

自主可控的科学计算与模拟仿真软件研究

◆ 李 若¹ 向 导² 王 程²

1. 北京大学数学科学学院, 北京 100871;

2. 北太振寰(重庆)科技有限公司, 重庆 401329

摘 要 科学计算与模拟仿真软件作为支撑航空航天、智能网联汽车和新型电力系统等重要战略产业的“数字底座”，是最重要的基础软件之一，而基础软件的自主可控已成为关乎国家科技安全的核心命题。系统梳理近半个世纪全球科学计算与模拟仿真软件的技术演进路径，揭示我国在该领域遭遇的“解释器—算法库—生态体系”三重困境。以具有完全自主知识产权的北太天元科学计算与模拟仿真软件为研究载体，介绍国产自主科学计算与模拟仿真软件的技术架构，并以两个应用案例说明北太天元的应用场景。旨在为国产软件从“可用”向“好用”转变提供可复制的技术范式与产业生态建设方案，探索科研成果落地与科技自立自强双向赋能机制。

关键词： 国产自主；科学计算；北太天元；科技创新

中图分类号： TP316 **文献标识码：** A

文章编号： 1009-2412(2025)03-0049-10

DOI： 10.3969/j.issn.1009-2412.2025.03.007

0 引言

科学计算利用计算机技术来解决科学研究和工程领域中的数学问题，通过数值模拟、数据分析等方法，帮助人们更好地理解和预测复杂的自然现象和工程问题，是现代科学研究中不可或缺的重要手段。科学计算的核心在于将复杂的科学问题转化为数学模型，并通过计算机进行高效的数值求解。这

一过程不仅需要强大的计算能力，还需要数学、物理学和计算机科学等多个学科的知识融合。科学计算的发展历程与计算机技术的进步紧密相连。从早期的机械式计算工具到现代的超级计算机，科学计算的工具和方法经历了巨大的变革。科学计算可预测、发现和再现客观世界的运动规律与演化特性，通过建立数学物理模型，研究计算方法，开发应用程序等进行模拟计算并分析计算结果^[1]。

随着科学计算和模拟仿真的不断发展，相关软件工具也应运而生。这些软件为复杂系统的建模、仿真和分析提供了强大的支持。美国、欧盟和日本等均将科学计算与模拟仿真软件研究上升为国家战略。2020年MATLAB软件禁用事件后，在科研教育层面，部分顶尖教育科研机构的研究项目因无法使用专业工具箱而遭遇研发瓶颈，同时相关科研人才的能力培养受到阻碍；在工业领域，汽车电子（尤其是符合ISO 26262安全标准的电控单元开发）、能源电力等核心产业失去仿真验证工具链，导致研发周期延长。更深远的影响在于切断了我国工业体系与全球技术生态的连接——无法兼容国际主流硬件设备的代码自动生成接口（如TI C2000、dSPACE），导致国产芯片与工业软硬件协同设计脱节，形成从底层算法、工程工具到产业标准的断链，严重影响了我国产业链安全与技术自主演进能力。因此，我国在该领域进行大力投入，以自主可控为目标的国产科学计算和模拟仿真软件开始备受关注，进入蓬勃发展阶段^[2]。

1 国内外研究现状

在科学计算与模拟仿真领域，全球已形成协同演进的技术生态体系。如表1所示，国外相关软件主要为MATLAB/Simulink、Scilab/Xcos、Maple/MapleSim和OpenModelica等，国内相关软件主要为北太天元、MWORKS和GCKontrol等。

收稿日期：2025-04-10 修回日期：2025-06-03

项目资助：国家重点研发计划项目（2023YFA1011500）。

通信作者：李若，教授，rli@math.pku.edu.cn。

表 1 国内外主流科学计算与模拟仿真软件

软件名称	研发公司	国家
MATLAB/Simulink	MathWorks	美国
Scilab/Xcos	Dassault Systemes	法国
Maple/MapleSim	Maplesoft	加拿大
OpenModelica	Open Source Modelica Consortium	瑞士
北太天元	北太振寰	中国
MWORKS	同元软控	中国
GCKontrol	世冠科技	中国

1.1 MATLAB/Simulink

MATLAB/Simulink 是一款强大的科学计算软件和动态系统仿真平台，广泛应用于科学研究、工程设计和数据分析等领域。MATLAB 的起源可以追溯到 20 世纪 70 年代，最初由 Cleve Moler 用 Fortran 语言编写，以方便学生和研究人员进行矩阵运算。1984 年，Jack Little 等人基于 C 语言将其重新实现并成立了 MathWorks 公司，并在 1989 年推出了可视化的仿真环境，命名为 Simulink^[3]。MATLAB 可提供强大的矩阵运算、数值分析和数据可视化功能，支持多种数据类型和编程范式。Simulink 则专注于动态系统的建模与仿真，用户可以通过拖拽模块创建模型，支持多领域建模，如控制理论、信号处理和通信系统。MATLAB/Simulink 支持代码生成和硬件部署，能够将模型直接转化为嵌入式代码^[4]。此外，Simulink 与 MATLAB 紧密集成，用户可以利用 MATLAB 的分析工具对仿真结果进行深入处理。MATLAB/Simulink 功能不断扩展，从最初的矩阵计算发展到如今涵盖大数据分析、人工智能和自动驾驶等多个领域的综合性平台，在教育、科研和工业界的应用也日益广泛，成为工程师和科学家不可或缺的工具。

