

# 机理与数据驱动的物理仿真计算范式及引擎架构

何小伟<sup>1</sup>, 石剑<sup>2</sup>, 刘树森<sup>1</sup>, 任丽欣<sup>1</sup>, 郭煜中<sup>1</sup>, 蔡勇<sup>3</sup>,  
王琥<sup>3</sup>, 朱飞<sup>4</sup>, 汪国平<sup>4</sup>

- 中国科学院软件研究所, 北京 100190;
- 中国科学院自动化研究所, 北京 100190;
- 湖南大学汽车车身先进设计制造国家重点实验室, 湖南 长沙 410082;
- 北京大学计算机学院, 北京 100871)

**摘要:** 物理仿真作为现代工业软件的基石, 其计算范式可分为机理驱动、数据驱动及混合驱动等模式。面对多样化物理仿真需求, 构建一个既能灵活适应各类物理仿真计算范式, 又能实现不同计算范式之间高效耦合的通用引擎架构, 已成为软件设计与开发领域亟待解决的关键难题与挑战。针对该问题, 提出面向多物理仿真计算范式的FNMS架构 Data Field-Node-Module-Scene Graph, 其核心在于四层结构的设计: 数据域(Data field)、节点(Node)、模块(Module)与场景图(Scene graph)。具体而言, 数据域层为仿真过程提供统一的数据管理与访问接口, 解决物理仿真计算数据共享的便捷性与高效性; 模块层封装各类物理仿真算法, 实现算法的模块化与可重用, 解决仿真计算、渲染与交互的异步协同问题; 节点层通过数据与算法模块的解耦实现算法在不同物理仿真计算范式之间的复用, 同时便于实现多物理场耦合过程的交换与共享; 而场景图层通过将节点组织成有向无环图, 支撑多种物理仿真计算范式的高效耦合计算。通过该四层结构的结合, FNMS架构不仅能提升物理仿真的计算效率与灵活性, 更为跨学科、跨领域的物理仿真研究提供了强有力的技术支持。

**关键词:** FNMS引擎架构; 机理驱动; 数据驱动; 物理仿真; 计算范式

中图分类号: TP 391

DOI: 10.11996/JG.j.2095-302X.2024061207

文献标识码: A

文章编号: 2095-302X(2024)06-1207-15

## The computational paradigm and software framework for mechanism and data-driven physical simulation

HE Xiaowei<sup>1</sup>, SHI Jian<sup>2</sup>, LIU Shusen<sup>1</sup>, REN Lixin<sup>1</sup>, GUO Yuzhong<sup>1</sup>, CAI Yong<sup>3</sup>,  
WANG Hu<sup>3</sup>, ZHU Fei<sup>4</sup>, WANG Guoping<sup>4</sup>

(1. Institute of Software, Chinese Academy of Sciences, Beijing 100190, China;

2. Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China;

3. State Key Laboratory of Advanced Design and Manufacturing for Vehicle Body, Hunan University, Changsha Hunan 410082, China;

4. School of Computer Science, Peking University, Beijing 100871, China)

**Abstract:** As the cornerstone of modern industrial software, physical simulation encompasses various computational

收稿日期: 2024-07-31; 定稿日期: 2024-10-05

Received: 31 July, 2024; Finalized: 5 October, 2024

基金项目: 国家重点研发青年科学家项目(2021YFB1715800); 国家自然科学基金(62302490)

Foundation items: National Key Research and Development Program of China (2021YFB1715800); National Natural Science Foundation of China (62302490)

第一作者: 何小伟(1985-), 男, 研究员, 博士。主要研究方向为计算机图形学与物理仿真。E-mail: xiaowei@iscas.ac.cn

First author: HE Xiaowei (1985-), researcher, Ph.D. His main research interests cover computer graphics and physical simulation.

E-mail: xiaowei@iscas.ac.cn

通信作者: 汪国平(1964-), 男, 教授, 博士。主要研究方向为计算机图形学、虚拟现实与物理仿真。E-mail: wgp@pku.edu.cn

Corresponding author: WANG Guoping (1964-), professor, Ph.D. His main research interests cover computer graphics, virtual reality and physical simulation. E-mail: wgp@pku.edu.cn

paradigms, including mechanism-driven, data-driven, and hybrid-driven models. Meeting the demands of diverse physical simulation requires the construction of a general framework capable of flexibly adapting to various physical simulation computational paradigms while achieving efficient coupling across various computational paradigms, presenting a critical challenge in software design and development. To address this, the Data field—Node—Module—Scene graph (FNMS) architecture was proposed, targeting multi-physics simulation computational paradigms. Its core lies in the design of a four-layer structure: Data field, Node, Module, and Scene graph. Specifically, the Data field layer provides a unified data management and access interface for the simulation process, enhancing the convenience and efficiency in data sharing for physical simulation computations. The Module layer encapsulated various physical simulation algorithms, realizing algorithm modularization and reusability while solving the asynchronous coordination of simulation computation, rendering, and interaction. Through data and algorithm decoupling, the Node layer enabled algorithm reuse across different physical simulation computational paradigms, and it facilitated the exchange and sharing within multi-physics coupling processes. The Scene graph layer supported efficient coupled computations of various physical simulation computational paradigms by organizing nodes into a directed acyclic graph. Through the combination of these four layers, the FNMS architecture not only enhanced the computational efficiency and flexibility in physical simulations but also provided strong technical support for interdisciplinary and cross-domain physical simulation research.

**Keywords:** FNMS framework; mechanism driven; data driven; physical simulation; computing paradigm

物理仿真指利用数学建模工具对真实物理系统所涉及的物体形状及其内在物理属性的状态和行为进行数学描述, 并对其数学描述进行数值离散、求解并分析的技术手段。物理仿真技术广泛应用于航空航天、医疗诊断、机器人和国防军事等工程设计与试验验证领域, 以及影视游戏、数字城市等数字娱乐与设计体验领域。例如在医疗领域, 物理仿真技术结合医学影像技术能为肝门静脉高压患者提供无创辅助诊断, 大幅降低患者诊疗成本<sup>[1]</sup>。在航空航天领域, 物理仿真技术可为飞行器设计、可靠性验证等提供重要科学依据<sup>[2]</sup>。相比于物理实验手段, 物理仿真技术具备多方面优势。首先, 物理仿真可大幅降低物理实验成本。如宝马集团在 GTC 2023 展示了英伟达 Omniverse 驱动的虚拟工厂<sup>[3]</sup>, 通过物理真实的仿真及可视化技术, 大幅消减了物理测试的时间, 预期可降低约 30% 的成本。其次, 物理仿真可扩展物理实验的边界, 提供极端测试条件下的性能判定依据<sup>[4]</sup>。此外, 物理仿真也能大幅降低专业领域准入门槛, 促进跨学科协作与交流<sup>[5]</sup>。

物理仿真相关技术的发展最早可追溯至上世纪 20 年代<sup>[6]</sup>。尽管彼时计算机技术尚未发展, COURANT 等<sup>[7]</sup>提出基于有限差分(finite different method, FDM)的思想用于验证拉普拉斯方程解的收敛性, 对后续偏微分方程的数值分析与求解产生了深远的影响。二战后, 伴随着计算机的发展, 较

大规模的偏微分方程数值求解逐渐成为可能, 包括早期的热传导问题<sup>[8]</sup>、非线性双曲方程求解<sup>[9]</sup>等, 以及近年来在不可压缩流体<sup>[10]</sup>、烟雾<sup>[11]</sup>、火焰模拟<sup>[12]</sup>等方面的广泛应用。FDM 具有易理解及实现简单等优势, 但处理复杂几何边界时易导致精度下降。因此, 有限元法(finite element method, FEM)<sup>[13-15]</sup>通过将计算域划分成互不重叠且与计算域边界契合的单元, 并采用变分原理进行数值建模与求解。FEM 在包括超弹性材料仿真<sup>[16]</sup>、塑性建模<sup>[17]</sup>等方面均展现了良好的性能。然而, 在处理金属冲压或冲击起爆等大变形问题时, FEM 求解精度和效率会受网格扭曲畸变问题影响。如不对网格进行重新划分可能导致精度降低甚至仿真失效。针对 FEM 在处理大变形问题时验证依赖网格剖分的效率和质量的问题, 无网格法应运而生, 其中最为典型的为光滑粒子动力学(smoothed particle hydrodynamics, SPH), 最初由 GINGOLD 和 MONAGHAN<sup>[18]</sup>以及 LUCY<sup>[19]</sup>提出用于求解星体运行问题。SPH 因其在处理大变形方面的优势被广泛用于模拟不可压缩流体<sup>[20-23]</sup>、流固耦合<sup>[24-25]</sup>、粘性流体模拟<sup>[26-27]</sup>等。由其发展而来的其他无网格包括 FPM (finite particle method)<sup>[28]</sup>和 MLPG (meshless local Petrov-Galerkin)<sup>[29]</sup>等。然而, 无论 FEM 还是无网格法, 依赖的理论基础均为连续介质力学, 在建模复杂系统动力学时主要采用偏微分方程来描述。其偏导数依赖物理场的连续性, 因而不

足以有效表征裂纹扩展等物理问题中的非连续边界。近场动力学(Peridynamics)<sup>[30]</sup>的提出有效弥补了其他方法在处理非连续性方面的短板,其核心思想是通过邻近作用力的积分来替代偏微分方程中的空间导数实现运动控制方程的建模与求解。而其不足之处在于近场动力学与连续介质力学的理论一致性尚未得到验证。早期近场动力学算法主要采用显式积分,稳定性和效率均较差。投影近场动力学引入隐式迭代计算,极大提升了仿真的效率和稳定性<sup>[31-32]</sup>。此外,由于物理仿真在对物理系统进行数学建模的过程中存在一定程度的假设与近似,导致仿真计算结果与实验验证存在偏差,一直以来是物理仿真面临的一大难点。近年来,随着深度学习等人工智能技术的发展,利用实验数据进一步提升物理仿真计算的可信度成为当下研究的热点<sup>[33-35]</sup>。

## 1 物理仿真关键问题

从上述物理仿真不同计算范式的发展历程中不难发现,每一种计算范式均有其自身优势和不足。因此,在解决实际问题时,需要结合问题特征、仿真需求、限制条件等多方面因素综合考虑来选择最佳的解决方案。评价物理仿真计算范式的优劣通常可以从如下几个方面进行考虑:

