier for programming.

1.2.1 Dense Matrix Numerical Algebra Libraries

Major dense matrix numerical algebra libraries include BLAS, LAPACK, OpenBLAS, ATLAS, BLIS, Intel oneMKL, Accelerate.Framework, etc. Distributed and heterogeneous versions include cuBLAS/cuSOLVER, MAGMA, ScaLAPACK, and SLATE.

1.2.2 Sparse Matrix Numerical Algebra Libraries

Sparse matrix numerical algebra libraries differ from dense matrix libraries in that they lack unified interface standards due to the close relationship between sparse matrix operations and specific applications. Some well-known sparse matrix libraries include METIS, SuiteSparse, SuperLU, PARDISO, MUMPS, ARPACK, FEAST, and cuSPARSE.

1.2.3 Fast Fourier Transform (FFT) Algorithms

The Fast Fourier Transform (FFT) is a fundamental algorithm in mathematical computing. Key libraries for FFT include FFTW, heFFTe, and cuFFT. FFTW, developed as the "Western Fast Fourier Transform library," is written in C and optimized for different processor architectures.

1.2.4 Integrated Software Packages and Development Frameworks

To improve development convenience, different software packages are often integrated into a unified underlying framework, unifying data structures and simplifying the calling process. Key integrated low-level software packages include EIGEN, STRUMPACK, Kokkos Kernels, Trilinos, Ginkgo, PETSc, etc.

EIGEN is a lightweight C++ numerical algebra template library based on object-oriented programming and template metaprogramming, which defines common objects for numerical computation such as matrices and arrays. Similar development frameworks include Armadillo, BLAZE, etc.

1.2.5 Mixed Precision Software Libraries

The development of mixed precision algorithms has been slower compared to mixed precision software. While some algorithms for solving linear systems with mixed precision have been developed, most mixed precision algorithms for basic problems remain in the theoretical or experimental stages without mature software implementations in major libraries like LAPACK.

In large-scale sparse matrix computations, key mathematical libraries supporting mixed precision include SuperLU, STRUMPACK, Trilinos, Ginkgo, and PETSc. Furthermore, MPI is developing functions that support different precision communication and reduction operations, with significant performance advantages for sufficiently large datasets.

Chapter 2: Mathematical Optimization Software

Mathematical optimization is an important part of applied mathematics and is also one of the foundations of current machine learning and artificial intelligence.

2.1 Current Status of Continuous Optimization Software

Continuous optimization problems can be divided into linear cone programming, nonlinear programming, tensor and polynomial optimization, manifold optimization, sparse optimization, variational inequalities and complementarity problems, and derivative-free optimization.

2.1.1 Linear Cone Programming

Linear cone optimization refers to optimization problems where the decision variables belong to a cone, and both the constraints and objective function are linear functions of the decision variables. Linear cone programming problems have shown a continuous growth trend in scale. Generally, computable linear cone constrained optimization problems include linear programming, second-order cone programming, and semidefinite programming.

Semidefinite programming (SDP) is a widely applied type of linear cone constrained optimization problem. The corresponding algorithms mainly focus on interior-point methods. Currently, commercial optimization software such as COPT, MOSEK, and MindOpt have integrated SDP solvers. There are also dozens of academic software or open-source codes available, primarily based on interior-point algorithms. Among the most well-known are SDPT3 and SeDuMi, which, together with MOSEK, form the three major SDP solvers supported by the famous convex optimization modeling system CVX. Currently, the performance of the domestic solver COPT is leading compared to the SDP software tested on the ASP platform.

2.1.2 Nonlinear Programming

Nonlinear programming refers to optimization problems where the constraints or objective function are nonlinear. It is a core component of computational mathematics.

Quadratic programming, as one of the most significant mathematical programming problems in nonlinear optimization, has received considerable attention. Based on the active-set method, Fletcher developed algorithm packages like BQPD, QPOPT, and Mosek for solving quadratic programming. Xpress, one of the three major commercial optimization solvers globally, can solve quadratic programming and second-order cone programming with millions of decision variables, and is applied to more general quadratic constraint quadratic programming and nonlinear programming. For sparse quadratic programming problems, the software package SQOPT efficiently solves large-scale convex quadratic programming problems with equality and inequality constraints based on sparse structures.

General nonlinear programming software includes DONLP2, LINGO, NPSOL, and GRG2.

For large-scale nonlinear programming problems, typical software packages include Artelys Knitro, LANCELOT, CONOPT, and SNOPT.

2.1.3 Tensor Optimization

Tensor optimization involves optimization problems where tensors are the variables, and tensor analysis is used as the basic research tool. Polynomial optimization focuses on optimization problems where the objective function has a polynomial structure. The SOS (Sum of Squares) polynomial is foundational to the theory and algorithms in polynomial optimization, and it remains an active research area.

Representative software packages for tensor optimization include the Tensor Toolbox and Tensorlab in MATLAB. In recent years, with the rise of artificial intelligence and big data technologies, the optimization software TensorLy has emerged for handling tensor methods and deep neural networks.

For polynomial optimization software, the most well-known package is SOSTOOLS. Wang Jie from the Chinese Academy of Sciences has developed several open-source software packages based on the Julia language, such as TSSOS, SONCSOCP, SparseJSR, and NCTSSOS, which are used for solving sparse polynomial optimization problems.

2.1.4 Manifold Optimization

Optimization on manifolds refers to optimization problems where the constraint set is a smooth manifold. It has important applications in computational science, statistics, machine learning, data science, and materials science. Compared with linear programming and nonlinear programming, the history of manifold optimization is shorter, but in the past decade, corresponding manifold optimization software has been continuously released.

For optimization problems on the Stiefel manifold, the main software packages include OptM and STOP. For Riemannian and general manifold optimization, the software packages include ROPTLIB, ARNT, CDOp, Manopt, RSOpt, ManPG, AManPG, and almssn.

2.1.5 Variational Inequalities and Complementarity Problems

Variational inequality problems frequently arise in game theory, traffic networks, and other areas. Solving variational inequalities includes various methods, such as subspace projection and proximal point algorithms. The most classic projection methods are the outer gradient projection method and projection-contraction algorithms.

As an important branch of variational inequalities, complementarity problems naturally attract attention in the research of their solving algorithms. The software for solving complementarity problems is more extensive and comprehensive than that for general variational inequality problems. While there are many solving methods for complementarity problems, existing algorithm packages are still scattered, and the integration is lacking, which hampers practical usage.

2.1.6 Sparse Optimization

In recent years, research on sparse optimization software and algorithm packages has gained increasing attention from scholars, becoming one of the most active topics in the fields of data science and optimization.