1.2 Scilab/Xcos

Scilab/Xcos 是一款开源科学计算软件及动态系统建模与仿真工具，广泛应用于教育、科研和工业领域。Scilab 的起源可以追溯到 1982 年，最初由法国国家信息与自动化研究所（Institut national de recherche en informatique et en automatique, INRIA）的 Francois Delebecque 开发，其灵感来源于早期的 MATLAB；1990 年，INRIA 和法国国立桥路学校决定开发 Scilab 的开源版本，并于 1994 年发布了第一个版本 Xcos。Xcos 最初作为 Scicos 工具箱

的一部分，由 Ramine Nikoukhah 开发，首次发布时包含了图形化编辑器；2008 年，Scicos 采用通用性公开许可证（GPL），进一步促进了其开源社区的发展^[5]。Xcos 还提供图形化编辑器，用于设计混合动态系统模型，支持离散和连续时间域的建模与仿真^[6]。Xcos 不仅能够进行复杂的系统仿真，还能将仿真结果进行实时图形化展示，并支持在 Scilab 中进行后续处理。2012 年，法国达索系统（Dassault Systèmes）收购 Scilab 项目，之后 Scilab 的开发逐渐集中在与达索系统其他产品的集成、企业用户的定制化需求上，可能会逐渐失去对学术界和纯粹开源用户的吸引力。

1.3 Maple/MapleSim

Maple/MapleSim 是一款数学计算与多领域系统建模仿真软件，广泛应用于教育、科研和工程设计等领域。Maple 最初由加拿大滑铁卢大学的研究人员开发，20 世纪 80 年代初推出第一个版本；1988 年，Maplesoft 公司成立后继续对 Maple 进行开发和商业化^[7]。2008 年，MapleSim 作为 Maple 的补充发布，专注于动态系统的建模与仿真^[8]。Maple 提供了超过 5 000 个符号和数值计算命令，覆盖微积分、线性代数、方程求解和概率论等多个数学领域，支持工程计算、数据可视化和编程。MapleSim 基于符号—数值混合计算技术，能够处理多领域复杂系统的建模和仿真任务，还提供广泛的预置建模元件库和专业工具箱，支持拖放式物理建模，使用户可以快速创建系统级模型。MapleSim 具备强大的多体建模功能、代码生成功能和单位管理工具，能够自动生成符号形式的系统方程和实时仿真代码。Maple/MapleSim 的紧密集成使用户可以利用 Maple 的数学引擎对 MapleSim 的仿真结果进行深入分析，同时 MapleSim 为 Maple 的数学模型提供了直观的可视化和仿真环境。Maple 在符号计算方面具有很大的优势，适合应用于数学研究、理论计算以及需要符号化推导的研究。然而，在面对需要大量数值计算的工程问题时，Maple 的表现往往不尽如人意，特别是在处理大规模数值数据和优化计算时，它的数值计算能力较为有限。

1.4 OpenModelica

OpenModelica 是一款开源的建模与仿真工具，主要用于处理基于 Modelica 语言的复杂系统建模问题^[9]。Modelica 是一种开放的、面向对象的物理建模语言，支持多领域复杂系统的建模与仿真，广泛应

用于机械、电气和热力学等多个领域。Modelica 的开发始于 1996 年，由非营利组织 Modelica 协会负责开发和维护，并公开其标准库。其核心优势在于面向对象的建模能力，允许用户通过定义类和对象来构建复杂的系统模型，支持多种数据类型和算法，能够处理从简单物理系统到复杂工程系统的建模需求。Modelica 标准库包含不同物理领域的 920 个元件模型，具有 620 种功能。此外，Modelica 模型可以导出多种格式，如 Simulink 的 S-Function、符合功能样机接口（FMI）标准的功能样机单元（FMU）、主流实时仿真机需要的实时模型格式以及 C 代码等。Modelica 协会虽然是一个国际性组织，但参与的企业和机构大多数来自欧美及其他发达国家，特别是一些大型的软件公司和研究机构。这意味着这些国家和地区对标准的制定与发展有较大的影响力，在某些情况下，这种国际主导会影响特定地区或国家在该标准发展中的话语权。另外，虽然 Modelica 本身是一个开放的标准，但很多应用与开发工具（如 Dymola、OpenModelica、SimulationX 等）由欧美的公司或机构主导，如果基于这些工具进行建模与仿真，会在技术上形成依赖，尤其是在自主可控方面存在技术瓶颈或隐性风险。

1.5 北太天元

北太天元采取“北京大学+北京大学重庆大数据研究院+公司”三位一体的运营模式，深度融合了北京大学“0~1”自由探索型学术与科学研究积累、北京大学重庆大数据研究院基础软件科学研究中心“1~10”需求型科学研究能力与北太振寰（重庆）科技有限公司“10~N”成果转化能力，实现了高效协同的理论转化与科技创新协同发展。北京大学凭借深厚的学术底蕴和前沿科研成果，为该模式提供坚实的理论支撑与高端人才保障。北京大学重庆大数据研究院作为核心枢纽，精准对接高校与企业需求，承接科研成果转化任务，开展定制化研究，推动技术深化与应用落地。企业则加速技术产品化与产业化进程，以市场反馈反哺科研环节，实现持续迭代。

该运营模式本质上是一种以国家战略与产业需求为牵引、以落地转化为核心导向的数学创新路径探索，打通了从理论研究、人才培养到技术转化的全链条。北太天元通过学术创新链、人才创新链与产业创新链的互联互通，回应了张平文院士提出的国家需求驱动数学发展的核心思想：一方面将前沿