1) 仿真精度。值得注意的是针对特定问题并不意味着仿真精度越高越好。仿真精度的提升通常意味着计算代价的增加,因此可结合实际需求设定仿真精度的阈值;

2) 仿真效率。不仅受关注问题自身复杂度的影响,同时也与算法是否具备并行性、拟部署的硬件性能等密切相关;

3) 便捷性。指研发或使用特定物理仿真计算范式带来了成本;

4) 建模能力。指建模实际问题的适应性,以及当物理仿真计算范式在处理特定问题时因自身局限性导致建模能力不足时,是否具备可扩展性或兼容其他物理仿真计算范式的能力。

在当前的物理仿真技术领域,尚未存在一种能够同时满足高精度、高效率、高度便捷性以及强大建模能力的物理仿真计算范式。这一现状极大地限制了物理仿真在复杂系统分析、设计与优化等多领域应用的广度与深度。鉴于此,构建一套兼容多种物理仿真计算范式,且能依据具体需求动态重构与优化这些范式的通用物理仿真框架,成为破解上述难题的关键途径。考虑到物理仿真所涉及的问题类型多样,本文首先以边值问题为例阐述不同物理仿真计算范式的原理,通过阐明不同计算范式之间的共性特征及差异,为构建兼容不同物理仿真计算范式的引擎架构提供理论支撑。随后,本文将结合物理仿真计算范式的特点,详细阐述 FNMS 架构——即包含数据域(Data field)、节点(Node)、模块(Module)以及场景图(Scene graph)的通用物理仿真框架,并深入探讨如何解决多物理仿真计算范式之间的数据协同及机理耦合问题。最后还将进一步探讨当前引擎架构研发过程中所面临的难点与挑战,并为进一步完善物理仿真计算范式及引擎架构指明方向。

## 2 物理仿真计算范式

如图 1 所示,一个完整的物理仿真计算范式主要包括理论建模、数字建模、数值求解及可视分析 4 个部分。其中,理论建模主要通过数学工具对物理系统的几何形态及运动规律进行抽象和概括,建立相应的数学模型,以便更好地理解 and 预测物理系统的运动状态。数字建模则主要利用计算机来生成

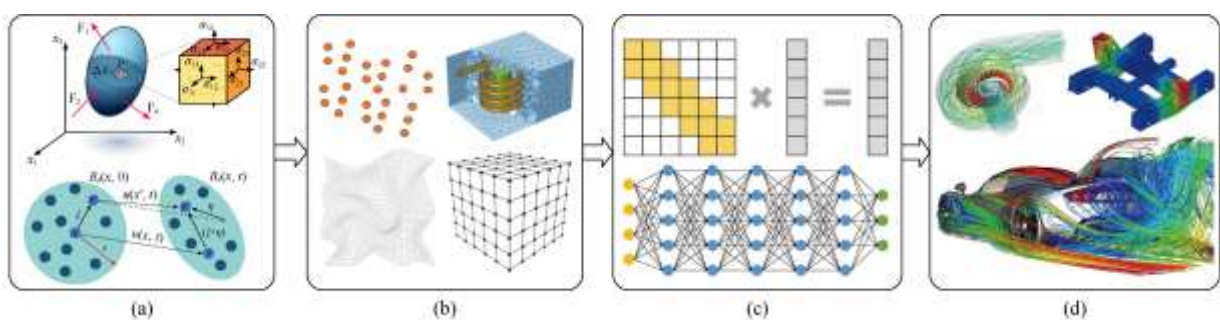


图 1 物理仿真计算范式流程图((a)理论建模; (b)数字建模; (c)数值求解; (d)可视分析)

Fig. 1 The computational paradigm of physical simulation mainly ((a) Theoretical modeling; (b) Digital modeling; (c) Numerical solving; (d) Visual analysis)

与理论模型对应的数字孪生模型。需要注意的是不同的物理仿真计算范式所依赖的数字模型不尽相同。如 FEM 需要将密度、温度、应力、应变等物理场映射到结构化网格上,而 SPH 则需要将物理场映射到离散粒子上进行求解。此外,因几何结构、物理属性等差异,同一种物理仿真计算范式也可能需要将理论模型映射到不同的数字模型上,如 FEM 可基于四面体或六面体单元来进行数值离散,进而通过利用数值计算方法对数字模型中的线性/非线性系统进行求解,其中典型数值计算方法包括直接法和迭代法 2 大类<sup>[36-37]</sup>。最后,在可视分析阶段,利用计算机图形学及可视化技术对物理仿真的结果进行可视化呈现,从而便于更加直观地对物理场进行分析。典型可视分析工具包括迹线、剖面、云图等。对照上述物理仿真计算范式构成,机理驱动与数据驱动的物理仿真计算范式的差异主要体现在理论建模阶段。其中机理驱动的物理仿真计算范式通常假定物理系统的力学机理及数学模型是已知的,如假定牛顿流体的运动规律可以采用纳维-斯托克斯方程进行描述,从而计算过程中通过选择特定数字建模工具和数值求解方法进行正向求解。相反,数据驱动的物理仿真计算范式假定物理系统的力学机理及数学模型不是完全已知的,如给定任意一种超弹性材料,其本构关系无法单纯地用某一类函数来刻画。因此,在理论建模阶段,数据驱动的方法可以借助神经网络<sup>[38]</sup>或一系列基底函数<sup>[39]</sup>并结合实验数据恢复出材料的真实本构关系。理论建模的差异性也进一步导致数字建模与数值求解上的差异,下文将结合边值问题具体阐述 2 类方法的特点。

### 2.1 边值问题描述

如图 2 所示,利用偏微分方程建模边值问题的一般形式可以表示为

$$\mathcal{L}u = f, \quad u \in \Omega \quad (1)$$

其中,  $\mathcal{L}$  为偏微分算子;  $u$  为偏微分方程关于空间坐标  $\mathbf{x}$  及时间  $t$  的解,且满足特定的边界条件,即

$$\begin{aligned} u(\mathbf{x}, t) &= g(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_D \\ u(\mathbf{x}, t) \cdot \mathbf{n}(\mathbf{x}, t) &= h(\mathbf{x}, t), \quad \mathbf{x} \in \Gamma_N \end{aligned} \quad (2)$$

其中,  $\mathbf{n}$  为朝外法向量;  $\Gamma_D$  为狄利克雷边界;  $\Gamma_N$  为纽曼边界;  $\Gamma_D$  和  $\Gamma_N$  满足  $\Gamma_N \cup \Gamma_D = \partial\Omega$  且  $\Gamma_N \cap \Gamma_D = \emptyset$ 。为简化讨论,后续公式将统一省略  $\mathbf{x}$  和  $t$ 。

在固体力学领域,固体材料的力学特性可以用应力  $\boldsymbol{\sigma}$  和应变  $\boldsymbol{\varepsilon}$  的关系来刻画,即材料本构关系<sup>[40]</sup>。

以线弹性材料为例,其本构关系可统一表示为

$$\boldsymbol{\sigma} = \mathbf{C} : \boldsymbol{\varepsilon} \quad (3)$$

其中,  $\boldsymbol{\sigma}$  和  $\boldsymbol{\varepsilon}$  分别为二阶柯西应力张量及柯西应变张量;  $\mathbf{C}$  为四阶弹性张量。根据线弹性材料的本构关系并结合连续介质力学理论,式(1)可进一步表示为如下运动控制方程

$$\rho \ddot{\mathbf{u}} - \nabla \cdot \boldsymbol{\sigma} = \mathbf{f} \quad (4)$$

其中,  $\rho$  为材料密度,计算应变张量式为

$$\boldsymbol{\varepsilon} = \frac{1}{2} [\nabla \mathbf{u} + (\nabla \mathbf{u})^T] \quad (5)$$

在流体力学领域,考虑到流体内部受力具有各向同性的特点,其二阶应力张量可以进一步拆解为  $\boldsymbol{\sigma} = -p\mathbf{I} + \boldsymbol{\sigma}'$ , 其中,  $p$  为流体内部压强;  $\mathbf{I}$  为二阶单位张量。各向异性张量  $\boldsymbol{\sigma}'$  通过应变率计算得到

$$\boldsymbol{\sigma}' = \frac{1}{2} \mu [\nabla \mathbf{v} + (\nabla \mathbf{v})^T] \quad (6)$$

其中,  $\mu$  为流体的粘性系数;  $\mathbf{v}$  为流体速度。结合上述特性,牛顿流体的运动控制方程可相应的表述成经典的纳维-斯托克斯方程为

$$\rho \dot{\mathbf{v}} = -\nabla p + \nabla \cdot \boldsymbol{\sigma}' + \mathbf{f} \quad (7)$$

基于上述理论建模,接下来将从机理驱动和数据驱动 2 方面进一步讨论如何对其进行数字建模。

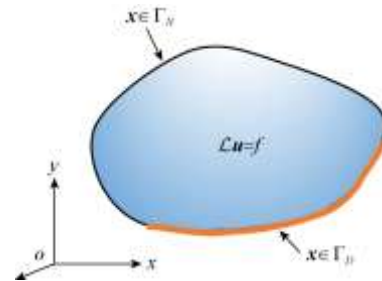


图 2 边值问题示意图

Fig. 2 Diagram of a boundary value problem

### 2.2 机理驱动的物理仿真计算范式

考虑到机理驱动的物理仿真计算范式多样性,本节主要通过选取 FDM, FEM 和 SPH 这 3 种最具代表性的经典物理仿真计算范式进行讨论,旨在阐明机理驱动的物理仿真计算范式的共性特征。在求解式(8)偏微分方程过程中,其目标是找到  $u$  在离散空间中的近似解  $\hat{u}$ , 并将其代入原偏微分方程时使得误差满足

$$\|\mathcal{L}\hat{u} - \mathcal{L}u\| = O(h^n) \quad (8)$$

其中,  $h$  为近似解  $u^h$  的空间特征尺度;  $n$  为近似解可拟合的阶数。从上式不难发现,求解偏微分方程当  $h$  趋近于 0 时,其近似解  $\hat{u}$  与精确解的误差同步