In the design of sparse optimization algorithms, incorporating necessary second-order information can accelerate the algorithms. A representative software based on the proximity point/augmented Lagrangian method and semi-smooth Newton method is SuitLasso.

2.1.7 Derivative-Free Optimization

In practical applications, many optimization problems have objective functions that do not have explicit expressions. Their function values are obtained from physical experiments or extensive computer simulations, and their first-order information is unavailable, unreliable, or difficult to observe. Derivative-free methods are mainly divided into direct search methods and model-based methods, both of which can be local or global. In addition, there are some random methods used for solving global optimization.

Representative software for direct search algorithms includes FMINSEARCH and NOMAD. Model-based methods include algorithms like Powell's NEWUOA series and Kelley's Implicit Filtering. Representative software includes BOBYQA, IMFIL, SID-PSM, SNOBFIT, and others.

2.2 Mixed-Integer Programming Software

Mixed-integer programming problems involve optimization problems that include both continuous and integer variables.

2.2.1 Mixed-Integer Linear Programming

General solvers for mixed-integer linear programming problems include CPLEX and GUROBI. GUROBI ranks first in mixed-integer linear programming in the famous global optimization solver comparison site "Decision Tree for Optimization Software," showing higher solving speed and accuracy. This solver can solve 227 out of 240 test cases within 4 hours. Additionally, domestic solvers such as COPT, developed by ShanShu Technology, and MindOpt, developed by Alibaba Damo Academy, are among the best domestic mixed-integer linear programming solvers. These two solvers rank highly in the same comparison website, solving 204 and 195 problems, respectively, indicating that domestically developed solvers have reached the top global level.

2.2.2 Mixed-Integer Quadratic Programming

Mixed-integer quadratic programming problems have important applications in portfolio optimization. Mixed-integer quadratic programming is NP-hard. Currently, solvers like BARON, SCIP, and GUROBI can solve non-convex mixed-integer quadratic programming problems. In the solver comparison website, for mixed-integer convex quadratic programming, international solvers include Gurobi, SHOT, OCTERACT, BARON, MOSEK, KNITRO, SCIP, MNTAUR, and BONMIN. The domestic solver COPT ranks first in performance, with Gurobi being 1.04 times slower than COPT, reaching international leading levels. However, global solvers for discrete non-convex quadratic programming problems have not yet been released by domestic solvers.

2.2.3 Mixed-Integer Second-Order Cone Programming

The objective function in mixed-integer second-order cone programming is linear, and the constraint function is a second-order cone, with integer variables included. This type of problem has broad applications, such as in option pricing and network design. The solving methods are divided into two categories: one developed from mixed-integer programming techniques, and the other specialized from general mixed-integer nonlinear programming methods. According to the solver comparison website, international solvers include Gurobi, Xpress, and SCIP. The domestic solver COPT performs only 2.13 times slower than Gurobi and outperforms Xpress, reaching international first-class standards.

2.2.4 Mixed-Integer Nonlinear Programming

Mixed-integer nonlinear programming involves both the challenges posed by integer variables and nonlinear functions. As a result, mixed-integer nonlinear solvers are complex and are typically built on the foundations of both mixed-integer programming solvers and nonlinear programming solvers.

SCICONIC is one of the earliest commercial mixed-integer nonlinear programming solvers.

International solvers for mixed-integer nonlinear programming include ANTIGONE, BARON, COUENNE, LINDO, OCTERACT, and SCIP, while no domestic solvers have been released yet.

2.3 Global Optimization Software

Currently, there are three major global optimization solvers worldwide: Baron, SCIP, and the Maple Global Optimization Toolbox.

2.3.1 Commercial Global Optimization Solver Baron

Baron is a software tool for solving nonlinear optimization problems. It is developed by Baron Optimization Inc., a U.S.-based company.

2.3.2 Open-Source Global Optimization Solver SCIP

SCIP is a powerful software suite for solving constrained integer programming problems. As an open-source solver, it provides a flexible set of functions and algorithms for solving various forms of constrained integer programming problems, including linear programming, mixed-integer linear programming, mixed-integer nonlinear programming, and mixed-integer nonlinear feasibility problems.

2.3.3 Maple Global Optimization Toolbox

The Maple Global Optimization Toolbox incorporates optimization algorithms from the Optimus software, developed by leading global optimization technology developer Noesis Solutions. Using the global optimization toolbox, users can build optimization models within the powerful Maple numerical and symbolic system and obtain robust, efficient results using advanced optimization algorithms.

2.4 Simulation Optimization Software

Simulation optimization is a powerful tool for optimizing large-scale complex systems. It refers to the optimization method that uses evaluation results from simulation models and optimization algorithms to find the optimal solution. Simulation optimization is widely applied in operations research problems. The optimizers in simulation optimization software include OptQuest, SimRunner, GoldSim Optimizer, Plant Simulation Optimizer, and the Global Optimization Toolbox.

Some open-source simulation optimization tools include SimOpt, Bayesian optimizers (e.g., DiceOptim, DACE, GPyOpt, Metrics Optimization Engine, Spearmint), Industrial Strength COMPASS, PyMOSO, Nevergrad, and PDFO. Some simulation optimization software, such as AnyLogic, is actively embracing artificial intelligence technologies and plans to release new simulation optimization products utilizing AI.

2.5 Optimization Software in Specific Industries

2.5.1 Optimization Methods and Software in Robotics and Autonomous Driving

Robotics and autonomous driving systems have a high demand for optimization computations because they require real-time decision-making and control in complex and dynamic environments. Typical optimization needs in robotics and autonomous driving include path planning, trajectory optimization, real-time decision-making, and control systems.

2.5.2 Optimization Methods and Software in Large-Scale Integrated Circuit Design

Large-scale integrated circuit design and simulation are some of the most challenging problems in electronic design automation (EDA). Mathematical optimization models, algorithms, and software play an essential role in various stages, from layout planning to simulation and verification.

In the EDA field, many specialized commercial software packages exist, such as Protel, Altium Designer, Cadence Allegro, Kicad, OrCAD, and EWB. On a global scale, Synopsys, Cadence, and Siemens dominate the EDA commercial software market.

2.5.3 Optimization Methods and Software in Aerospace Manufacturing

Manufacturing is the foundation of a country, and “promoting new industrialization” is a major transformation requirement for China’s manufacturing industry. In manufacturing, many application scenarios require optimization algorithms and software.

2.5.4 Optimization Methods and Software in the Economic and Financial Sector

Many practical problems in the economic and financial sectors are naturally optimization problems, such as enterprise production planning and optimal asset allocation for fund companies. The demand for optimization algorithms and software in the economic and financial sector includes several aspects: resource allocation optimization, enterprise production planning, asset allocation optimization, and index tracking and enhancement.