理论主动对接企业和行业中的关键技术难题，另一方面通过企业问题反向激发理论创新与交叉融合，推动应用数学真正“走进工程、落到一线、服务国家”^[10]。在该体系推动下，北太天元作为国内首个自主可控的科学计算软件平台实现了快速发展。北太天元平台以数学建模、仿真求解与运筹优化为核心，通过内核算法与高性能计算架构的深度融合，以“基础研究+技术产品化”为路径，构建从上游基础研发、中游技术集成到下游行业应用的全链条系统。在上游，北太天元平台已实现对主流国产操作系统（如银河麒麟、统信等）和芯片（如龙芯、鲲鹏、飞腾等）的全面兼容；与摩尔线程联合开发图形处理器（GPU）加速方案，完成元计算统一系统架构（MUSA）适配，构建了从算法到硬件的全栈优化体系；完成对国家超算平台的适配，夯实在信创环境中的核心算力地位。在中游，北太天元平台推动与高校、研究机构及企业的联合攻关，构建组件化工具链与模型库，提升通用性与行业适应性。目前，北太天元已在 156 所高校实现部署，建设 18 个产学研合作基地，设立 1 个校企研发中心，完成 10 所高校的正版化平台建设，初步形成“科研—教学—产业”联动的协同生态。该平台成功入选工业和信息化部首批先进适用技术名单，标志着其自主核心技术能力与产业服务价值获得国家层面认可。在下游，聚焦定制化解决方案，通过插件机制满足不同应用场景需求。例如，北太天元平台已完成与杉数 COPT 求解器的适配，支持线性规划与二阶锥规划等复杂模型的构建与求解，显著提升科研建模与工程优化能力。

北太天元是一款面向科学计算与系统仿真的国产通用型软件，具备强大的科学计算、系统建模与仿真能力。2020 年，北京大学数学科学学院开始进行内核技术研究，随后在大数据分析与应用技术国家工程实验室进行技术孵化；2022 年，北太天元 2.0 版本正式发布；2023 年，北太天元 3.0 版本发布，包含重量级工具箱——北太真元^[11]。

北太天元提供了科学计算、可视化与交互式程序设计环境，具备丰富的底层数学函数库，支持数值计算、数据分析、数据可视化、数据优化和算法开发等工作。北太真元则基于北太天元内核，提供完善的模块化建模仿真环境，配备多类基础、行业模块库、定步长和变步长求解器，支持复杂系统的模块化建模、仿真与验证。北太真元还提供高精度、

高效率的求解器，支持定步长、变步长求解仿真模型，求解器参数支持用户自定义配置和系统自动配置。此外，北太真元提供用户友好型图形化开发环境，用户可以通过拖放、连接和配置图形元素来建

立复杂系统模型。北太天元凭借强大的功能和自主可控的特点，成功打破了国外企业垄断国内科学计算软件市场多年的局面^[12]。北太天元软件界面如图 1 所示。

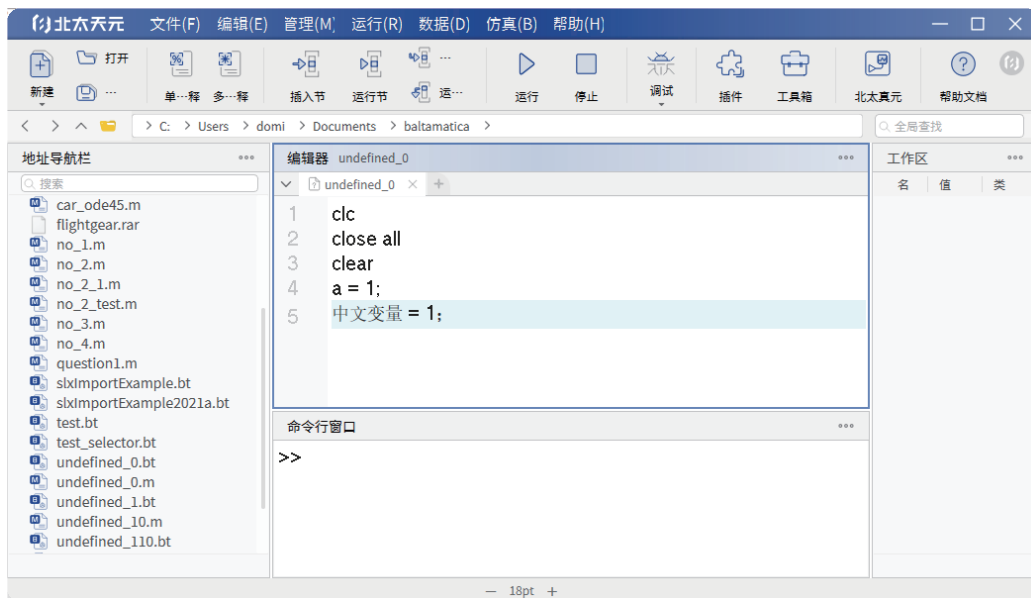


图 1 北太天元软件界面

北太天元目前完成了 30 多门课程建设和 30 多本教材建设，涉及数值方法、数值分析、机器学习、数学实验、数字图像处理、最优化方法、数学建模、线性代数和高等数学等课程。其中北京大学的《北太天元科学计算编程与应用》、四川大学的《线性

代数》、北京理工大学的《数值计算方法》、陕西科技大学的《数学建模 基于北太天元软件》、四川建筑职业技术学院的《线性代数》已正式出版发行（图 2）。



图 2 出版教材

1.6 MWORKS

MWORKS 是一款集科学计算与系统建模仿真功能于一体的工业软件平台，应用于航天、航空、能源、车辆、船舶和教育等多个领域^[13]。2008 年同元软控在苏州工业园区成立，2009 年正式发布商业软件 MWORKS，2010 年 MWORKS 进入航空航

