arXiv:2302.09149v1 [cs.MS] 17 Feb 2023

一种针对大规模空间并行与分布式但时间串行数据的增量奇异值分解方法——应用于技术流动

Niklas K¨uhl*[1,2]，Hendrik Fischer[3,4]，Michael Hinze[5]，和 Thomas Rung[2]

[1]汉堡船模试验池，Bramfelder Strasse 164，D-22305 汉堡，德国[2]汉堡工业大学，流体动力学与船舶理论研究所，Am Schwarzenberg-Campus4，D-21075 汉堡，德国[3]汉诺威莱布尼茨大学，应用数学研究所，Welfengarten 1，D-30167 汉诺威，德国[4]巴黎萨克雷大学、CentraleSup´elec、巴黎萨克雷高等师范学院、巴黎萨克雷力学实验室（LMPS），F-91190 吉夫叙尔伊韦特，法国[5]科布伦茨-兰道大学，数学系，科布伦茨校区，Universit¨atsstrasse 1，D-56070 科布伦茨

2023年2月21日

摘要

本文提出了一种策略，用于对随时间演化的空间三维离散数据集构建增量奇异值分解（SVD）。文中介绍了一种低内存访问开销的流程，用于缩减和部署快照数据。所考虑的示例涉及从非定常流动模拟中提取的计算流体动力学（CFD）结果，这些模拟采用区域分解策略和消息传递接口（MPI）处理器间通信协议进行空间并行计算。该框架面向用于实际应用的先进偏微分方程求解器。具体而言，假设数据的空间规模不随时间变化，且由于计算网格规模庞大、场量数量众多，其规模远大于时间维度。尽管该方法应用于技术流动，但也适用于计算科学与工程（CSE）领域中类似的应用。

这项研究重点探讨了沿单指令多数据（SIMD）模式，在分布式内存系统上通过空间分解数据集执行大规模仿真所涉及的问题，并详细审视了与性能相关的各个方面。为此，我们引入一种批次矩阵，用于聚合多个时间步和奇异值分解更新，从而显著提高计算效率。

通过模拟圆柱周围的二维层流单相流，首先对增量SVD策略进行了验证和确认。随后，研究分析了所提出的策略在湍流两相流中的二维浸没水翼上的应用。重点考察了基于局部和全局流动量的基于奇异值分解的重构的准确性、其物理可实现性、对区域划分的独立性以及相关的实现细节。此外，还评估了较低构造秩阈值和（自适应）较高构造秩阈值对工作量和准确性的影响。最后，提供了各种伪代码，以支持过程的可追溯性和重新编程。

采用增量SVD过程，对谐和迎浪中Fn $= 0.26$ 和$\mathrm{Re_L} = 1.4 \cdot 10^7$条件下Kriso集装箱船周围的预测流场进行分析和压缩。在数值开销为$\mathcal{O}$(10%)的情况下，在约3000个处理器上计算得到的大小为 $\mathcal{O}(\mathbb{R}^{10E8\times 10E4})$的快照矩阵可通过增量方式压缩 $\mathcal{O}$(95%)。存储量的减少伴随着积分力和局部波面升高量误差 $\mathcal{O}(10^{-2}$%)。

关键词： – 主成分分析、降阶建模、增量奇异值分解、大型时空数据集、纳维–斯托克斯流、计算流体动力学

*niklas.kuehl@tuhh.de

1 引言

如今，工程计算方法不仅需要提供设备在特定运行条件下进行性能分析所得的结果。此外，仿真结果——其中通常涉及数百万乃至更多的自由度（DoF）——还应能够用于研究设计修改、不同的代价泛函和/或不同的运行条件，并且最好能够以较低的计算成本实现。

当关注点集中于少数几个（积分）代价函数（即输出参数），并将其视为少数输入参数的函数时，如今人们广泛采用基于数据的机器学习（ML）方法来建立这种输入输出关系。一个典型例子是翼型的动态力对其迎角的响应。数据驱动的 ML 方法能够以几乎可以忽略不计的计算成本，估计此前未被明确研究的输入参数所对应的代价泛函（输出）。然而，这类方法通常要求拥有大量数据，而这些数据是通过详尽的第一性原理模拟获得的，例如瞬态三维流体动力学模拟。由于自由度众多，后者计算成本高昂，并会产生大量的离线或后台计算开销。作为 ML 策略的替代方案，基于物理的降阶模型（ROM）被用于将详尽的第一性原理模拟中的信息压缩为可快速分析的形式，所采用的方法包括主成分分析（PCA），即本征正交分解（POD）、奇异值分解（SVD）以及基于 Karhunen-Lo`eve 展开的投影等不同变体，参见 Benner 等（2015）；Mainini 和 Willcox（2015）。ROM 旨在提取模拟场中占主导地位的物理特征或模态，而这些特征通常仅限于几十个贡献项。线性假设在一定程度上限制了 ROM 策略的应用。然而，由于其 Kolmogorov N-宽度快速衰减，ROM 近似的指数收敛性通常也能在非线性流体问题中得到恢复，例如 Lassila 等（2014），参见 Kolmogoroff（1936）。因此，解流形通常可以由低秩子空间良好地近似，而 POD（Sirovich（1987）；Kunisch 和 Volkwein（2002）；Willcox 和 Peraire（2002））仍然是非定常非线性系统模型降阶中广受认可的工具。一个例子是流体动力学 Navier-Stokes系统，其中该框架已成功应用于层流问题（Caiazzo 等人（2014）；Stabile 和 Rozza（2018）；Gr¨aßle 等（2019））以及湍流问题（Wang 等（2012）；Lorenzi 等（2016）；Hijazi 等（2020））。此外，对于线性问题，采用卷积神经网络（CNN）方法的 ROM 与 ML 方法之间存在明确联系（Murata 等（2020）；Agostini（2020））；这些方法将 CNN自编码器的约化空间与 ROM 策略的模态联系起来。因此，将基于数据的信息与基于物理的信息相融合的混合 ROM/ML策略正日益受到关注 Benner 等（2015）；Mainini 和 Willcox（2015）；Hesthaven 和 Ubbiali（2018）；Swischuk 等（2019）；Pache 和 Rung（2022）。其中许多技术都需要使用 SVD，这凸显了开发高效、理想情况下可增量执行且适用于大规模应用的并行奇异值分解的必要性。