2.6 Optimization Modeling Software

Algebraic modeling languages (AML) are high-level computer programming languages used to describe and solve large-scale mathematical optimization problems. GAMS and AMPL are two of the most commonly used languages, widely applied in both academia and industry.

Other widely used modeling languages include Pyomo, PuLP, YALMIP, and SAS/OR, which are chosen and used for their compatibility with general-purpose computing software. With Python's widespread application and strong ecosystem, Pyomo and PuLP have gained significant attention and have become popular choices.

2.7 Major Gaps and Cause Analysis

China still has a considerable gap in the development of mathematical optimization software compared to the international leading level, with many areas lacking domestically owned optimization software. This is mainly due to China's late start in numerical computing and software development, requiring time and resources to transform theoretical results into practical software packages. Furthermore, the development of optimization software requires significant investment in manpower, financial resources, and time. In the past, China faced funding, specialized developers, and computing resource constraints, which slowed the development of software packages.

2.8 Policy Recommendations

Establish major or key projects related to optimization software by the National Natural Science Foundation or the Ministry of Science and Technology.

Strengthen the development of domestic open-source optimization software communities.

Integrate talent resources across fields and establish societies or associations related to optimization software.

Strengthen the recognition and support of optimization software achievements by social and academic institutions.

Actively introduce foreign resources with an open mindset, promoting domestic and international collaboration in optimizing software development.

Increase efforts to cultivate specialized optimization software talents to meet the growing technological demands.

Chapter 3: Numerical Solution Software for Differential Equations

Numerical solution software for differential equations serves as a powerful simulation tool, helping engineers efficiently solve and analyze complex scientific and engineering problems through computational means. It plays a significant role in scientific research and engineering applications.

There is a significant difference in the development of numerical solution software for differential equations between domestic and international markets. International software in this field reflects the industry’s constant pursuit of high integration, multifunctionality, and wide applicability, and highlights the critical role of large software companies in advancing technological innovation and market development. In contrast, domestic software for partial differential equation (PDE) numerical solutions lags behind in terms of technical level, integration, and market share. However, China has certain advantages in basic mathematical theory, talent reserves, research institutions, and policy support for the development of differential equation numerical solution software, and by seizing the development opportunities and growth trends, it is possible to achieve leapfrog development in this field.

3.1 Domestic and International History and Current Status of Differential Equation Numerical Solution Software

International software for the numerical solution of differential equations holds a dominant position in terms of technology, market share, and ecosystem development.

3.1.1 Numerical Solution Software for Partial Differential Equations

Software for solving partial differential equations (PDEs) has profound application value in engineering and physics.

The origins of computer-aided engineering (CAE) software in aerospace engineering can be considered as the precursor to PDE numerical solution software. CAE software has penetrated more application scenarios, including integrated circuits and other engineering fields, leading to the development of other PDE numerical solution software, such as TCAD (Technology Computer-Aided Design) for semiconductor process and device simulation.

The development of CAE software dates back to the 1960s and 1970s, initially driven by the needs of high-tech industries like aerospace and nuclear energy. As computer technology developed, the functionality and application scope of CAE software expanded. Over time, more CAE software companies emerged, such as Fluent, CD-adapco, COMSOL, and Altair. These companies' software primarily serves computational fluid dynamics, thermal analysis, electromagnetic field analysis, and other areas. By the early 21st century, CAE software had further expanded its applications. Some large software companies, such as Siemens, Dassault, and PTC, entered the CAE market, offering not only CAE functions but also CAD, CAM, and other features, thus creating a complete product lifecycle management solution.

With the rapid development of semiconductor technology and integrated circuits, TCAD software emerged as a powerful computational tool for simulating semiconductor processing and device operations. As semiconductor processes continued to shrink and designs became more complex, TCAD software evolved towards more integrated and automated solutions. TCAD software now not only simulates devices and optimizes processes but also integrates directly with electronic design automation (EDA) toolchains, enabling full automation from device simulation to circuit design.

The emergence of new materials and devices has surpassed the capabilities of traditional TCAD. When devices reach deep nanometer or even atomic scales, quantum effects become significant. At this point, traditional models fail to fully account for quantum effects, and obtaining reliable parameters through experimental means has become increasingly difficult and time-consuming. Furthermore, the continuous emergence of new electronic devices and materials has completely exceeded the scope of traditional TCAD methods. TCAD software has adapted by incorporating quantum effects.

International CAE software has evolved from serving a single domain to spanning multiple domains, and from small companies to large software enterprises. The development of international CAE software reflects the industry's ongoing pursuit of highly integrated, multifunctional, and widely applicable solutions, as well as the key role of large software companies in driving technological innovation and market development. These factors have made CAE software an indispensable tool in modern engineering design and analysis.

Domestic CAE software development has not started late; it began with specialized programs in the 1970s. In recent years, a batch of domestic CAE companies has grown in response to national development needs. China's CAE software has gone from non-existence to a certain scale, becoming a recognized industry with strength. However, domestic software faces challenges such as a lack of technological leadership, lower integration levels, lower autonomy, and a small market share compared to international competitors.

3.1.2 Numerical Solution Software for Ordinary Differential Equations (ODEs)

In the 1970s, ODE numerical solution software started to appear, with MATLAB and Mathematica offering toolboxes for solving ODE problems. Other international ODE software includes LSODE, ODEPACK, DASSL, and SCIPY.

The core tool for electronic design automation (EDA) software is SPICE. SPICE simulates electronic circuits by solving large-scale differential-algebraic equations. It is used in integrated circuit and board-level designs to check the integrity of circuit designs and predict circuit behavior. The first commercial version of SPICE was ISPICE, and other notable commercial versions include ISPICE, HSPICE, PSPICE, and SPECTRE.

Molecular dynamics (MD) simulation software originated in the 1960s to simulate molecular interactions and motions. With advances in computational power and theory, MD simulation software has become widely applied, especially in the fields of physics, chemistry, and biology. AMBER, a popular MD simulation software, was introduced by Peter Kollman at the University of California, San Francisco in the early 1980s. GROMACS, developed at the University of Groningen in the early 1990s, is another famous MD simulation software.

In recent years, with improvements in hardware performance and algorithm optimization, MD simulation software has undergone significant development. These advances enable the simulation of larger and more complex systems, providing more accurate predictions. AMBER, GROMACS, and other software are constantly releasing new versions with enhanced capabilities. Other popular MD software includes NAMD, LAMMPS, CHARMM, PL_POLY, and DESMOND.

The core tool of EDA software, SPICE, reflects the growing maturity of computational technology and the continuous demand for more advanced tools in engineering and scientific research.