趋近于 0。结合计算域空间局部性, 假设任意偏微分的近似解可表示为具备紧致支撑形函数的积分形式, 即

$$\hat{u} = \sum_i \Phi_i \hat{u}_i \quad (9)$$

其中,  $\Phi_i$  为定义在积分点  $x_i$  上的形函数;  $\hat{u}_i$  为近似值。鉴于近似解  $\hat{u}$  无法精确表示偏微方程的解, 需进一步引入测试函数对产生的误差进行度量, 即要求近似解与精确解的残差与测试函数满足如下的正交性

$$\int \Psi (\mathcal{L}\hat{u} - f) d\Omega = 0 \quad (10)$$

通过对上述弱形式表示的偏微分方程离散并整理可得关于  $\hat{u}_i$  的线性方程组  $A_i \hat{u}_i = b_i$ , 从而进一步借助数值求解方法 (如雅可比迭代法, 共轭梯度法等) 求解偏微分方程的近似解。

根据上述经典物理仿真计算范式的基础理论, 图 3 给出了当选择不同的支撑域形态、形函数以及测试函数时, 可以分别得出 FDM 法、FEM 以及 SPH 法。

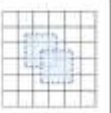


	有限差分法	有限元法	光滑力粒子动力学
支撑域			
形函数	$\Phi_i$	$\Phi_i$	$\Phi_i$
测试函数	$\Psi_i = \delta(x-x_i)$	$\Psi_i = \Phi_i$	$\Psi_i = \delta(x-x_i)$

图 3 经典物理仿真计算范式分类

Fig. 3 Classification of computing paradigms of physical simulation

从上述讨论不难发现, 机理驱动的经典物理仿真计算范式符合发现问题、提出假设到检验假设的规律。也就是说, 具体实现过程中会结合问题特征选取最合适的支撑域形态、形函数及测试函数, 并根据计算结果验证先验假设的准确性。譬如, 利用 SPH 中求解不可压缩流体时普遍选择 MÜLLER 等<sup>[41]</sup>采用的 Spiky 形函数计算粒子密度, 从而可实现较好的保体积特性。然而, 当预设的本构关系或形函数与实验数据产生偏差时, 机理驱动的经典物理仿真计算范式通常缺乏行之有效的自我调节手段。

### 2.3 数据驱动的物理仿真计算范式

如图 4 所示, 数据驱动的物理仿真计算范式一般流程可分为 3 个阶段<sup>[42]</sup>。

第一阶段, 利用全连接神经网络并以空间坐标  $x$  和时间坐标  $t$  作为输入前向传播来得到偏微分方

程的近似解, 即

$$\begin{aligned} z^0 &= (x, t) \\ z^k &= \alpha(\mathbf{W}^k z^{k-1} + \mathbf{b}^{k-1}), \quad 1 \leq k \leq L-1 \\ z^L &= \mathbf{W}^L z^{L-1} + \mathbf{b}^{L-1}, \quad k = L \end{aligned} \quad (11)$$

其中,  $L$  为隐藏层的层数;  $\mathbf{W}^k$  和  $\mathbf{b}^k$  分别为第  $k$  层网络的权重系数和偏置量;  $\alpha(\cdot)$  为非线性激活函数。

第二阶段, 利用全连接神经网络得到的近似解  $u \approx z^L$  进一步计算损失函数

$$\hat{L} = \omega_1 \hat{L}_{\text{PDE}} + \omega_2 \hat{L}_{\text{data}} + \omega_3 \hat{L}_{\text{IC}} + \omega_4 \hat{L}_{\text{BC}} \quad (12)$$

其中,  $\hat{L}_{\text{PDE}}$  为偏微分方程的残差  $\hat{L}_{\text{data}}$  用于度量数据偏差;  $\hat{L}_{\text{IC}}$  和  $\hat{L}_{\text{BC}}$  分别为度量与初值条件和边值条件的偏差;  $\omega_{1-4}$  为调节各项权重的系数。其中度量上述公式各项偏差最为普遍采用的方法为最小二乘法<sup>[43]</sup>。需要值得注意的是, 损失函数包含的项数会根据考虑问题的差异性有所不同, 譬如在准静态问题求解过程中并不需要考虑初值边界条件<sup>[38]</sup>。此外损失函数的计算依赖自动微分技术来完成<sup>[44]</sup>。最后阶段, 通过反向传播调整神经网络以及控制方程中的参数来最小化损失函数, 从而拟合出控制方程的解空间。

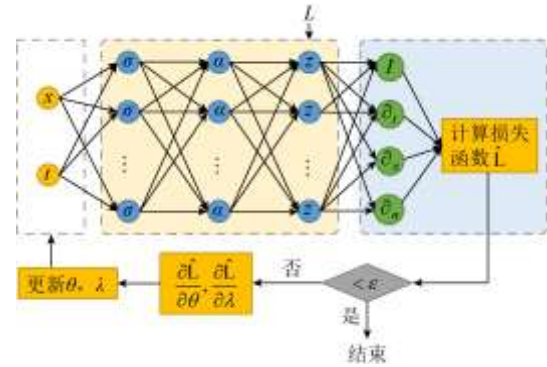


图 4 数据驱动的物理仿真计算范式流程图

Fig. 4 Flowchart of the data-driven physical simulation

### 2.4 对比分析

综上所述, 机理和数据驱动的物理仿真计算范式核心均在于如何精确、高效求解控制方程的近似解, 且理论上, 2 类方法均可通过细化网格或提升函数阶数来无限逼近精确解, 如 LU 等<sup>[45]</sup>指出可通过引入足够多的神经元从而实现仅利用包含单一隐藏层的神经网络拟合任意非线性连续函数。

对比 2 类方法的差异, 可分别从支撑域、形函数以及测试函数 3 方面来比较: ①支撑域形态, 机理驱动的方法通常需要根据问题特征及离散网格形态显式选取合适的支撑域形态, 且为保证计算的

高效性一般要求支撑域是紧致支撑的。数据驱动的方法则由于采用全连接神经网络，支撑域形态往往隐式包含于神经网络结构中；②形函数，机理驱动的方法每个采样点所采用的形函数是一致的，如 SPH 中普遍采用的形函数包含 Gaussian 函数、三次样条函数等<sup>[46]</sup>。数据驱动方法由于是从全连接神经网络学习得出，因此每个采样点最终的学习得到形函数不尽相同，且与采样点的分布密切相关；③测试函数，机理驱动的方法普遍采用基于克罗内克性质或变分原理的方法来度量误差。数据驱动的方法则可统一归结为基于克罗内克性质对损失函数进行度量。

基于上述讨论，可以看出基于神经网络的数据驱动方法核心在于泛化形函数或偏微分算子来实现运动控制方程的求解。针对机理明确问题的求解，经典机理驱动的方法往往表现出更高可靠性和高效性，数据驱动的方法则适用于内在机理不清晰，从而借助神经网络探寻更广义的解空间。然而需要注意的是，基于神经网络的数据驱动方法受数据质量影响较大，且在拟合控制方程解时可能存在过拟合的问题，如何提升模型可靠性、简化网络结构以最低代价实现高效计算值得进一步深入研究和探讨。

### 3 兼容多种物理仿真计算范式的通用引擎架构

根据上述讨论，物理仿真的核心关注点在于理论建模和数值求解。然而，由于物理仿真计算范式的多样性给软件架构的设计带来了诸多挑战，具体体现在如下几个方面：

1) 拓扑结构多样性。各种物理仿真计算范式均紧密依赖于特定的拓扑结构。例如，FDM 法能够灵活适应于均匀网格或动态调整的自适应网格结构；FEM 则通常构建在四面体或六面体网格之上，以精准捕捉复杂几何形状；而 SPH 则采用离散化的粒子集合作为其核心表示形式；至于数据驱动的方法，尤其是深度学习领域，则普遍倾向于利用前沿技术神经网络，以高效处理并解析大规模数据。这些多样化的拓扑结构设计，不仅体现了不同计算范式在方法论上的深刻差异，也彰显了其在解决特定物理仿真问题时的独特优势与灵活性。

2) 数据结构与数值方法可复用性。尽管各种计算范式在数值离散化、积分策略及后处理等多个维

度上展现出显著的差异性，但其之间亦蕴含诸多共通的核心算法元素，如基础数据结构的构建与利用、积分方法的选择与优化等。因此，在充分尊重并发挥各计算范式独特优势的同时，探索并实践提高数据结构与数值方法的可复用性策略，对于减轻软件开发过程中的重复劳动以及提升整体开发效率而言，具有举足轻重的意义。

3) 可扩展性。通常而言，物理仿真引擎的核心在于不断迭代与优化计算范式，旨在追求功能的全面性、精度的极致性以及运行效率的最优化。为了实现这一目标，要求每个引擎组件均具备高度的独立性与可扩展性。这意味着，在引入新功能时，能够以最小的干预对现有数据结构或算法进行调整，确保系统的灵活性与可维护性。这一特性极大地便利了多人员、跨学科的团队协作，共同推动物理仿真技术向更高水平迈进。

4) 多物理场耦合及计算范式协同。考虑单一物理仿真计算范式在建模复杂物理问题上局限性，需要软件架构具备支撑多种物理仿真计算范式协作解决复杂物理系统的能力，具体包括不同计算范式之间的数据映射、时间步长控制、单/双向耦合等。

5) 计算设备多样性。在当今计算技术的多元化格局中，主流计算设备涵盖了从单核/多核 CPU 到 GPU，乃至超大规模计算集群的广泛范畴。物理仿真引擎的架构设计面临着—项核心挑战：如何在有效屏蔽复杂硬件细节的同时，确保算法实现的便捷性与计算的高效性。这要求在架构规划中，既要构建—层抽象层来简化硬件差异对开发者的影响，又要优化内部机制，以最大化利用各类计算资源的潜力，实现算法的高效执行。

当前物理仿真主流软件采用的引擎架构可笼统的归结为：服务导向的架构(Service-oriented architecture)及基于远程调用协议(Remote procedure call) 2 类<sup>[47]</sup>。服务导向的架构对各个代码组件不施加任何约束，其组件间的通信主要依赖于 I/O 数据来实现。即在服务导向的架构下，耦合不同代码组件时无需对原有代码进行任何修改，从而大幅度减少了代码重构的工作量。然而，其不足之处在于，基于 I/O 数据传递的软件架构性能难以得到保证，特定模块的性能问题可能会影响到整个软件的运行效率。相比之下，基于远程调用协议的架构则需要对原有代码进行一定程度的抽象和重构，以实现不同组件之间更为高效的信息共享。这种信息共享不仅涵盖了各种物理场数据，还包括了具备特定功