天领域，2018 年形成系统设计与仿真验证平台。MWORKS 包含科学计算环境 Syslab，基于科学计算语言 Julia，提供通用编程与算法开发、数据分析与可视化等功能。其系统仿真环境 Sysplorer 基于 Modelica 建模语言，提供物理/框图/状态机等多范式系统建模、编译分析和仿真求解等功能。从技

术结构上看, MWORKS 是国外开源计算语言 Julia 和系统建模语言 Modelica 的结合体, 整个系统采用了 Julia 作为内核, 其底层科学计算功能的支撑是 Julia, 建模则采用 Modelica 语言, 这样的结合在一定程度上实现了科学计算与模拟仿真的功能。如果其内核依赖的语言 Julia 和仿真依赖的建模语言 Modelica 一直得到国外的支持, MWORKS 将可能成为科学计算和模拟仿真软件的一个选项。

1.7 GCKontrol

GCKontrol 是一款国产系统设计与仿真软件。2003 年, 世冠科技成立并引进国际先进技术; 2019 年, 世冠科技自主研发 GCAir, 持续推出 1.0 ~ 5.0 版本, 并成功将其应用于多家航空航天头部用户; 2024 年, 世冠科技发布 GCKontrol & GCAir 9.0, 取得 ISO 26262 认证^[14]。GCKontrol 能够提供多元化模型库, 支持图形化信号流建模、状态图建模, 用户可快速搭建复杂系统模型并进行仿真, 可以通过 C-Code 模块、M-Code 模块将 C/C++ 代码和 M 脚本导入。GCKontrol 能生成满足 MISRA-C 代码规范的高质量 C 代码, 支持定点计算, 满足嵌入式系统的需求, 可以提供丰富的模型调试和分析功能、2D 可视化面板, 支持在 2D 面板配置所需显示的模型变量, 能够把曲线导出为矢量图片和数据。技术上仅使用 python 和 C++ 语言, 并且完全由国内自主研发, 自主可控方面的风险很小。

2 科学计算与模拟仿真软件的关键技术——以北太天元为例

目前国内已有一些科学计算与模拟仿真软件,

但是这些软件往往都存在一些不可忽视的问题: 一类软件专注于模拟仿真, 但因缺少科学计算内核, 用户在仿真时无法便捷地使用数值分析工具进行数据处理; 另一类软件既有科学计算功能, 又有模拟仿真能力, 但均依赖国外成熟的开源软件或语言标准, 这类软件不仅在科学计算和模拟仿真的整合上存在不足, 而且无法从根本上解决技术安全隐患问题。由本文作者所在团队开发的北太天元软件是唯一一款技术全栈自主的科学计算与模拟仿真软件, 因此本文以北太天元软件为例总结国产自主科学计算与模拟仿真软件的关键技术。

2.1 科学计算语言内核

科学计算语言内核的基本组件之一是解释器。解释器是科学计算语言的实际执行者, 其本质是将源代码转化为计算机可执行的程序指令。与一般的解释型编程语言不同, 科学计算语言会引入更多的数学符号, 从而导致一些符号具有多义性。例如, 科学计算语言中的单引号既可以表示字符的开始或结束, 又可以表示矩阵的转置。当单引号表示字符的开始或结束时, 其类型为有界限符; 当单引号表示矩阵的转置时, 其类型为运算符。单引号作为不同的类型出现时, 源代码生成的语法树和语法树节点具有本质区别。在一般解释型编程语言架构的基础上, 解释器只会将单引号当作固定的一种类型处理, 从而导致错误。科学计算语言的解释器与一般编程语言的解释器不同, 其有特殊的工作流程, 如图 3 所示。解释器的算法伪代码如表 2 所示。



图 3 科学计算语言中解释器的工作流程

表 2 解释器工作算法

工作阶段	处理过程
词法处理	①将字符流转换为词法单元序列 ②标记歧义词法单元和非歧义单元
语义预处理	③对歧义语法单元进行上下文相关类型判定, 更新属性值
语法解析	④使用自适应 LL(k) 算法构建分析树, 特征包括: 动态确定向前看的符号数 k ($k \geq 1$); 自动转换左递归文法
语义生成	⑤基于符号表信息精简语法树节点, 生成最终抽象语法树

科学计算语言通常具备弱类型特性,这意味着变量无须预先声明即可直接赋值或使用,并且在程序的执行过程中,变量的类型可能会发生改变。这种灵活性为用户提供了极大的便利,使其能够迅速实现各种计算任务,但也给语言实现带来了较大的挑战,如变量需携带动态类型信息导致内存效率降低,类型多态导致即时编译难以推导稳定类型特化路径,以及静态分析器需模拟执行流推导类型导致可靠性下降等。为应对这些挑战,北太天元提出了一种统一的数据类型。该数据类型在设计中同时包含了实际的数据类型和数据指针,从而确保无论变

量在程序运行中如何变化,其基本数据结构和内存管理方式始终保持一致。这种统一的数据类型作为语法树节点的标准中间表示(IR),保证了不同计算环境下结果的一致性。在解释过程的后续阶段,语言会根据具体需求定义数组、标量和内部类型。类型定义不仅有助于优化内存使用和计算性能,还能提供更为精确的数据操作和控制。统一数据类型的结构通常包括类型标识符、数据值以及可能的其他元数据,如数据精度、维度等。这种结构设计允许语言在保持灵活性的同时,提供必要的数据类型安全性。统一数据类型的结构示意图如图4所示。