Similarly, ODE numerical solution software is evolving towards higher efficiency, greater flexibility, more user-friendliness, and greater integration.

In China, ODE software development is mainly led by research institutes and university teams. For example, the BeiTai Tianyuan Numerical Computing General Software was independently developed by the Numerical Computing Laboratory of the Software Science Research Center at the Chongqing Big Data Institute, Peking University. BeiTai Tianyuan provides ODE solver functions such as ode23, ode45, ode78, ode89, ode113, as well as dde23, ddesd, and ddensd functions for solving delay differential equations.

MWORKS was developed by Tongyuan Soft Control Company to meet the needs of domestic engineers and scientists. ALPS is a SPICE simulation tool developed by HuaDa Jiutian Company. Concept Electronics is one of China's early domestic EDA vendors and is currently one of the leading domestic EDA companies.

The development of domestic ODE numerical solution software can be seen as a process of gradual maturity, from its initial stage to increasing autonomy. While there are still gaps in terms of tool completeness, autonomy, and market share, domestic ODE numerical solution software is showing promising development potential.

3.1.3 Development Platforms for Differential Equation Numerical Solution Software

As the complexity of real-world differential equation models increases, there is a growing need for development platforms specifically designed for numerical solutions of differential equations.

Important international development platforms include MFEM, FreeFEM, Deall.II, Firedrake, FEniCS, and OpenFOAM. Domestic development platforms include FEPG, AFEPack, JASMIN, and PHG.

These platforms provide a rich set of numerical algorithms and tools that support multi-physics and the solution of various types of differential equations.

3.2 Trends in the Development of Differential Equation Numerical Solution Software and Successful International Experiences

Differential equation numerical solution software integrates mathematics, computer science, and engineering applications, serving as a digital representation of mathematical models and numerical methods. Its core development focuses on the underlying aspects of software design.

3.2.1 Development Trends

CAX Integration Method

High-order numerical methods and high-quality mesh generation for PDEs in complex three-dimensional geometries have long been bottlenecks in fundamental industrial software. CAX integration technology is gradually becoming a focal point of interest. Developed countries such as those in Europe and the U.S. have started to lay the groundwork for CAX integration research, aiming to maintain a leading edge in industrial software development.

Software Cloudification

With the continuous development of cloud computing and big data technologies, the computational and data processing capabilities of application software are constantly improving. Cloud-based CAE software is gaining increasing attention.

Platformization of Applications

International companies expand their market share and improve competitiveness through acquisitions, creating product systems with a diversified range of products. The application fields of domestic CAE software are also expanding, covering industries such as aerospace, automotive, mechanical engineering, electronics, and construction.

3.2.2 International Successful Experiences

Emphasis on technological innovation, continuously releasing new algorithms, models, and tools to meet the needs of different fields.

The tendency to acquire specialized technology service providers through mergers and acquisitions.

Strengthening partnerships between enterprises and academic research institutions.

3.3 Issues and Development Opportunities for Domestic Numerical Solvers of Differential Equations

3.3.1 Challenges Faced Domestically

Incomplete Industry-Academia-Research Ecosystem

There is a lack of an integrated ecosystem for industry, academia, and research, which limits the efficiency and impact of innovation and development in this sector.

Lag in Industrialization and Insufficient Technological Accumulation

The industrialization of numerical solvers for differential equations has not kept pace with international standards, and there is insufficient technological accumulation in key areas.

Weak Intellectual Property Protection

Intellectual property protection for domestic software is inadequate, which leads to challenges in maintaining competitiveness and securing technological advantages.

3.3.2 Development Opportunities for Domestic Software

Due to "Bottleneck" Challenges from the West

As Western countries continue to "bottleneck" China’s access to key technologies, the demand for domestic substitution has become a crucial need. This creates an opportunity for China to develop its own software solutions.

Technological Turning Point as a New Opportunity

By seizing the technological turning point, China has the potential to leapfrog in industrial software development, similar to how the country’s automotive industry transitioned from being behind in gasoline engines to developing electric vehicles in sync with global trends.

Formation of Open Cooperation Ecosystem

With the ongoing trends of globalization and informatization, the open collaboration ecosystem in China’s industrial software sector is gradually taking shape. As cooperation and exchanges between enterprises become more frequent, they collectively drive the development of the software industry.

3.4 Development Suggestions and New Discipline Growth Points

3.4.1 Growth Points in Disciplines

Computational Modeling

Numerical solvers for differential equations represent a digital expression of mathematical methods and theories. The vitality of these solvers lies in the mathematical models they employ.

There is growing interest in integrating machine learning to generate causal understandings of the world. New modeling methods that combine domain knowledge, symmetry, and invariance into AI systems are gaining attention. This integration can be achieved through data augmentation, deep learning system design, potential energy models, architectural features, loss functions, and performance quantification techniques.

New Computational Methods

Most current numerical methods for differential equations are based on boundary-conforming grids, but generating high-quality grids in complex geometric regions remains a challenge, limiting further improvements in simulation accuracy and efficiency.

The CAX integration method, which considers both CAD modeling and CAE simulation, aims to achieve automatic mesh generation that satisfies any desired precision. This requires the development of new computational methods and mathematical theories.

The remarkable advantages of deep neural networks in image processing have proven their strong ability to solve high-dimensional problems. Applying artificial intelligence to solve partial differential equations, especially for high-dimensional problems, is a promising research direction.

Heterogeneous parallel computing, a current trend in computational hardware development, is particularly relevant as domestic computer architectures diversify. Large-scale engineering problem-solving technologies and software computational capacities are severely inadequate, and the scalability of parallel algorithms is poor, making it difficult to integrate them with large domestic computing systems. Therefore, researching heterogeneous parallel algorithms compatible with domestic heterogeneous systems is an important direction for future research.

Software Implementation Technologies

Cloud-native technologies, as an advanced approach for software development and deployment, aim to solve the challenges of integrating traditional application software with cloud computing. In the context of the digital intelligence era, cloud-native technology has become an effective means for software companies to achieve rapid innovation and respond to market competition due to its lightweight and highly scalable characteristics.

With the development of cloud-native technologies, low-code development platforms have emerged, providing a simpler and more efficient development model. These platforms aim to reduce the complexity of development and accelerate project delivery through visual development and drag-and-drop operations.

3.4.2 Development Recommendations

Recommendation 1: Strengthen Talent Development and Provide Long-term Support

It is crucial to build research teams that align with China’s national conditions and provide sustained support over the long term.

Recommendation 2: Emphasize the Development of the Industry-Academia-Research Ecosystem

It is important to integrate basic research with engineering applications in the field of industrial software. Strengthening collaboration between industry, academia, and research institutions will help drive innovation and ensure that research findings are effectively translated into practical applications.