能的 API。为了实现代码组件之间的高效信息共享, 基于远程调用协议的架构通常需要结合具体应用场景内各功能组件所包含的各种数据及功能, 建立一个中间层。其典型的实现的引擎架构为 SOFA Framework<sup>[48]</sup>, 并通过建立层次化抽象将不同算法组件高效耦合从而实现虚拟手术的实时仿真。

### 3.1 FNMS 物理仿真引擎架构

为兼容上述 2 种工作模式, 本节以流固耦合为例阐述 FNMS 架构。FNMS 架构的核心构成要素包

括数据域(Data field)、节点(Node)、模块(Module)以及场景图(Scene graph)4 部分, 如图 5 所示。FNMS 架构为各种物理仿真计算范式提供了统一的抽象表示。在算法实现层面, 允许研发人员仅在模块层面进行操作, 既可以选择单纯依赖数据传递来调用原有算法, 从而降低工程研发难度, 也可以对原有算法进行深度重构, 以确保高效性。这样做能够同时兼顾服务导向和基于远程调用协议这 2 种模式下的物理仿真计算。

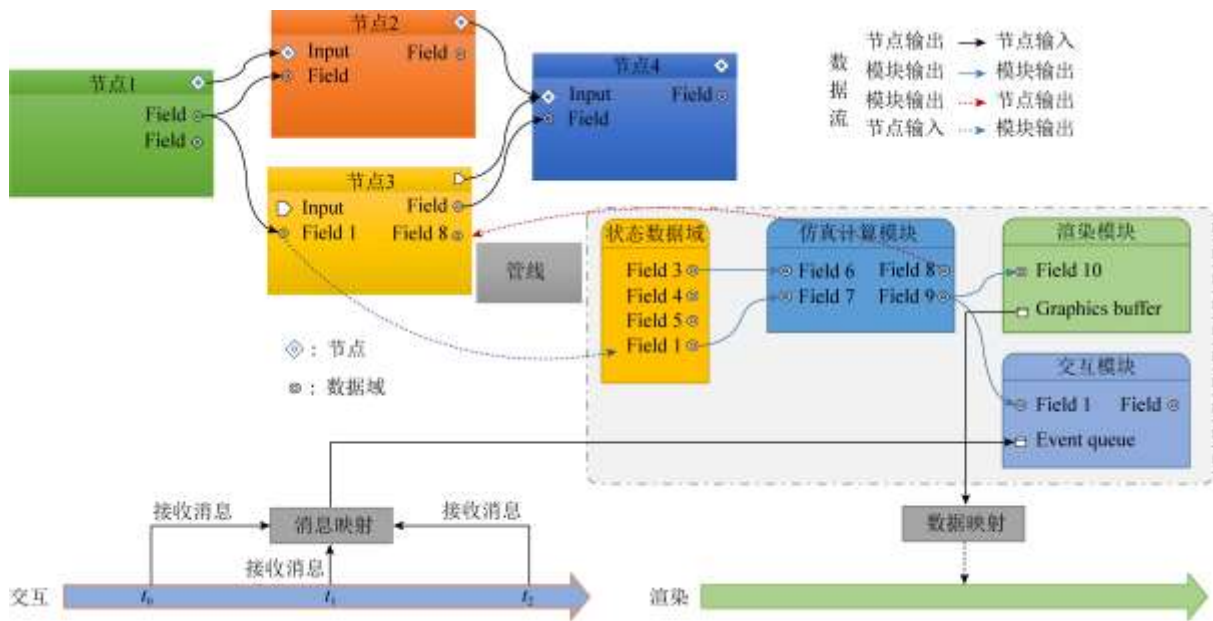


图 5 FNMS 架构示意图

Fig. 5 Diagram of the FNMS architecture

#### 3.1.1 数据域

数据域包含特定结构和功能的数据。数据域结构: 以描述流体运动的速度场为例, 其可以定义在一维、二维或三维网格上, 也可以定义于离散的粒子或结构化网格。为简化数据域在 C++ 中的定义和使用, 可从 FBase 类派生出如图 6 所示的多种数据域, 其中包括用于定义基本变量与参数的 FVar、一维网格向量场 FArray、二维网格向量场 FArray2D、三维网格向量场 FArray3D 以及邻接列表 FArrayList 等。此外, 为兼容复杂数据结构, 可进一步引入 FInstance 用于模块或节点之间的复杂对象传递。为保证数据域使用的便捷性, 所有数据域均通过抽象设计屏蔽了 CPU/GPU 底层实现上的差异, 并提供统一的访问接口。

根据功能的不同, 数据域可进一步划分为 4 大类: 控制参数域、状态数据域、输入数据域及输出数据域, 在基于 PeriDyno<sup>[49]</sup>实现的 C++ 代码中分别

对应 DEF\_VAR, DEF\_{\*}\_STATE, DEF\_{\*}\_IN 和 DEF\_{\*}\_OUT 4 个宏定义用于相应数据域的快速声明。其中, 控制参数特指那些经过 FVar 特化处理的类成员变量, 其扮演着调节特定节点或模块关键参数的重要角色, 如流体不可压缩求解器中的迭代步数。状态数据域则用描绘特定对象在某一时刻的物理状态, 为系统提供即时的状态反馈。而输

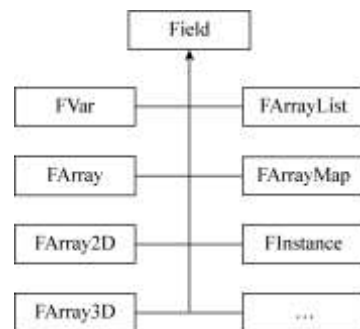


图 6 数据域示意图

Fig. 6 Diagram of the data domain



入与输出数据域，顾名思义，是节点或模块间数据流通的桥梁，确保信息的有效传递与交换。值得一提的是，无论是被定义为输入、输出还是状态的数据域，均支持通过 `connect()` 与 `disconnect()` 函数实现灵活的数据域链接与断开操作。这一过程的核心在于仅传递数据的指针而非数据本身，该设计不仅显著提升了数据访问的效率，还有效减少了存储资源的占用，是优化系统性能与资源利用的关键策略。

### 3.1.2 模块

模块作为构建物理仿真计算范式的基石，是构成物理仿真机理的最小独立算法单元。并由控制参数域、输入数据域、输出数据域，及核心的算法主体的关键部分组成。针对具体的物理仿真问题，合理设定模块的颗粒度是提升仿真方法复用性与操作便捷性的关键所在。过细的颗粒度虽能细化功能，但也能引发算法模块数量激增，进而拖累模块间数据交换的效率。反之，若颗粒度设置过于宽泛，则会削弱模块的通用性和复用潜力，导致代码冗余上升，增加后期维护的复杂性与成本。尤为重要的是，在模块设计时应明确排除状态数据域，此举旨在消除物理仿真动态过程中积累的历史数据对模块独立功能的潜在干扰，确保模块在给定相同参数与输入时，能够产生一致且可预测的运算结果。这种设计不仅强化了模块功能的明确性与稳定性，还极大地促进了不同物理仿真计算范式间算法模块的互操作性与可复用性，为物理仿真的高效开发与灵活应用奠定了坚实基础。

### 3.1.3 节点

节点是将数据域和模块有机组合形成的整体。其主要构成包括：控制参数域、输入数据域、输出数据域、状态数据域以及特定算法或模块集合。相比于模块设计，其增加的状态数据域主要用于定义被仿真对象在当前时刻的物理状态。以基于 SPH 实现的不可压缩流体仿真为例，其当前时刻的状态主要包括粒子位置、粒子速度以及外力等。也就是说，节点的状态数据域是用于刻画物理系统当前运动状态的最小数据集合。结合特定功能模块，可认为节点是特定物理仿真计算范式的数字化身。其通过数据与算法模块的解耦一方面可以实现算法在不同物理仿真计算范式之间的复用，同时便于实现多物理场耦合过程的交换与共享。

### 3.1.4 场景图

场景图在图形引擎中被广泛采纳用于构建三维场景。通过树状或图结构布局，对场景中的物体、

灯光、摄像机等元素进行了层次分明的组织与管理，极大地提升了场景管理的效率与灵活性。在这一领域内，场景图的典型应用包括 `OpenSceneGraph`<sup>[50]</sup> 与 `Unity`<sup>[51]</sup> 等开源图形引擎。与图形引擎聚焦于静态物体的精细管理不同，物理仿真引擎的核心在于动态时变对象的状态与行为模拟，其中各个物理对象之间错综复杂的相互作用力或耦合关系构成了其研究的重点。为了精准建模这些物理对象间的相互影响，FNMS 架构采用基于节点的有向无环图来重构物理仿真场景。其功能不仅包括节点的动态创建、删除、遍历、查询等基础功能，也包括仿真单步执行与调试、外设交互与响应、辅助信息显示等。这种设计不仅保留了节点对象管理的灵活性，还通过高效的数据传递机制，实现了不同物理对象间无缝的耦合与交互，为物理仿真的灵活性奠定坚实的基础。

## 3.2 FNMS 架构运行原理

本节以图 7 所示的流固耦合为例进一步阐述 FNMS 架构的运行原理。

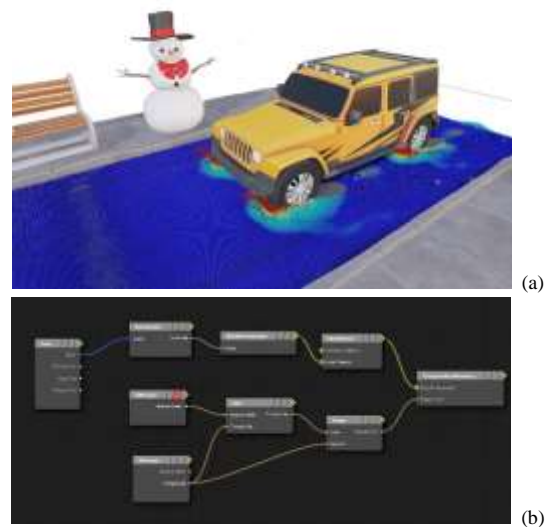


图 7 流固耦合场景((a)效果图; (b)节点图)