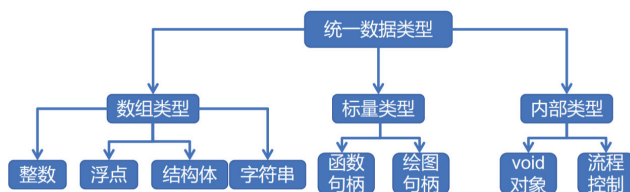


图4 统一数据类型的结构示意图

2.2 仿真引擎

国内相关机构在开发国产自主可控仿真软件时通常面临多重困境:一是底层数学算法库(如大规模稀疏矩阵求解、实时仿真引擎)的自主性薄弱;二是多域物理系统耦合建模的核心求解器(如Modelica编译器)高度依赖国外开源技术且难以突破性能瓶颈。北太真元的仿真引擎基于北太天元的计算能力,在底层数学算法库上有天然优势,在此基础上进一步结合求解器开发技术实现了自主可控仿真引擎的设计与开发。

模拟仿真软件中搭建的通常是一个动态系统的图形化模型,与控制理论中的方框图类似,其由一系列被称为计算单元的符号组成,每一个计算单元代表一个基本的动态系统。计算单元既可以是连续的,也可以是离散的,这代表一类基本动态系统的输出和输入与状态的关系。图形化模型中的计算单元通过线相互连接,这些线也代表动态系统间信号的流向。通常,模拟仿真软件会提供大量内置的计算单元,用户只需要从库浏览器中选择组成模型的计算单元并进行连接即可完成模型的搭建。模型通常可以抽象为:

$$\begin{cases} \frac{dx}{dt} = f(t, x, z, u, p), x(t_0) = x_0 \\ Y = g(t, x, z, u, p) \end{cases} \quad (1)$$

式中, $x \in \mathbb{R}^n$, 为状态变量; $u \in \mathbb{R}^m$, 为输入信号;

$p \in \mathbb{R}^k$, 为时变或定常的参数; Y 为模型的输出; t 为时间变量; t_0 为初始时刻; z 为代数变量, 当模型中不包含代数环(Algebraic Loop)或隐式关系时, z 为空; x_0 为系统中各个状态的初值; f 和 g 为非线性函数, 由计算单元的连接关系与代数运算定义。

动态系统的仿真分为解析阶段和执行阶段。北太真元在初始化阶段通过多个步骤完成模型解析工作, 先计算模型中各计算单元的参数表达式以确定其数值, 随后对未显式定义数据类型和维度的信号属性进行推导, 并验证各计算单元接口对输入信号的兼容性, 在此过程中采用属性传播机制将源信号属性传递至下游计算单元输入端, 以解决未指定属性问题。之后计算每个计算单元的中间向量、状态变量及运行时参数的内存需求, 执行删除冗余计算单元、合并连续运算等精简优化操作, 将子系统替换为其内部计算单元, 实现模型层级扁平化处理, 基于数据依赖关系拓扑排序确定计算单元执行顺序, 推导未明确指定采样时间的计算单元基准速率, 最终完成状态变量存储空间分配并按照计算单元初始条件进行内存初始化。仿真引擎的整体解析流程如图5所示。上述解析阶段会将模型解析为计算机可执行的语言, 后续的执行阶段会按照解析结果进行相应计算。

2.3 行业工具包扩展

为扩展生态, 北太天元还提供了开发者工具

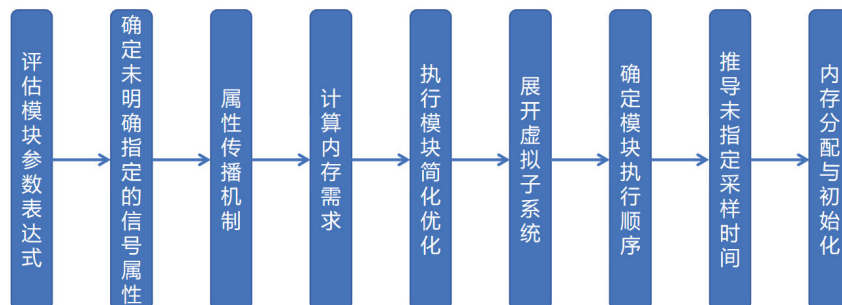


图5 仿真引擎的整体解析流程

(SDK), 允许用户和开发者基于软件主体开发不同类型的拓展功能。开发者可通过 SDK 直接访问北太天元的底层数据, 将自己的 C/C++ 程序和 Fortran 程序整合到软件中直接调用。SDK 随北太天元主体发布, 提供北太天元底层的各类接口。用户可利用 SDK 直接调用北太天元底层的扩展功能模块。SDK 现已有 200 余个接口函数, 支持不同类型的矩阵、字符串、结构体和元胞数组等底层数据的访问。SDK 提供包装编译器 `bex`, 用户可使用 `bex` 快速编译出适用于北太天元的扩展模块。`bex` 利用 SDK 将现有的 C/C++/Fortran 代码编译为北太天元可调用的函数模块 (BEX 函数), 其使用方式与内置函数和脚本相同。例如, 如果用户将 C 代码 `create.c` 编译为 BEX 文件 `create.bexa64` (Linux 平台), 那么软件中就可以使用名称 `create` 来调用相应 BEX 函数。