Chapter 4: Computational Geometry Software

4.1 Development History of Computational Geometry Related Software

4.1.1 CAD/CAM Software

In the 1960s, the first CAD software, Sketchpad, was developed at the Massachusetts Institute of Technology (MIT), marking a revolutionary advancement in the way designers interacted with computers by drawing on computer screens.

Due to the high cost of early computers and the unique mechanical engineering requirements of the aviation and automotive industries, large aerospace and automotive companies became the first commercial users of CAD software. Many companies also developed their own specialized CAD software, including car manufacturers such as Ford (PDGS), General Motors (CADANCE), Mercedes-Benz (SYRCO), Nissan (CAD-I), and Toyota (TINCA and CADETT), as well as aerospace manufacturers such as Lockheed (CADAM), McDonnell Douglas (CADD), and Northrop (NCAD). These companies established large internal software development teams dedicated to creating proprietary CAD programs. In 1975, Dassault Aviation in France developed CATIA, a 3D CAD software program, which has since become one of the most successful commercial CAD software solutions.

In the 1970s, modeling systems based on boundary representation (B-rep) and constructive solid geometry (CSG) were released by Cambridge University and Stanford University. Subsequently, two of the world’s most well-known commercial geometry engines, Parasolid and ACIS, were developed at the Cambridge University CAD laboratory. Geometry engines are critical foundational software relied upon by CAD, CAE, and CAM software. Parasolid and ACIS have become the most widely used geometry engines in mainstream CAD software worldwide. Parasolid was later integrated into the U.S. UG software suite, becoming a common geometric platform for solid and surface modeling. ACIS was later acquired by Dassault.

In the 1980s, Boeing, General Electric, and the National Institute of Standards and

Technology (NIST) released the IGES (Initial Graphics Exchange Specification) format, which enabled the transfer of complex 3D curves and surfaces between different CAD software systems. IGES remains one of the most widely used data conversion formats in CAD software today.

As CAM is closely related to CAD, CAM software is often integrated as a feature within CAD software. Since the 1990s, many powerful CAD/CAM software programs have emerged globally. Widely used software includes Unigraphics NX, CATIA, MASTERCAM, CIMATRON, SolidWorks, PowerMill, among others.

China's early CAD system development progressed almost simultaneously with international efforts. However, compared to well-known foreign CAD software such as CATIA and UG, domestic CAD software still lags significantly in terms of functionality and technical maturity. The lack of self-controlled geometric kernels and certain core geometric modeling and computing technologies remains a major reason for the lag in China's CAD software. Additionally, the industry ecosystem contributes to this disparity.

4.1.2 Mesh Generation Software

Since the 1980s, a series of mature mesh generation software solutions have been developed, and these have been integrated into industrial design and simulation workflows. Most mesh generation software is closely linked to the development of industries like aerospace and defense. Starting in 1996, the European Union, Israel, and other regions set up multiple project funds to develop CGAL (Computational Geometry Algorithms Library), including mesh generation. In 2000, ANSYS acquired ICEM CFD mesh software and integrated it into its workflow, now part of one of the leading industrial software companies.

Currently, mesh generation software is divided into commercial and free/open-source software. Commercial mesh software is often integrated into CAE software, such as the ANSYS software suite, which includes ICEM CFD and Meshing. Other standalone mesh software includes Pointwise. Free and open-source software is often supported by national or laboratory funding, such as MeshLab, developed by the Visual Computing Lab of the Institute of Information Science and Technology (ISTI-CNR) in Italy; Gmsh, supported by funding and technical support from the EU, the U.S., and other regions; CGAL, funded by multiple European and Israeli project funds; and Open CAX (including BRL-CAD, FreeCAD), which originated from U.S. military projects and now serves as a non-profit organization but still supports the U.S. military.

The development of mesh generation software in China started relatively late. Current domestic mesh generation software systems include: HEPD/Pre developed by Zhejiang University, NNW-GridStar developed by the China Aerodynamics Research and Development Center, AUTOMESH-2D developed by Shandong University, JIFEX software pre-processing subsystem Auto FEM developed by Dalian University of Technology, and SuperMesh developed by the Institute of Engineering Physics. Overall, domestic mesh generation software still lags behind international solutions in terms of technology and market share. Most practitioners still use foreign open-source or commercial software.

Despite this, there are still many defects and challenges in mesh generation software, both domestically and internationally.

4.1.3 Isogeometric Analysis (IGA) Software

CAE simulation software has seen rapid development since the 1960s, with a number of excellent simulation programs emerging, such as ANSYS, LS-DYNA, etc. These software programs typically grew stronger through mergers and acquisitions and integrated CAD, mesh generation, CAE, and CAM systems.

Isogeometric analysis (IGA) is an extension of the finite element method (FEM). However, since IGA was proposed later, its commercialization is still in its early stages. Current notable IGA companies include Boeing, LS-DYNA, and CoreForm. Boeing uses IGA software for several advantages: first, the IGA method can precisely calculate shape sensitivity, with accuracy exceeding 10 digits of precision; second, simulations performed in the parameter space of the geometric model can help overcome many computational barriers. Coreform IGA is a solver that can run directly on lattice structures, CAD models, and STL files, making it much more efficient than traditional FEM methods that require meshing of geometric shapes. Coreform Cubit can convert CAD models into high-quality hexahedral meshes, greatly reducing the time spent on FEA and CFD model preparation while maximizing mesh quality and thus improving computational efficiency.

IGA is one of the promising directions for achieving CAD/CAE integration. Currently, IGA research in China is still limited to universities, and no systematic software development teams or companies have emerged. In 2022, the Ministry of Science and Technology launched a key development plan for theoretical research, algorithms, and software development based on IGA, but overall investment remains low. Overall, China is significantly behind in terms of basic research, talent, and software development compared to international standards. To bridge the gap in industrial software development and reduce reliance on foreign products, greater attention must be paid to IGA software development.

4.2 Development Trends and Challenges

4.2.1 Development Trends

Policy Trends

Currently, Autodesk, PTC, Dassault, and Siemens are the four companies that dominate over 90% of the CAD/CAM market. The U.S. government places high importance on the development of industrial software, providing ample support to its industrial software market. Large industrial software companies in European countries also play a dominant role in the development of their respective industries. For example, Dassault controls more than 20% of the market share in France, significantly influencing the entire industrial software industry.