Fig. 7 Solid-fluid coupling scenario ((a) Visual effect; (b) Node graph)

### 3.2.1 网格拓扑层次化表示及结构映射

网格拓扑是物理仿真计算的核心基础，每种计算范式都需要将物理场定义在特定的网格结构上，并依据网格拓扑结构对偏微分算子或相关操作进行离散化处理。在物理仿真计算的整个流程中，网格拓扑的作用不局限于偏微分方程的数值离散和力学建模，还广泛应用于碰撞检测、物理场可视化等领域。换言之，网格拓扑是物理仿真计算范式中连接理论建模、数值计算与可视分析的



桥梁。为确保网格拓扑的灵活性和可复用性, FNMS 架构借鉴了 SOFA 框架<sup>[48]</sup>的理念, 将涵盖典型物理仿真计算范式所依赖的网格拓扑通过结构抽象, 图 8 为组织成的层次化表示。该表示方法有助于简化数据接口, 提高算法模块的可复用性。

以碰撞检测为例, 若碰撞检测算法模块已针对三角网格(TriangleSet)实现了自碰撞功能, 则该算法模块无需修改任何代码, 即可为 TriangleSet 的子类(如四面体网格 TetrahedronSet)提供同样功能的碰撞检测算法。

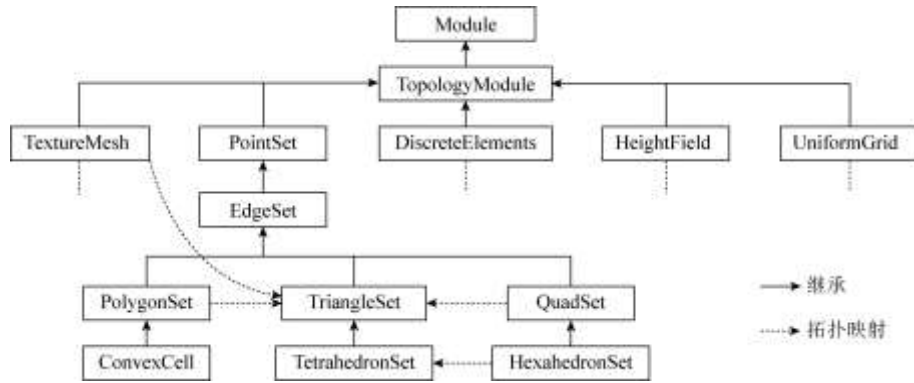


图 8 网络拓扑层次化结构表示  
Fig. 8 Hierarchical representation of mesh topology

此外, 针对不具备继承关系的拓扑网格, FNMS 架构进一步引入拓扑映射模块实现不同网格拓扑之间的数据映射。如图 8 所示, TextureMesh 可以通过拓扑映射转换为 TriangleSet, 从而极大的方便不同计算模块的耦合。

3.2.2 物理对象的多模态表示

物理对象的多模态表示指的是同一个物理对象为了适应不同的物理仿真计算范式, 而具备多种不同的数字形态。以流固耦合中的吉普车为例, 其包含了 3 种不同的模态表示(图 9)。最左边的 TextureMesh 包含了定义三角网格相关的顶点坐标、顶点法线、纹理坐标、三角网格索引以及材质信息, 主要服务于渲染引擎, 以实现高真实感的绘制。最右边的 DiscreteElements 则包含了定义车辆部件的刚体、关节等, 主要用于刚体动力学求解以及与地形的碰撞检测。中间的 TriangleSet 则主要用于 SPH, 实现车辆与流体粒子的精细碰撞检测与耦合计算。通过针对物理对象选择合适的多模态表示, 并结合网格拓扑结构映射, 可以增强算法模块的可复用性, 实现不同物理仿真计算范式之间的高效耦合。

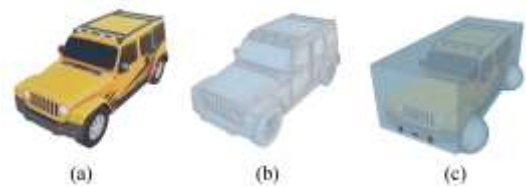


图 9 吉普车多模态表示  
Fig. 9 Multimodal representation of a jeep  
(a) TextureMesh; (b) TriangleSet; (c) DiscreteElements)

3.2.3 管线

当节点内包含多个相互依赖的功能模块时, 管线扮演着模块管理者的角色。可根据模块间的依赖关系, 构建一个模块执行队列, 以确保数据的正确传递。如图 10 所示, 基于 SPH 实现的不可压缩流体包含 4 个模块: NeighborPointQuery, ParticleIntegrator, IterativeDensitySolver 及 ImplicitViscosity。其中,

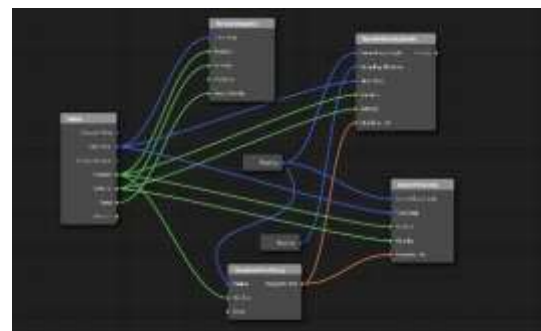


图 10 吉普车刚体动力学仿真管线  
Fig. 10 Simulation pipeline of rigid body dynamics for a jeep

NeighborPointQuery 负责并行查找粒子的邻域列表; ParticleIntegrator 实现显式欧拉积分; ImplicitViscosity 基于 XSPH<sup>[52]</sup>实现隐式粘性计算; IterativeDensitySolver 则基于 PBF<sup>[53]</sup>实现流体的不可压缩性。当 2 个算法模块之间存在数据依赖时, 管线会依据数据域的链接关系, 将所有算法模块构建成一个有向无环图, 并利用深度优先的遍历算法来构建模块执行队列, 从而确保模块能够按照正确的顺序执行。管线的引

入有效地解决了物理仿真计算范式的可扩展性问题。如 FNMS 架构通过替换不可压缩流体仿真管线中的模块，可实现更高效的邻域查找或不可压缩性求解算法。

### 3.2.4 数据域同步

为了保证仿真计算的高效性，FNMS 架构确保在输入或控制参数未发生变化时，节点或模块尽可能不重复执行计算。为了实现这一功能，FNMS 架构为每个数据域定义了 2 个类型为 TimeStamp 的时间戳：其中 TickTime 用于记录数据发生更改的时间，而 TackTime 用于记录数据被其父类模块或节点使用的时间。如图 11 所示，当输入数据域 A 发生更改时，FNMS 架构通过调用 tick()函数来更新数据域 A 的 TickTime，从而记录输入发生更改的时间。当数据域 B 对应的模块调用 update()函数执行更新操作时，首先会通过检查数据域 B 中的 TackTime 来判断输入数据域 B 是否已被使用。如果 B 的 TackTime 小于 A 的 TickTime，表明输入数据已经发生了改变，因此模块需要执行算法进行更新。在模块更新结束时，会调用 tack()函数来同步更新数据域 B 的 TackTime。基于 tick-tack 的数据同步机制，可以有效避免冗余计算，从而提升物理仿真和数据交换的效率。

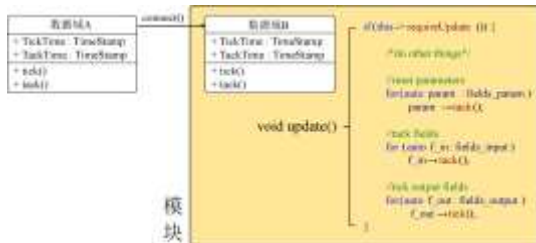


图 11 数据域同步机制示意图

Fig. 11 Diagram of the data field synchronization mechanism

### 3.2.5 仿真、渲染与交互的异步协同

为了优化仿真计算、渲染及交互的统一建模与管理，FNMS 架构设计了模块化的封装策略，将渲染算法与交互算法整合为独立模块，并通过第 3.2.3 节管线实现集中而高效地管理。在单线程模式下，该管线机制确保了各模块依据数据依赖顺序执行，维护了流程的连贯性。然而，面对多线程环境下的挑战，即仿真计算、渲染与交互并行运行时，FNMS 架构采取了异步协同的解决方案来应对潜在的数据冲突问题。具体而言，是为每个渲染与交互模块分配了专属的互斥数据区域，并通过定制化的 API

接口，实现了仿真计算与渲染、交互之间的无缝且安全的互斥访问机制。以鼠标输入与响应流程为例，FNMS 架构通过图 12 直观展示了该机制的运作原理。在该流程中，enqueueEvent()函数负责接收鼠标事件，并尝试将其加入消息队列以待处理。关键的是，在执行插入操作前，该函数会先检查消息队列是否已被仿真计算线程锁定，从而避免数据冲突。只有当队列处于未锁定状态时，鼠标事件才会被安全地加入到队列中。类似地，当仿真计算需要响应鼠标输入事件时，可通过 onEvent()接口进行操作。但在此之前，该接口同样会检查消息队列的锁定状态，确保没有其他线程(如鼠标输入线程)正在访问队列，以此保障操作的互斥性和数据的完整性。通过设立互斥数据区并应用严格的互斥锁策略，FNMS 架构不仅成功解决了仿真计算、渲染与交互模块在统一管理下的异步协同问题，还极大地增强了物理仿真计算的交互性与系统的可扩展性，为复杂场景的实时仿真与交互提供了坚实的技术支撑。