基于北太天元先进的内核和开发者工具, 可以为不同应用领域开发专门的功能和工具, 这些功能和工具的组合作为工具箱。北太天元现有的工具箱有优化工具箱、符号计算工具箱、曲线拟合工具箱、图像处理工具箱^[15]、统计工具箱、数值积分和微分方程工具箱等。

3 北太天元应用案例

当前, 北太天元已在教育、汽车和航空航天等领域得到广泛的应用。Wang 等^[16]基于北太天元使用分片多项式配置方法数值求解 Caputo 分数阶微分方程 (FDEs); 江雪等^[17]将北太天元应用在数值计算方法教学中; 向导等^[18]使用北太天元对分数阶比例—积分—微分 (PID) 控制器进行了优化, 该研究团队还使用北太天元对汽车制动防抱死装置控制系统的算法进行了研究^[19]。本节主要以两个应用案例说明北太天元的应用场景。

3.1 基于卡尔曼滤波的全球定位系统信号处理

卡尔曼滤波 (Kalman filtering) 是一种利用线性系统状态方程, 通过系统输入输出观测数据, 对系统状态进行最优估计的算法。卡尔曼滤波常用于融合传感器数据、预测未来状态以及对噪声进行滤波处理。

卡尔曼滤波器的递归过程如下。

(1) 估计时刻 k 的状态:

$$X(k) = AX(k-1) + Bu(k)。 \quad (2)$$

式中, X 为采集的全球定位系统 (GPS) 信号; A 、 B 分别为系统状态矩阵和输入矩阵; u 为输出控制变量。

(2) 计算误差相关矩阵 P , 度量估计值的精确程度:

$$P(k) = A^*P(k-1)A^* + Q。 \quad (3)$$

式中, Q 为系统噪声的协方差阵。

(3) 计算卡尔曼增益, 以下略去 (k) :

$$K = PC^*(C^*P^*C^* + R)^{-1}, P = P(k), X = X(k)。 \quad (4)$$

式中, R 为测量噪声的协方差。由于 GPS 信号是单输入单输出, 所以 R 是一个常数, C 为输出矩阵, 式 (4) 可以简化为:

$$K = PC^* / (CP^*C^* + R)。 \quad (5)$$

(4) 状态变量反馈的误差量:

$$e = Z(k) - C^*X(k)。 \quad (6)$$

式中, $Z(k)$ 是带噪声的测量, 在本案例中, 一维输入信号 $C=1$ 。

(5) 更新误差相关矩阵 P 和状态变量 X :

$$P = P - K^*C^*P, \quad (7)$$

$$X = X + K^*e = X + K(Z(k) - C^*X(k))。 \quad (8)$$

(6) 最后输出:

$$Y = C^*X。 \quad (9)$$

在北太天元软件中编写卡尔曼滤波算法, 结果如图 6a 所示, 可以发现卡尔曼滤波去除了原始信号曲线中的噪声和毛刺, 使得曲线更为平滑。在

MATLAB 中执行同样的处理程序并将结果导入北太天元中进行对比。MATLAB 的运行结果如图 6b 所示, 误差对比如图 6c 所示, 误差对比如图 6c 所示, 北太天元的运行结果与

MATLAB 的误差为 0, 说明本示例中北太天元的计算精度已达到国际一流水平。

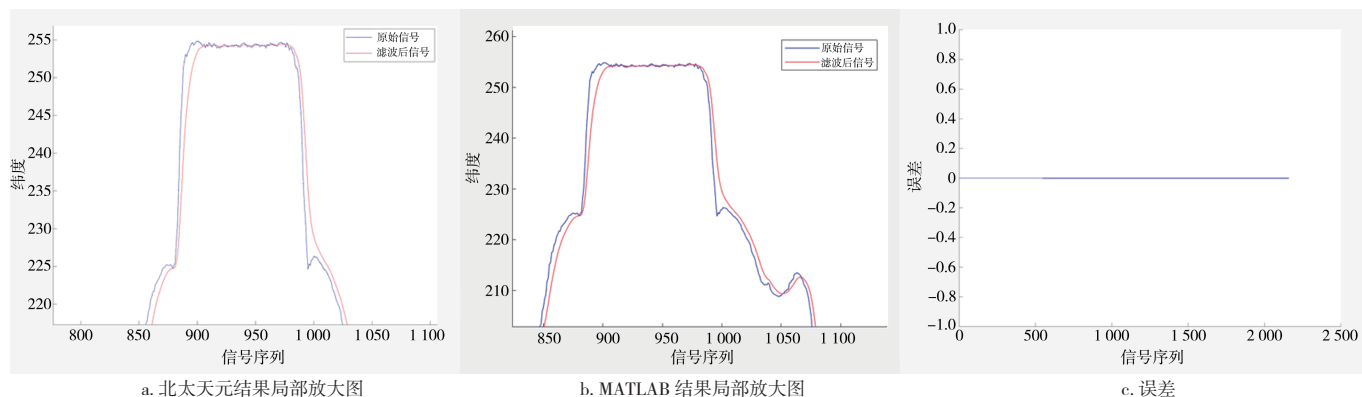


图 6 北太天元与 MATLAB 运行结果对比

3.2 四自由度半车振动模型

汽车的行驶平稳性直接关系到乘客的舒适感, 也对车辆动力性和经济性产生影响, 因此是车辆在市场竞争力中争取优势的一项重要性能指标。由此, 如何将汽车在行驶过程中产生的振动降到最低, 成为汽车行业的研究焦点。将复杂的多自由度车辆振动模型简化为四自由度半车振动模型, 其仿真结果不仅能解释汽车行驶过程中的典型振动现象机制, 还为优化悬架设计、提升乘坐舒适性和安全性提供量化依据。汽车四自由度半车振动模型如图 7 所示。