In China, CAD/CAM software development began in sync with international efforts from the "Seventh Five-Year Plan" to the "Fifteenth Five-Year Plan," with projects like the "CAD Key Project" by the Ministry of Mechanical and Electrical Industry and "863/CIMS, Manufacturing Industry Informatization" by the Ministry of Science and Technology. However, domestic three-dimensional CAD software often lags behind foreign counterparts due to high technical barriers, which require significant long-term investment in basic scientific talent. Meanwhile, as China’s manufacturing sector enters a period where high-end software design capabilities are increasingly required, foreign software companies have captured the domestic market. Since the "Eleventh Five-Year Plan," China's investment in 3D CAD software research has decreased significantly, failing to meet the needs of catching up with international products.

Although several industrial software companies have emerged in China, only a few large software enterprises are recognized, and many small and medium-sized companies face financial difficulties. The primary reason is that policy support is not sufficiently focused, and the lack of targeted policies has led to weak industry-academic collaboration, reducing the rate of technological transformation.

Technology Trends and Discipline Growth Points

Current CAD modeling systems still have many technical shortcomings, such as challenges with operations like intersection, smoothing, and filleting, and the lack of watertight models under commonly used B-rep-based CAD representations. Moreover, the conversion from B-rep to solid representations is difficult, and there is a lack of systematic error control methods. Future trends in CAD technology will focus on cleaner, more interactive modeling methods, using advanced mathematical theories such as new splines to resolve issues like watertight models and solid representations, while also improving local refinement capabilities. Additionally, there will be a push to incorporate physical properties, such as density and temperature, into geometric representations for later simulation needs.

CAD and CAE have traditionally developed in separate fields, with their respective software developed independently. However, since CAD and CAE represent the front and back ends of the design and simulation process, and their geometric data representations differ, data conversion efficiency between CAD and CAE has been a significant issue. As future aerospace equipment design requires earlier integration of simulation analysis to fully simulate and optimize performance, CAD/CAE software integration will become a key factor for accelerating design iterations.

Cloud Computing Trends

Cloud computing is becoming a major trend in CAD/CAE/CAM software. With the rapid development of the internet industry, new technologies are emerging, and integrating these with the industrial software sector is a natural direction. Cloud computing services are already mature enough to assist in CAD design and CAE simulations, allowing users to upload calculation files and choose solvers through cloud platforms, supporting product design.

4.2.2 Challenges

Talent Development Challenges

The CAX software industry is a knowledge- and technology-intensive sector, requiring developers to possess expertise in geometry, algebra, topology, mathematical modeling, and numerical computing. Talent development takes a long time and requires a conducive environment. There is a shortage of skilled interdisciplinary professionals to meet the increasing demand, which hinders the industry’s rapid development.

There is a lack of specialized programs in computational geometry at universities, and established interdisciplinary training systems around CAX software are not yet in place. Furthermore, there is a lack of vocational education and systematic training programs for CAX software development in China, resulting in insufficient social reserves of professionals.

Industrialization Challenges

Currently, CAX industrial software manufacturers in China face insufficient competitiveness and remain largely dominated by foreign companies. There are significant gaps in the integration of key technologies and industrial demand between Chinese and foreign products. Moreover, China’s industrial software value chain is incomplete, with a lack of emerging industry segments.

4.3 Policy Recommendations

4.3.1 Fund Policy Support

Computational geometry is the core foundation of CAX industrial software, and key technologies for CAD, CAM, CNC, mesh processing, and isogeometric analysis (IGA) all stem from computational geometry. However, under the current classification of the

National Natural Science Foundation of China (NSFC), computational geometry is a subfield under computational mathematics, and it receives far less funding compared to other fields such as numerical solutions of partial differential equations. It is recommended that the NSFC provide special funding and talent support for computational geometry, offering stable support for this important but relatively weak discipline.

4.3.2 Government Decision-Making and Coordination

China has already made multi-faceted policy efforts to support key industrial software domains like CAD. However, projects from different ministries often overlap in scope, which leads to unnecessary duplication of human resources and academic efforts. It is recommended to establish an inter-ministerial coordination meeting under a higher-level body, such as the Central Science and Technology Commission, to unify decision-making, planning, and technical exchanges.

4.3.3 Interdisciplinary Organization Integration

CAD/CAE/CAM integration is the international frontier in the development of industrial software. In the past, these three fields were independently developed, but in the digital design and manufacturing process, they are interdependent and essential. It is recommended that the NSFC and the Ministry of Science and Technology integrate these fields at the organizational level and strengthen collaboration.

4.3.4 Talent Training and Industry-Academia Collaboration

Currently, there is a severe shortage of talent in CAX research and development. To cultivate more CAX R&D talent, it is crucial to offer foundational courses in computational geometry at the graduate level and provide specialized training for R&D personnel.

Chapter 5: Symbolic Computation Software

5.1 Overview of Symbolic Computation and Software

Symbolic computation (also known as computer algebra) uses symbols to represent and operate on mathematical objects for precise mathematical calculations.

Symbolic computation originated as a branch of artificial intelligence. The first generation of artificial intelligence was symbolic AI, which focused on logic and reasoning processes, including automated reasoning, expert systems, and knowledge engineering. To implement logical reasoning on computers, symbolic computation software was necessary.

A landmark in symbolic computation software development occurred in 1961 when J. Slagle, using the programming language Lisp, created the first automatic symbolic integration program, SAINT. The first general-purpose symbolic computation software, Macsyma, was developed at MIT with the support of DARPA by Martin, Moses, and others, under the guidance of Minsky, one of the founders of artificial intelligence. Early symbolic computation systems were mainly developed in the United States, particularly at MIT, Bell Labs, and IBM.

Symbolic computation is also closely related to the work of Chinese mathematician Wu Wenjun, who advocated for the mechanization of mathematics. Two monumental results in logical reasoning are Gödel's proof that any consistent axiom system containing Peano arithmetic is undecidable, and Cook's proof that automatic theorem proving in propositional logic is NP-complete.

5.2 General-Purpose Symbolic Computation Systems

Currently, the most widely used commercial general-purpose symbolic computation software includes Maple and Mathematica. These systems share a common focus on symbolic computation while also incorporating modules for numerical computation, data analysis, system simulation, and other applications. For example, Maple initially was purely symbolic, designed for solving various mathematical problems symbolically, catering to scientific research and teaching. Over time, it expanded to include functions for numerical computation, data analysis, and graphical plotting. The recently released MapleSim is used for modeling and simulating complex systems.

5.3 Specialized Symbolic Computation Systems

In addition to general-purpose symbolic computation software, several specialized symbolic computation systems exist. Notable examples include:

GAP (Group Theory Computation),

CoCoA (Commutative Algebra Computation),

PARI (Number Theory Computation), and

NTL (Number Theory Library).

The development of specialized symbolic computation software is driven by three main goals:

Specialized software can implement more complex and advanced algorithms specific to their field, offering stronger computational capabilities compared to general-purpose software.