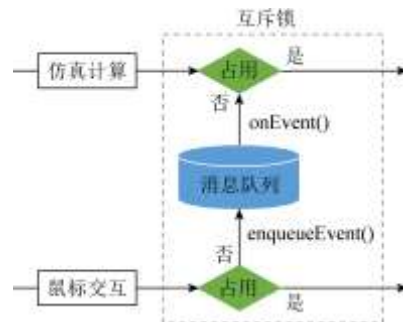


图 12 仿真与交互异步协同示意图

Fig. 12 Diagram of asynchronous collaboration between simulation and interaction

## 4 应用案例

### 4.1 有限差分法

FDM 方法是物理仿真中的经典算法之一，该方法将流体、固体等所处的计算域离散为结构化网格，通过对方程中的微分项进行差分近似，将偏微分方程求解转化为线性方程组求解，从而得到目标的速度、压强等物理属性。基于本文提出的 FNMS 引擎架构，结合血液动力学问题的特征，本文选用 FDM 方法进行血流仿真。图 13 为采用扩散边界法<sup>[1]</sup>仿真血流以进行门脉高压的无创诊断的示意图(上)以及场景图(下)。血液动力学仿真首先需要根据输入的血管模型生成计算网格，包括均匀网格和自适应网格。图 13(a1)展示了根据血管模型构建的自

适应网格的二维截面图。不同的网格构建算法封装为不同的模块, 在节点内部进行调用。血流仿真的核心是基于构建的网格拓扑进行求解的节点, 该节点调用多个模块包括粘性求解模块、压力求解模块等。图 13(a2)展示采用扩散边界得到的速度场。血管和血流流速可视化的节点分别调用了点和边 2 种基础拓扑的可视化模块, 通过封装这些节点可以复用并服务于多个样例。

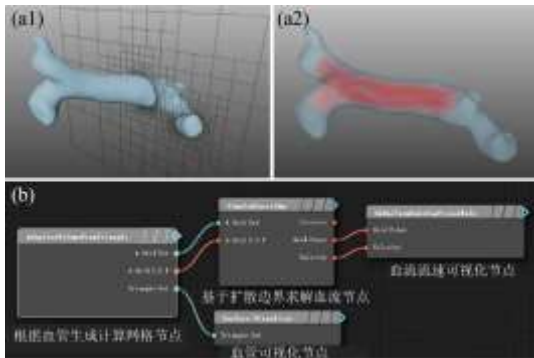


图 13 基于扩散边界的血流仿真场景((a1)自适应网格二维截面图; (a2)速度场; (b)节点图)  
 Fig. 13 Diffuse interface model for blood flow simulation ((a1) A two-dimensional section of the adaptive grid; (a2) Velocity field; (b) Node graph)

#### 4.2 有限元法

FEM 是求解工程和数学建模中出现的微分方程的常用方法。典型的问题领域包括结构分析、传热、流体流动和电磁等传统领域。该方法将复杂的连续体结构分割成有限数量的简单子区域(有限元), 并对其进行离散化建模, 然后利用数值计算技术求解这些子区域的方程, 最终得到整个结构的近似解。基于本文论述的引擎架构, 易于实现基于 FEM 的全流程结构仿真。图 14(a)和(b)分别为基于 FEM 的泵体刚强度仿真分析场景及场景图。首先创建几何节点, 进行几何模型导入和显示, 图中模型为 .step 格式文件。然后创建网格剖分节点, 将几何节点所产生的数据接入网格节点, 打开网格节点属性编辑器, 通过设置网格阶次、网格继承、网格类型、网格尺寸、网格剖分算法、网格清理等参数, 进行网格生成。网格剖分完成后, 进入前处理节点, 该节点可以进行单元类型、截面属性、材料类型和参数、边界条件类型和参数、载荷类型和参数、接触关系、连接关系等参数的设置。接下来基于引擎内部标准模型格式, 生成描述仿真场景的数据文件, 利用 FEM 求解节点进行求解计算。并将求解数据接入后处理节点进行云图渲染, 得到计算结果云图。

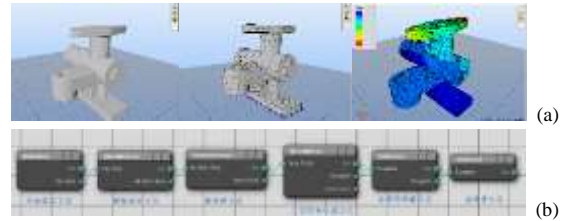


图 14 基于有限元法的泵体刚强度仿真分析((a)效果图; (b)节点图)  
 Fig. 14 FEM-based simulation and analysis for pump ((a) Visual effect; (b) Node graph)

#### 4.3 光滑粒子动力学

SPH 方法是实现流体、流固耦合仿真的重要方法之一, 该方法将流体、固体等材料视为由离散无网格粒子组成的系统, 基于 SPH 核函数以及各粒子的邻域粒子, 实现速度、压强等物理场以及微分场的离散化, 进而完成粒子运动计算与流体模拟。图 15(a)和(b)分别为基于对偶粒子法<sup>[54]</sup>实现的流体动仿真场景以及场景图。基于对偶粒子法实现的流体仿真需要包含粒子发射器、粒子采样等流体粒子的生成节点, 并将其生成的粒子数据输入到基于粒子法实现的仿真节点中。SPH 仿真节点中包含渲染管线与仿真管线, 渲染管线通过互斥锁策略获取仿真计算的结果数据, 并完成粒子的实时渲染; 仿真管线则用于实现完整的 SPH 仿真计算流程, 流程中的各主要计算环节均为独立计算模块, 并置于节点内部, 具体包含邻域粒子查找模块、压强计算模块、用于辅助计算的虚粒子生成模块、粘性计算模块等。各计算模块之间的数据传递基于上述数据域方法实现。此外流体仿真过程中需要包含固体边界, 并与流体发生碰撞作用, 此类边界可能为隐式化的有向距离场、固体几何边界面片等, 由于几何拓扑结构与实现方式相差较大, 需要针对性地设计不同边界计算节点, 串接在粒子法动力学计算节点之后施加碰撞作用。

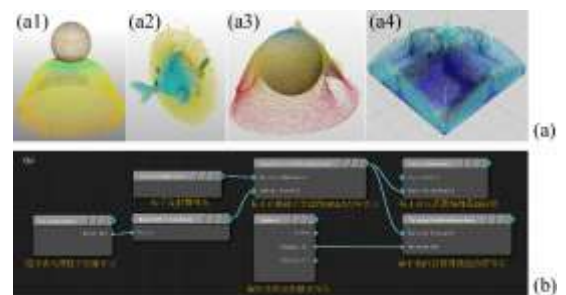


图 15 基于对偶粒子的流体仿真((a)效果图; (b)节点图)  
 Fig. 15 Fluid simulation based on a dual particle approach ((a) Visual effect; (b) Node graph)



#### 4.4 数据驱动的物理仿真计算范式

数据驱动的物理仿真方法是一种融合实验/仿真数据收集与分析并依托机器学习模型进行训练的方法,旨在精准捕捉并预测复杂多变的物理现象。该方法巧妙融合了传统的物理仿真方法与先进的数据科学理念,实现了在提高仿真精确度的同时,显著缩短计算周期,并展现出对难以通过传统理论模型全面描述的复杂系统的高度适应性。本文探讨的引擎架构已成功实现数据驱动技术在处理静态、准静态及动态问题上的应用与整合。

针对静态问题,基于图像数据驱动的冲压反求方法<sup>[55]</sup>将图像作为设计参数和目标函数之间的中间变量,所构建的流形空间不仅适用于正向预测目标函数,也适用于反问题的求解。此外,基于数据驱动可加速仿真的特性,由一个骨干神经网络和一个系数神经网络组成的基于物理信息神经网络的拓扑优化(DCPINN-TO)方法<sup>[56]</sup>通过动态配置训练参数,大幅降低了大规模静态问题的计算成本。

面向复杂的动态问题,如涉及材料、几何及接触非线性的时空演变过程,在碰撞数值模拟领域内,数据样本的生成往往伴随着高昂的成本。鉴于动态过程所固有的高度非线性特性,数据驱动方法在这些应用场景中面临了显著的挑战。为了有效应对这些难题,一种基于随机依赖神经估计器(stochastic dependency neural estimator, SDNE)的方法通过从输入参数域到高维时空响应的映射<sup>[57]</sup>,并引入了图神经网络(graph neural networks, GNN)和

时序卷积神经网络(temporal convolutional neural networks, TCN),旨在通过构建与动态过程随机性紧密相关的神经网络模型,以更精准地捕捉并预测动态系统中的复杂行为,从而在不显著增加数据成本的前提下,提升数值模拟的精度与效率。该网络基本架构如图 16 所示。为了验证其有效性,该算法分别解决了二维成形问题(图 17)和三维碰撞问题(图 18)计算结果与传统数值算法和实验数据均能良好吻合。

## 5 总结

物理仿真作为工业软件的基石,是一门高度综合性的交叉学科,深度融合了数学、物理学与信息技术等多学科的理论与方法,并紧密关联理工医等多个应用领域的实际需求。持续推动物理仿真计算范式的创新与发展,旨在探索科学新知、解决复杂工程及科学问题。本文深入探讨了物理仿真计算范式的共性特征,旨在提出一种更为高效、灵活的解决方案。为此,本文设计了 FNMS 架构体系,以兼容并整合多种物理仿真计算范式,通过机理与数据的深度融合,实现计算范式的无缝衔接。本文架构的核心创新在于其 4 层结构的设计:数据域层、节点层、模块层与场景图层,从根本上解决了物理仿真算法在可复用性、可扩展性上面临的挑战,以及仿真计算与渲染、交互的高效异步协同。同时,该架构设计有效降低了多计算范式耦合以及异构计算单元(如 CPU 与 GPU)集成过程中产生的复杂性问题。