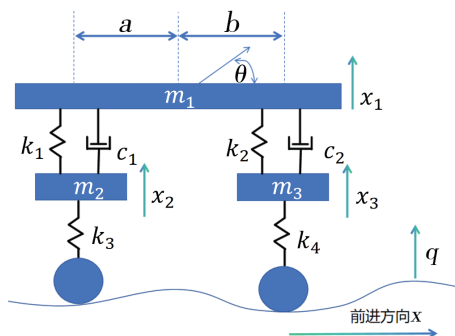


图 7 四自由度半车振动模型

注: m_1 为车体质量, m_2 为前车轮质量, m_3 为后车轮质量, k_1 和 k_2 是悬架刚度, k_3 和 k_4 是轮胎刚度, c_1 和 c_2 是悬架阻尼, a 是质心到后轮的距离, b 是质心到前轮的距离, θ 是车体的俯仰角, x_1 、 x_2 、 x_3 分别为车体、前轮和后轮的垂向位移, q 为路面垂向位移。

对车体进行受力分析:

$$m_1 \ddot{x}_1 = [x_2 - (x_1 - \theta a)]k_1 + [\dot{x}_2 - (\dot{x}_1 - \dot{\theta}a)]c_1 + [x_3 - (x_1 + \theta b)]k_2 + [\dot{x}_3 - (\dot{x}_1 + \dot{\theta}b)]c_2, \quad (10)$$

$$J\ddot{\theta} = [x_3 - (x_1 + \theta b)]k_2b + [\dot{x}_3 - (\dot{x}_1 + \dot{\theta}b)]c_2b - [x_2 - (x_1 - \theta a)]k_1a - [\dot{x}_2 - (\dot{x}_1 - \dot{\theta}a)]c_1a. \quad (11)$$

对车轮 m_2 进行受力分析:

$$m_2 \ddot{x}_2 = [q(t - \tau) - x_2]k_3 - [x_2 - (x_1 - \theta a)]k_1 - [\dot{x}_2 - (\dot{x}_1 - \dot{\theta}a)]c_1. \quad (12)$$

对车轮 m_3 进行受力分析:

$$m_3 \ddot{x}_3 = [q(t) - x_3]k_4 - [x_3 - (x_1 + \theta b)]k_2 - [\dot{x}_3 - (\dot{x}_1 + \dot{\theta}b)]c_2. \quad (13)$$

式中, J 为转动惯量; τ 为后轮时延, $\tau = (a + b)/v$, v 是车速。

根据上述分析, 在北太真元中搭建仿真模型 (图 8)。



图 8 四自由度半车仿真模型

四自由度半车仿真模型的车身点头角仿真输出结果如图 9a 所示, 北太真元能准确对四自由度半车振动模型进行模拟仿真, 且对指定信号进行输出和可视化。在 Simulink 中对相同模型进行仿真并将结果导入北太天元进行对比。Simulink 的仿真结果如图 9b 所示, 误差对比如图 9c 所示。结果显示, 北太真元与 Simulink 的误差小于 10^{-12} , 说明本示例中

北太真元的仿真已达到国际一流水平。

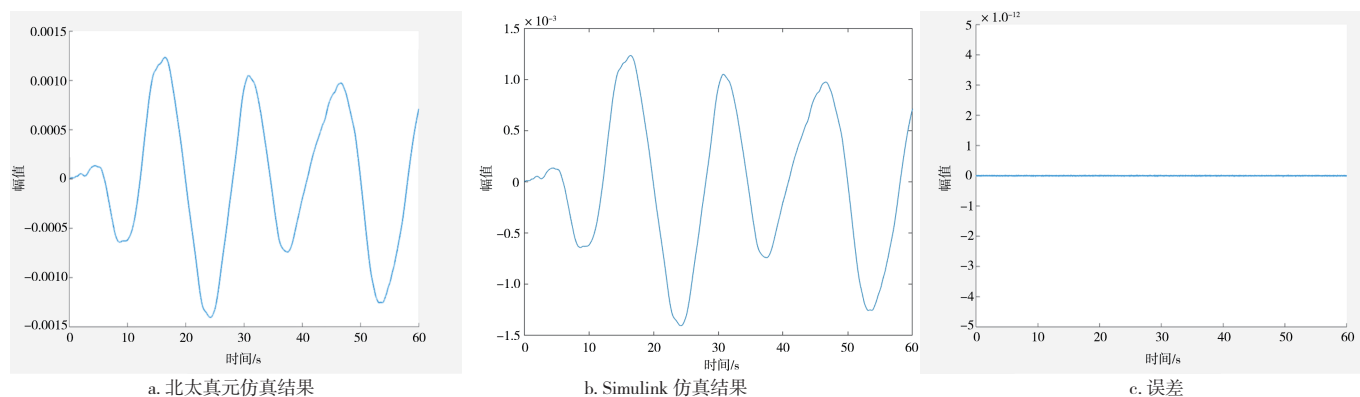


图9 车身点头角的北太真元仿真结果及其与 Simulink 仿真结果的对比

4 结语

本文系统梳理了科学计算与模拟仿真软件的国内外研究现状，分析了我国在该领域面临的“解释器—算法库—生态体系”三重困境，并以北太天元为例，深入探讨了国产自主科学计算与模拟仿真软件的技术架构及其核心能力。通过对科学计算语言内核、仿真引擎和工具箱扩展等关键技术进行详细阐述，展示了北太天元在打破国外技术垄断、推动国产软件从“可用”向“好用”转变方面的技术优势和应用潜力。