To address particular needs in fields like group theory, commutative algebra, and number theory.

To support the development of more powerful mathematical tools for research and education.

Notable international specialized symbolic computation systems include GAP, CoCoA, and Singular.

In the 1980s, Wu Wenjun developed "China-Prover," which efficiently automated geometric theorem proving. Over the next two decades, China developed multiple successful software systems for mathematical mechanization, focusing on equation solving, machine proofs, geometric drawing, Ore polynomials, and mixed computation. However, most of these systems were based on foreign software, such as Maple, and lacked a truly independent, general-purpose software platform based on Chinese theories. The Mathematical Mechanization Platform (MMP), supported by China’s “973” project, was developed to fill this gap.

MMP is a mathematical software based on C++ and developed for the Microsoft Windows interface. Its core functionality centers around the basic mathematical mechanization theories developed by Wu Wenjun, such as polynomial systems, algebraic ordinary differential equations, algebraic partial differential systems, and machine proofs for geometric theorems. The software also integrates a basic symbolic computation system and provides various methods for equation solving and machine proofs. MMP is composed of four parts: support systems, symbolic computation systems, core modules, and application modules.

5.4 Automated Reasoning Software Systems

Automated reasoning and formal methods are techniques based on strict mathematical theories in computer science. Theorem-proving techniques use formal methods to represent systems and system properties as mathematical theorems and complete verification through axioms and reasoning rules.

Theorem proving is categorized into two types: automated theorem proving and interactive theorem proving.

Historically, the first developed theorem proving systems were automated. For example, the "Logic Theorist" program by Newell and Simon proved a large subset of the propositional logic portion of Whitehead and Russell's Principia Mathematica. Comprehensive automated theorem proving systems include Otter (a first-order logic theorem prover), NQTHM (Boyer-Moore Theorem Prover), and Prolog. Currently, SAT solvers, such as MiniSat, are the most widely used automated theorem proving tools. Although SAT solving is NP-hard, it is a fundamental problem in automated reasoning because nearly all automated reasoning can ultimately be reduced to SAT solving. As a result, many effective algorithms and software for NP problem-solving have been developed.

Due to the NP-hard nature of automated theorem proving, interactive theorem proving systems are more widely used. These systems combine efficient but incomplete automated reasoning tools with human intervention. When automated reasoning cannot proceed further, users can guide the proof by adding lemmas or insights, allowing the system to generate a final proof. Though the proof process is not fully automated, the resulting proof is rigorously verified. Interactive theorem proving has been used to solve several mathematical problems and verify the accuracy of key software and chip designs.

5.5 Issues and Countermeasures

Currently, the leading general-purpose symbolic computation software systems are mostly developed and controlled by Western countries. While there are some alternative products in certain fields, their technological maturity and functionality still lag far behind foreign products. China faces significant gaps in this area, and bridging these gaps is a top priority.

The development of symbolic computation software depends not only on breakthroughs in basic research in symbolic computation but also on support from computer science, especially software engineering, to enhance software development efficiency, management, and tools. Symbolic computation software, particularly application-related modules, is a cross-disciplinary issue that requires collaboration between different fields.

A new trend is the application of deep learning, especially large language models, in symbolic computation and automated theorem proving, which has already achieved some progress. However, this approach requires substantial computational resources, and thus research in this field requires multidisciplinary collaboration and funding support.

Developing general-purpose symbolic computation software, particularly from an academic research laboratory to market commercialization, demands long-term commitment from development teams, requiring stable funding for initial development and sustained support for ecosystem building, software updates, and maintenance. Moving from experimental platforms to commercial applications also requires substantial policy support.

Chapter 6: Statistical Computing Software

6.1 Status and Future of Statistical Software

Statistical computing software plays a crucial role in data analysis and statistical modeling, providing powerful tools for researchers, statisticians, and data scientists to handle, analyze, and interpret data.

In recent years, driven by the growing demand for data analysis and machine learning, there has been significant progress in the development of statistical computing software. Some notable advancements include:

Integrated Development Environments (IDE): IDEs like RStudio, Jupyter Notebook, and Spyder provide comprehensive statistical computing environments, including code editing, interactive execution, and data visualization.

Open-Source Software: Open-source statistical computing software, such as R, Python, and Julia, has been widely adopted due to its flexibility, community support, and rich libraries. These tools have democratized statistical analysis by making advanced technologies available to a broader user base. R's extensive package ecosystem and Python's flexibility, especially its integration with machine learning frameworks like TensorFlow and PyTorch, have made them popular choices for data scientists.

Cloud-Based Solutions: Cloud computing has made it possible to develop cloud-based statistical computing platforms. Services like Google Colaboratory, Microsoft Azure Notebooks, and Amazon SageMaker allow users to perform data analysis and modeling in the cloud, utilizing scalable computing resources.

The future of statistical computing software holds many exciting prospects:

Integration with Big Data and AI: Statistical software will continue to evolve to handle large-scale data and integrate seamlessly with artificial intelligence (AI) and machine learning technologies. Efficient algorithms and distributed computing frameworks will be developed to process and analyze massive datasets.

Interactive and Visual Data Analysis: Emphasis will be placed on enhancing interactive and visual data analysis capabilities. Tools enabling exploratory data analysis, real-time visualization, and interactive dashboards will be developed, helping users gain insights quickly.

Automation and Automated Machine Learning (AutoML): Statistical computing software will increasingly automate complex tasks, such as feature engineering, model selection, and hyperparameter optimization. AutoML technology will be integrated into statistical software, enabling users to focus on higher-level analysis.

Reproducibility and Collaboration: There will be a stronger emphasis on reproducibility and collaboration in statistical computing. Tools that promote reproducible research, version control, and collaborative workflows will be developed to enhance the transparency and repeatability of data analysis.

Privacy and Ethics: With the growing importance of data privacy and ethics, statistical computing software will evolve to incorporate privacy-preserving technologies and ethical guidelines. Tools will be developed to conduct privacy-conscious analysis and address bias in data analysis.

6.2 Research Foundation, Features, and Strengths in China

China has many internet companies with strong data analysis capabilities. Numerous companies offer data management and analysis solutions, mainly for large enterprises, although the underlying software may not always be domestically developed. When these companies face restrictions on using foreign software, their products may also be limited. For example, Xinghuan Technology, a publicly listed company, provides big data solutions and software products, mainly focused on machine learning modeling for statistical analysis. FanRuan Software, with products like FineReport (reporting software) and FineBI (business intelligence software), is also highly regarded.