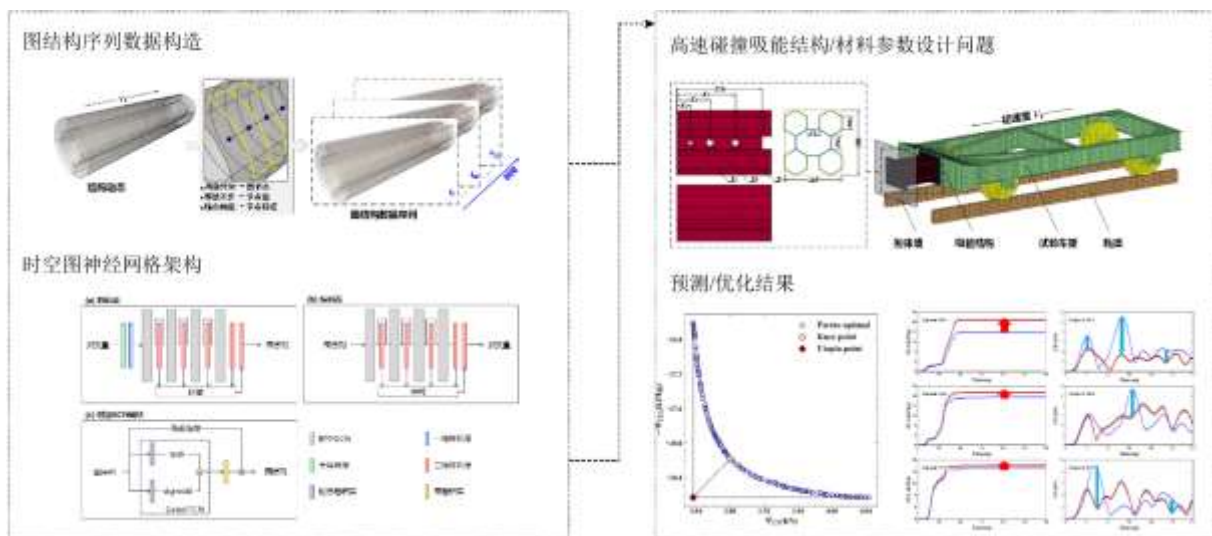


图 16 基于图神经网络的瞬态碰撞问题求解

Fig. 16 Solving transient collision problems based on graph neural networks



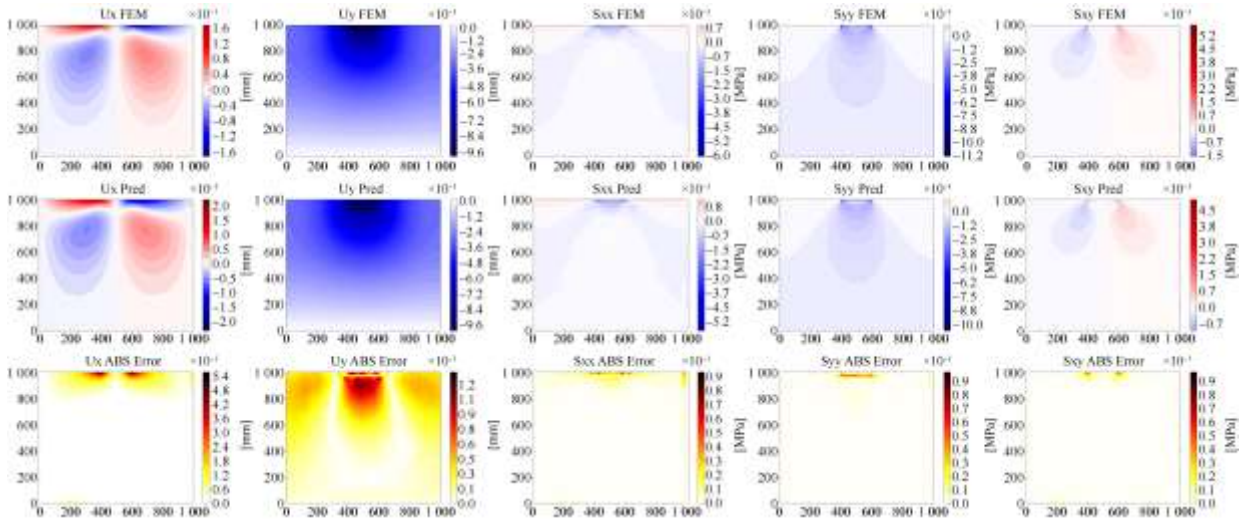


图 17 面向二维动态问题的智能求解

Fig. 17 Intelligent solving for two-dimensional dynamic problems

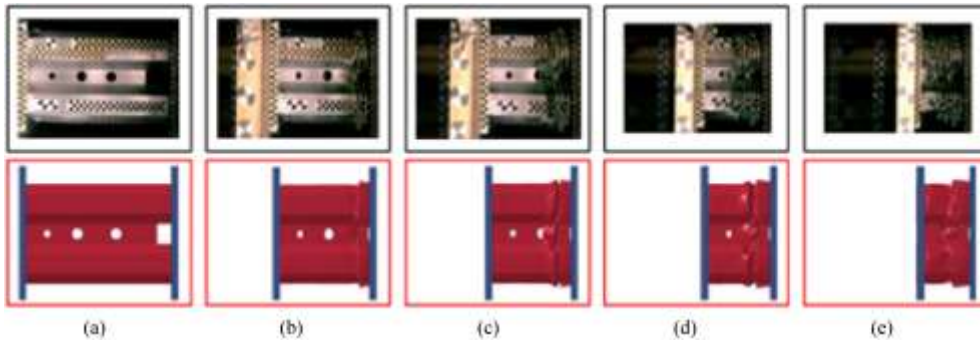


图 18 面向三维碰撞问题的智能快速仿真结果与实验对比

Fig. 18 Comparison of intelligent rapid simulation results and experiments for three-dimensional collision  
(a) 0.00 ms; (b) 4.38 ms; (c) 7.29 ms; (d) 11.56 ms; (e) 18.00 ms

对于物理仿真未来的发展趋势, 本文认为可以从如下几个角度进行考虑:

1) 更加注重实时性、准确性和高效性。随着 GPU 等并行计算设备的普及, 许多传统基于 CPU 实现的数值算法变得不再适用。如非线性问题求解普遍采用的牛顿法在移植到 GPU 时, 因海森矩阵计算的复杂性导致牛顿法无法充分发挥 GPU 的并行计算优势。为兼容 GPU 硬件架构, 一种典型的方案是采用无矩阵的迭代算法, 如半隐连续迭代方法<sup>[32]</sup>。因此, 需要充分发掘数值方法中的并行特征, 结合新型并行计算体系实现更为高效、高精度的仿真计算方法。

2) 与人工智能、大数据等技术的深度融合。机理与数据是物理仿真计算范式的 2 大核心要素, 其相辅相成, 共同构筑了仿真的坚实基础。经典的物理仿真方法以其深厚的理论根基著称, 而数据驱动方法则以其对海量信息的敏锐捕捉能力见长。因此, 如何将数据融入物理仿真的真实机理模型

中, 实现两者的深度融合, 以及如何通过泛化机理模型来洞察新问题、阐释新原理, 成为了推动物理仿真与人工智能、大数据等前沿技术深度融合的关键路径。这一融合不仅将促进物理仿真技术的飞跃式发展, 还将为科学探索和技术创新开辟更为广阔的空间。

3) 非连续、非线性复杂物理系统机理建模。复杂物理系统中存在的非线性及非连续一直以来都是数学、物理及计算机等领域研究的难点。其主要表现为依托连续介质力学理论发展而来的传统数值建模方法在建模系统动力学时主要采用偏微分方程来表述, 无法有效处理系统中的非连续特征。依托非局部作用理论建立的近场动力学可以有效缓解非线性、非连续带来的复杂系统建模难题, 然而近场动力学的精度及仿真一致性问题依然需要结合试验数据进行充分验证。

4) 多物理场耦合。多物理场耦合问题因涉及要素的多样性给物理仿真计算带了诸多挑战, 包括因

时空尺度差异带来的耦合计算难题,因几何形态差异带来的建模与表示难题等。目前解决多物理场耦合难题策略主要包括 2 类,第一类通过采用统一的物理仿真计算范式实现,如 MPM<sup>[58]</sup>。该方法尽管实现简单,但可扩展性和兼容性相对较差,不利于跨领域协作;第二类方法则通过耦合多种物理仿真计算范式来实现,可更好实现代码的复用。FNMS 架构尽管在架构层面保证了对上述 2 类策略的兼容性,不足在于处理强多物理场耦合问题时依然无法提供简洁有效的统一解决方案保证多物理场耦合过程高效性和高可复用性,需依赖研发人员结合特定问题分析处理。

5) 物理仿真与大模型的融合。随着 Sora 等前沿大模型的横空出世,其卓越的视频生成技术深刻触动了人们对构建全面世界模型的广泛探讨<sup>[59]</sup>。诚然,这些模型以其惊人的以假乱真能力令人瞩目,但其局限性亦不容忽视:单纯依赖海量数据的堆砌,尚不足以捕捉并再现视频画面背后复杂的物理规律,进而在某些特定场景中不可避免地出现了物理失真现象。更为关键的是,物理规律的缺失严重制约了这些模型的泛化能力,使其难以跨越至如自动驾驶等需要高度实时交互与精确力学反馈的应用领域。因此,探索如何将多元化的物理仿真模型与先进的计算范式无缝融入数据驱动的大模型之中,自底层开始逐步构建并揭示世界运行的内在物理逻辑,已成为通往构建既具解释性、又可实现高度交互与广泛扩展的世界模型的必由之路。这一路径的开辟,将为人工智能技术的未来发展开辟新的篇章。