本文通过基于卡尔曼滤波的 GPS 信号处理和四自由度半车振动模型的仿真研究，展示了北太天元在实际工程问题中的应用效果。这些案例表明，北太天元不仅能够满足复杂的科学计算与模拟仿真需求，还具备良好的扩展性和用户友好性，为国产科学计算与模拟仿真软件的产业化发展提供了有力支持。

本文的研究表明国产科学计算与模拟仿真软件的发展需要从技术突破、生态建设和应用推广等多个维度协同推进。未来，可进一步加强基础研究，完善技术生态，推动科研成果的产业化应用，为我国科技自立自强和战略性新兴产业的发展提供坚实的“数字底座”。

参考文献

- [1] 薛岩. 国产科学计算软件产业：悄然崛起未来可期 [N]. 科技日报, 2024-05-14(6).
- [2] 环球网. 哈工大等被禁用美国软件 MATLAB, 华春莹：任何打压都无法阻挡中国科技发展进步的步伐 [J]. 船舶标准化与质量, 2020(3): 69.

- [3] MOLER C, LITTLE J. A history of MATLAB [J]. Proceedings of the ACM on programming languages-pacmpl, 2020, 4(HOPL): 1–67.
- [4] MathWorks. Simulink [EB/OL]. [2025-04-24]. <https://www.mathworks.cn/products/simulink.html>.
- [5] Scicos Team. History of Scicos [EB/OL]. [2025-04-24]. <http://www.scicos.org/history.html>.
- [6] Scilab Enterprises. Xcos: Scilab's dynamic systems modeler and simulator [EB/OL]. [2025-04-24]. <https://www.scilab.org/software/xcos>.
- [7] Maplesoft. About Maplesoft [EB/OL]. [2025-04-24]. <https://www.maplesoft.com.cn/about/index.shtml>.
- [8] Maplesoft. History of Maple [EB/OL]. [2025-04-24]. <https://www.maplesoft.com/products/maple/history/>.
- [9] Open Source Modelica Consortium. OpenModelica [EB/OL]. [2025-04-24]. <https://openmodelica.org/>.
- [10] 张平文. 应用数学落地研究—国家需求驱动应用数学发展 [M]. 北京：高等教育出版社, 2024: 22–24.
- [11] 李迪, 周金钢. 北太天元的探索之路 [J]. 数值计算与计算机应用, 2025, 46(1): 24–43.
- [12] 蔡雨耘, 杨士达. 北太天元逐梦中国“自”造 [J]. 当代党员, 2023(15): 29–30.
- [13] 同元软控. 同元软控官方网站 [EB/OL]. [2025-04-24]. <https://www.tongyuan.cc/>.
- [14] 世冠科技. 发展历程 [EB/OL]. [2025-04-24]. <https://www.globalcrown.com.cn/history/>.
- [15] 佚名. 数学科学学院郭蔚团队开发首款国产通用型科学计算软件图像处理工具箱 [J]. 河北师范大学学报 (自然科学版), 2024, 48(2): 104.
- [16] WANG L, LIANG H. Analysis of direct piecewise polynomial collocation methods for the Bagley-Torvik equation [J]. BIT numerical mathematics, 2024, 64(4): 1–29.
- [17] 江雪, 黄秋梅. 北太天元在数值计算方法教学中的应用 [J]. 数学

的实践与认识, 2024, 54(2): 226–231.

- [18] 向导, 廖汉卿, 郭佳豪. 基于淘金算法和分数阶 PID 控制的汽车半主动悬架研究 [C/OL]. [2025-04-24]. <https://read.cnki.net/web/Conference/Article/GXRH202411001011.html>.
- [19] 向导, 张敏, 廖汉卿, 等. 基于北太天元 / 真元的汽车 ABS 控制系统算法研究 [C]// 第三十六届中国仿真大会论文集. 北京: 中国仿真学会, 2024: 320–325.

Research on Autonomous and Controllable Scientific Computing and Simulation Software

LI Ruo¹, XIANG Dao², WANG Cheng²

1. School of Mathematical Sciences, Peking University, Beijing 100871;
2. Beitaizhenhuan (Chongqing) Technology Co., Ltd, Chongqing 401329

Abstract: Scientific computing and simulation software, as the “digital foundation” supporting strategic industries such as aerospace, intelligent connected vehicles, and new power systems, is the most important foundational software. Basic software is a key area and weak link in the national science and technology and industrial system. In the new era of accelerating changes over the past century, its independent and

controllable nature has become a core proposition related to national scientific and technological security. This paper systematically reviews the technological evolution of global scientific computing and simulation software over the past half-century, revealing the “triple bottleneck” challenges faced by China in this field, specifically in interpreters, algorithm libraries, and ecosystem systems. Taking the Baltamatica scientific computing and simulation software, which has full independent intellectual property rights, as the research carrier, this paper introduces the technical architecture of domestic independent scientific computing and simulation software, and illustrates the application scenarios of Baltamatica with two application cases. The aim is to offer a replicable technical paradigm and industrial ecosystem development strategy for the transition of domestic software from “usable” to “high-quality,” while exploring mechanisms for bidirectional empowerment between the translation of scientific research achievements and technological self-reliance.

Keywords: domestically developed and independent; scientific computing; Baltamatica; technological innovation