In recent years, China has made significant progress in the field of statistical computing software by actively developing and promoting various tools and platforms for data analysis, modeling, and decision-making. While there are several domestic small-scale statistical analysis software tools, their influence in the academic community is not yet widespread. For example, Markwei Analysis System, developed by Shanghai Tianlu

Information Technology Co., is a commercial software with complete data management, statistical modeling, and machine learning features.

Some notable characteristics of the current state of statistical computing software in China include:

Open-Source Tools: China actively participates in the development and adoption of open-source statistical computing tools. R, Python, and their related libraries and frameworks are widely used in both academia and industry for statistical analysis, data visualization, and machine learning.

Rapid Development of Data Science and AI: The rapid development of data science and artificial intelligence in China has led to an increasing demand for statistical computing software. Many organizations and research institutions are actively developing tools tailored to meet the specific needs of the Chinese market and research community.

Integration with Big Data Analysis: China's progress in big data analysis has led to the integration of statistical computing software with big data platforms. Tools like Apache Hadoop, Spark, and TensorFlow are widely used for large-scale data processing, distributed computing, and machine learning tasks.

Domain-Specific Software: China has also developed domain-specific statistical computing software tailored to the needs of particular industries. For example, in finance, healthcare, and e-commerce, there are specialized tools for statistical analysis designed to address the unique challenges of these sectors.

Government Initiatives and Support: The Chinese government recognizes the importance of statistical computing software in driving technological innovation and has launched initiatives to support its development. These include funding research projects, promoting collaboration between academia and industry, and advocating for the use of statistical computing across various fields.

Education and Training: Universities and research institutions offer comprehensive courses and training programs in statistical computing and data analysis, aiming to cultivate skilled professionals who can effectively use statistical software tools.

Given its large population, China also has several advantages in the field of statistical computing software:

Large Talent Pool: China has a vast talent pool in mathematics, statistics, and computer science, with many professionals skilled in statistical computing and data analysis. This talent pool is a key contributor to the development and advancement of statistical computing software.

Market Potential: With a population of 1.4 billion, China represents a massive market for statistical computing software. Sectors such as finance, healthcare, e-commerce, and manufacturing have a growing demand for data analysis tools and platforms. This market potential attracts investment and drives the development of innovative statistical computing solutions.

Government Support: The Chinese government recognizes the significance of statistical computing software in fostering technological innovation and economic growth. It has implemented initiatives to support research and development in this field, including funding programs, research grants, and policy incentives. Government support has accelerated the development of statistical computing software in China.

Integration with Big Data: China's big data analysis capabilities have significantly developed, benefiting from the availability of massive datasets from various sources. China's statistical computing software is well-positioned to integrate with big data platforms, enabling efficient processing and analysis of large-scale datasets. This integration enhances the ability of statistical computing software to handle complex data analysis tasks.

Industry Collaboration: China has a thriving tech industry and a culture of collaboration between academia and industry. This collaboration fosters the exchange of knowledge, ideas, and resources, driving the development of practical, industry-specific statistical computing software solutions. Close collaboration between researchers, software developers, and industry professionals strengthens the competitiveness and innovation of this field.

Adaptability and Innovation: Chinese software developers and researchers in the field of statistical computing exhibit high adaptability and innovation. They actively participate in open-source projects, develop new algorithms and methods, and adapt existing tools to meet the specific needs of the Chinese market. This adaptability and innovation drive the development of statistical computing software in China.

6.3 New Growth Areas in the Discipline

The close integration of statistical computing with big data analysis and machine learning presents an exceptional development opportunity for the field of statistical computing software. Among these, probabilistic programming languages hold the potential to shape the future of statistical computing software and are an exciting area of growth in the discipline. The following advantages of probabilistic programming align with the evolving demands of data analysis and modeling, making it a crucial element in the future of statistical computing software:

Model Flexibility: Probabilistic programming allows users to express complex statistical models in a more natural and flexible way. It enables the introduction of uncertainty, dependencies, and hierarchical structures in the model, which is particularly important as data becomes increasingly diverse and complex.

Inference and Uncertainty Quantification: Probabilistic programming frameworks provide built-in algorithms for Bayesian inference and uncertainty quantification. These frameworks automate the estimation of model parameters, conduct posterior inference, and propagate uncertainty throughout the analysis process. This simplifies the modeling process and provides richer insights into the uncertainty of results.

Integration with Machine Learning: Probabilistic programming bridges the gap between statistical modeling and machine learning by allowing the combination of probabilistic models with machine learning techniques. It integrates structured statistical modeling with powerful deep learning and neural network architectures. This integration is essential when solving complex real-world problems that require both probabilistic reasoning and pattern learning from data.

Reproducibility and Transparency: Probabilistic programming enhances the reproducibility and transparency of statistical analysis. By expressing models and inference processes explicitly as code, sharing, reproducing, and verifying results become easier. This promotes collaboration, knowledge sharing, and supports rigorous scientific research.

Automatic Differentiation: Many probabilistic programming frameworks leverage automatic differentiation techniques to enable efficient gradient-based optimization. This allows for the use of powerful optimization algorithms for model calibration and parameter estimation, leading to faster and more accurate analysis.

Domain-Specific Libraries: Probabilistic programming frameworks often come with domain-specific libraries that offer predefined models, distributions, and inference algorithms. These libraries help users quickly get started and facilitate building complex models for specific fields such as finance, healthcare, or social sciences.

Integration with Modern AI Technologies: Many artificial intelligence techniques, such as deep generative models, are inherently probabilistic and can be implemented using probabilistic programming frameworks. This integration further expands the potential of statistical computing software.

6.4 Thoughts on Developing Domestic Statistical Computing Software

Although there are some domestic statistical computing software solutions, they are rarely used in research and enterprise settings. The main reasons for this are twofold: on one hand, well-established foreign software is already widely used, and researchers often begin their studies with these tools, making it harder to transition to new domestic software; on the other hand, researchers in academic settings face pressure to publish in international journals, where results typically need to be verified using well-known software. Similarly, in industry, companies are more inclined to use trusted free software and more reliable commercial products.

The small user base of domestic software limits the amount of development funding and user feedback, which impedes software quality improvement. This is similar to the situation with domestic operating systems.

With increasing international competition, the Western bloc, led by the United States, may use various methods to limit the use of Chinese software. Therefore, some key functions in statistical computing software should be pre-designed with frameworks and algorithm reserves to ensure rapid development when needed. The most immediate focus should be on foundational computational functions such as mathematical programming languages, matrix computation, numerical integration, optimization, and stochastic simulation. These foundational features require high-level algorithm experts and skilled programmers to develop software that is reliable, accurate, efficient, and user-friendly. After these foundational features are developed, additional functionalities can be incorporated by organizations specializing in various statistical fields.