### 参考文献 (References)

- [1] REN L X, WAN S, WEI Y, et al. Towards a non-invasive diagnosis of portal hypertension based on an Eulerian CFD model with diffuse boundary conditions[C]//The 24th International Conference on Medical Image Computing and Computer Assisted Intervention. Cham: Springer, 2021: 107-116.
- [2] KROLL N, GAUGER N R, BREZILLON J, et al. Flow simulation and shape optimization for aircraft design[J]. Journal of Computational and Applied Mathematics, 2007, 203(2): 397-411.
- [3] Geyer M. BMW group starts global rollout of NVIDIA omniverse[EB/OL]. (2023-03-21). [2024-05-31]. <https://blogs.nvidia.com/blog/bmw-group-nvidia-omniverse/>.
- [4] SZLUFARSKA I, RAMESH K T, WARNER D H. Simulating mechanical behavior of ceramics under extreme conditions[J]. Annual Review of Materials Research, 2013, 43: 131-156.
- [5] MARLOW F, JACOB J, SAGAUT P. A multidisciplinary model coupling Lattice-Boltzmann-based CFD and a social force model for the simulation of pollutant dispersion in evacuation situations[J]. Building and Environment, 2021, 205: 108212.
- [6] THOMÉE V. From finite differences to finite elements: a short history of numerical analysis of partial differential equations[J]. Journal of Computational and Applied Mathematics, 2001, 128(1/2): 1-54.
- [7] COURANT R, FRIEDRICHS K, LEWY H. Über die partiellen differenzgleichungen der mathematischen physik[J]. Mathematische Annalen, 1928, 100: 32-74.
- [8] CRANK J, NICOLSON P. A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type[J]. Mathematical Proceedings of the Cambridge Philosophical Society, 1947, 43(1): 50-67.
- [9] COURANT R, ISAACSON E, REES M. On the solution of nonlinear hyperbolic differential equations by finite differences[J]. Communications on Pure and Applied Mathematics, 1952, 5(3): 243-255.
- [10] FOSTER N, FEDKIW R. Practical animation of liquids[C]//The 28th Annual Conference on Computer Graphics and Interactive Techniques. New York: ACM, 2001: 23-30.
- [11] RASMUSSEN N, NGUYEN D Q, GEIGER W, et al. Smoke simulation for large scale phenomena[J]. ACM Transactions on Graphics, 2003, 22(3): 703-707.
- [12] HONG J M, SHINAR T, FEDKIW R. Wrinkled flames and cellular patterns[C]//ACM SIGGRAPH 2007 Papers. New York: ACM, 2007: 47.
- [13] HRENNIKOFF A. Solution of problems of elasticity by the framework method[J]. Journal of Applied Mechanics, 1941, 8(4): A169-A175.
- [14] COURANT R. Variational methods for the solution of problems of equilibrium and vibrations[J]. Bulletin of the American Mathematical Society, 1943, 49(1): 1-23.
- [15] 冯康. 基于变分原理的差分格式[J]. 应用数学与计算数学, 1965, 2(4): 238-262.
- [15] FENG K. Difference schemes based On variational principle[J]. Journal of Applied and Computational Mathematics, 1965, 2(4): 238-262 (in Chinese).
- [16] YOSIBASH Z, WEISS D, HARTMANN S. High-order FEMs for thermo-hyperelasticity at finite strains[J]. Computers & Mathematics with Applications, 2014, 67(3): 477-496.
- [17] MARTIN A, ESNAULT J B, MASSIN P. About the use of standard integration schemes for X-FEM in solid mechanics plasticity[J]. Computer Methods in Applied Mechanics and Engineering, 2015, 283: 551-572.
- [18] GINGOLD R A, MONAGHAN J J. Smoothed particle hydrodynamics: theory and application to non-spherical stars[J]. Monthly Notices of the Royal Astronomical Society, 1977, 181(3): 375-389.
- [19] LUCY L B. A numerical approach to the testing of the fission hypothesis[J]. The Astronomical Journal, 1977, 82: 1013-1024.
- [20] SOLENTHALER B, PAJAROLA R. Predictive-corrective incompressible SPH[J]. ACM Transactions on Graphics, 2009, 28(3): 40.
- [21] HE X W, LIU N, LI S, et al. Local Poisson SPH for viscous incompressible fluids[J]. Computer Graphics Forum, 2012, 31(6): 1948-1958.
- [22] IHMSEN M, CORNELIS J, SOLENTHALER B, et al. Implicit incompressible SPH[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(3): 426-435.
- [23] BENDER J, KOSCHIER D. Divergence-free smoothed particle hydrodynamics[C]//The 14th ACM SIGGRAPH/Eurographics Symposium on Computer Animation. New York: ACM, 2015: 147-155.

- [24] HE X W, LIU N, WANG G P, et al. Staggered meshless solid-fluid coupling[J]. *ACM Transactions on Graphics*, 2012, 31(6): 149.
- [25] BAND S, GISSLER C, IHMSEN M, et al. Pressure boundaries for implicit incompressible SPH[J]. *ACM Transactions on Graphics*, 2018, 37(2): 14.
- [26] PEER A, TESCHNER M. Prescribed velocity gradients for highly viscous SPH fluids with vorticity diffusion[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2017, 23(12): 2656-2662.
- [27] LIU S S, HE X W, WANG W C, et al. Adapted SIMPLE algorithm for incompressible SPH fluids with a broad range viscosity[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2022, 28(9): 3168-3179.
- [28] OÑATE E, IDELSOHN S, ZIENKIEWICZ O C, et al. A stabilized finite point method for analysis of fluid mechanics problems[J]. *Computer Methods in Applied Mechanics and Engineering*, 1996, 139(1/4): 315-346.
- [29] ATLURI S N, ZHU T L. New concepts in meshless methods[J]. *International Journal for Numerical Methods in Engineering*, 2000, 47(1/3): 537-556.
- [30] SILLING S A. Reformulation of elasticity theory for discontinuities and long-range forces[J]. *Journal of the Mechanics and Physics of Solids*, 2000, 48(1): 175-209.
- [31] HE X W, WANG H M, WU E H. Projective peridynamics for modeling versatile elastoplastic materials[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2018, 24(9): 2589-2599.
- [32] LU Z X, HE X W, GUO Y Z, et al. Projective peridynamic modeling of hyperelastic membranes with contact[J]. *IEEE Transactions on Visualization and Computer Graphics*, 2024, 30(8): 4601-4614.
- [33] YANG Y B, PERDIKARIS P. Adversarial uncertainty quantification in physics-informed neural networks[J]. *Journal of Computational Physics*, 2019, 394: 136-152.
- [34] HAGHIGHAT E, RAISSI M, MOURE A, et al. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics[J]. *Computer Methods in Applied Mechanics and Engineering*, 2021, 379: 113741.
- [35] JIN X W, CAI S Z, LI H, et al. NSFnets (Navier-Stokes flow nets): physics-informed neural networks for the incompressible Navier-stokes equations[J]. *Journal of Computational Physics*, 2021, 426: 109951.
- [36] SAAD Y. *Iterative methods for sparse linear systems*[M]. 2nd ed. Philadelphia: Society for Industrial and Applied Mathematics, 2003: 105-132.
- [37] NOCEDAL J, WRIGHT S J. *Numerical optimization*[M]. New York: Springer, 1999: 101-134.
- [38] KIRCHDOERFER T, ORTIZ M. Data-driven computational mechanics[J]. *Computer Methods in Applied Mechanics and Engineering*, 2016, 304: 81-101.
- [39] XU L Y, HE X W, CHEN W, et al. Reformulating hyperelastic materials with peridynamic modeling[J]. *Computer Graphics Forum*, 2018, 37(7): 121-130.
- [40] LAI W M, RUBIN D, KREML E. *Introduction to continuum mechanics*[M]. 4th ed. Boston: Butterworth-Heinemann, 2010: 220-225.
- [41] MÜLLER M, CHARYPAR D, GROSS M. Particle-based fluid simulation for interactive applications[C]//The 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Goslar: Eurographics Association, 2003: 154-159.
- [42] CAI S Z, MAO Z P, WANG Z C, et al. Physics-informed neural networks (PINNs) for fluid mechanics: a review[J]. *Acta Mechanica Sinica*, 2021, 37(12): 1727-1738.
- [43] CIFTCI K, HACKL K. A physics-informed GAN framework based on model-free data-driven computational mechanics[J]. *Computer Methods in Applied Mechanics and Engineering*, 2024, 424: 116907.
- [44] ABADI M, BARHAM P, CHEN J M, et al. TensorFlow: a system for large-scale machine learning[C]//The 12th USENIX symposium on Operating Systems Design and Implementation. Berkeley: USENIX Association, 2016: 265-283.
- [45] LU L, JIN P Z, PANG G F, et al. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators[J]. *Nature Machine Intelligence*, 2021, 3(3): 218-229.
- [46] LIU M B, LIU G R. Smoothed particle hydrodynamics (SPH): an overview and recent developments[J]. *Archives of Computational Methods in Engineering*, 2010, 17(1): 25-76.
- [47] KEYES D E, MCINNES L C, WOODWARD C, et al. Multiphysics simulations: challenges and opportunities[J]. *The International Journal of High Performance Computing Applications*, 2013, 27(1): 4-83.
- [48] FAURE F, DURIEZ C, DELINGETTE H, et al. SOFA: a multi-model framework for interactive physical simulation[M]//Soft Tissue Biomechanical Modeling for Computer Assisted Surgery. Heidelberg: Springer, 2012: 283-321.
- [49] HE X W, CHANG YUE, SHI J, et al. PeriDyno: an AI-Targeted physical simulation platform[EB/OL]. [2024-06-15]. <https://github.com/peridyno/peridyno/>.
- [50] WANG R, QIAN X L. Welcome to osg3-cookbook's documentation[EB/OL]. [2024-06-15]. <https://osg3cookbook.readthedocs.io/en/latest/index.html>.
- [51] Unity Technologies. ECS for unity[EB/OL]. [2024-06-15]. <https://unity.com/ec3#resources>.
- [52] SCHECHTER H, BRIDSON R. Ghost SPH for animating water[J]. *ACM Transactions on Graphics*, 2012, 31(4): 61.
- [53] MACKLIN M, MÜLLER M. Position based fluids[J]. *ACM Transactions on Graphics*, 2013, 32(4): 104.
- [54] LIU S S, HE X W, GUO G Z, et al. A dual-particle approach for incompressible SPH fluids[J]. *ACM Transactions on Graphics*, 2024, 43(3): 28.
- [55] LI Y, WANG H, LI B Y, et al. An image-driven uncertainty inverse method for sheet metal forming problems[J]. *Journal of Mechanical Design*, 2022, 144(2): 022001.
- [56] YIN J C, WEN Z M, LI S H, et al. Dynamically configured physics-informed neural network in topology optimization applications[J]. *Computer Methods in Applied Mechanics and Engineering*, 2024, 426: 117004.
- [57] WEN Z M, LI Y, WANG H, et al. Data-driven spatiotemporal modeling for structural dynamics on irregular domains by stochastic dependency neural estimation[J]. *Computer Methods in Applied Mechanics and Engineering*, 2023, 404: 115831.
- [58] STOMAKHIN A, SCHROEDER C, CHAI L, et al. A material point method for snow simulation[J]. *ACM Transactions on Graphics*, 2013, 32(4): 102.
- [59] ZHU Z, WANG X F, ZHAO W B, et al. Is Sora a world simulator? A comprehensive survey on general world models and beyond[EB/OL]. [2024-05-31]. <https://doi.org/10.48550/arXiv.2405.03520>.