典型的降维空间应用包括近实时提取进入决策支持系统的输出参数（Walton等（2013）；Mainini和 Willcox（2015）；Pache和Rung（2022）），或用于瞬态优化研究的场重构（Stoll和Breiten（2015）；Dolgov和Stoll（2017）；Vezyris等（2019）；B¨unger等（2020）；Benner等（2020a）；Li等人（2021）；Margetis等（2021、2022）；Kodakkal等（2022）；Nobis等（2023））。表面上看，数据约简/阶约简策略要求具备完整数据。然而，例如流体动力学模拟方法生成的时空数据集，其规模可能涉及 $O(10^8)$ 个网格点和 $O(10^5)$ 个时间步，因此数据处理可能成为一个问题。在此类情况下，采用适度计算开销的数据压缩具有重要意义，即利用现有的领域并行高性能计算（HPC）片段，同时在瞬态框架中增量式地准备约简。

论文的目标与提纲

本文旨在阐述一种针对随时间演化、空间划分的三维数据集实施增量阶约简方法的策略。文中提出了一种低内存访问过程，用于并行约简和部署快照数据。所考虑的示例涉及从非定常流动模拟中提取的计算流体动力学（CFD）结果；这些模拟采用区域分解策略和消息传递接口（MPI）处理器间通信协议（消息传递接口论坛（2021））进行空间并行计算。鉴于实际仿真中采用的庞大计算网格和所涉及的场量数量，假定数据的空间规模与时间无关，且显著大于时间规模。所提出的过程与网格类型/结构无关。该研究的部分内容此前已有论述，例如偏微分方程解跟踪（Fareed等（2018）；Fareed和Singler（2019，2020））、低秩偏微分方程近似（Stoll和Breiten（2015）；B¨unger等（2020）；Benner等（2020a））或大规模

、部分分布式系统/网络（Mastronardi等（2010）；Iwen和Ong（2016）；Lin和Tong（2021））。Li等（2022）最近发表的一篇论文与本文具有相同的研究动机，并讨论了类似问题。与本文的工作不同，该论文作者专注于构建POD基，而不是以高效利用内存的方式压缩全部快照数据。与本研究类似，他们对分布于各处理器上的数据逐步构建全局SVD，并使用libROM库Arrighi等（2015），但未深入描述并行化策略的细节，而这正是本文所提供的内容之一。上述研究未涉及大型工程系统与复杂物理现象的结合。二者的结合给增量构建降阶模型带来了特殊挑战，尤其是在计算效率方面，以及随后对ROM重构值物理可实现性的评估方面。本文研究中的示例涉及相浓度和湍动能的正值。不同于其他研究，我们在并行算法中采用快照分组，旨在显著降低在线SVD的计算开销，从而能够处理更大的数据集。此外，我们还结合了基于SVD模态能量含量的自适应秩确定准则。与先前工作相比，我们以标量积复杂度精确计算该量，从而无需复杂的能量界。采用快照分组的秩自适应并行增量SVD的潜力，体现在一个针对Re 14×10^6 and Fn $=0.26$的三维两相流复杂工程应用中；该应用涉及约3000万个控制体、在2880个处理器上执行的1万个时间步。

本文组织如下：第 2 节概述算法起点和理论框架。第 3 节专门讨论空间分布式、时间串行的实现，其中给出了若干伪代码，以提高对并行化方面的理解。此外，还提供了示例 Matlab©代码，参见 K¨uhl（2023）。第 4 节给出了所研究流场的控制方程，以及对所采用求解器的简要说明。第 5 节报告了二维通用层流绕圆柱流动的验证、验证和并行化结果，以及准确性和效率方面的内容。随后，第 6 节针对二维湍流两相流，研究了实际实现方面和物理可实现性，以及过程自适应性问题。第 7 节展示了规则迎浪作用下三维集装箱船基准算例周围两相流的性能。最后，第 8 节给出结论。

在本文的其余部分中，场量均采用由拉丁字母下标表示的笛卡尔坐标进行定义，并对重复下标应用爱因斯坦求和约定。此外，我们分别使用小写和大写粗体字母表示向量和张量。所有数值实验均在 NHR G¨ öttingen 和柏林完成，参见 www.hlrn.de。

2 算法起点

我们认为流场由 $s=[1,...,\mathrm{S}]$ 个独立状态组成，例如压力、密度、纵向速度等。此外，我们假设快照在分区（或处理器、线程） $p\in[1,...,\mathrm{P}]$ 的全部 $l=[1,...,\mathrm{L}]$ 个空间自由度上共存，例如有限体积格式中的控制体数量。

将所有长度为 $l=[1,...,\mathrm{L}]$ 的 $s=[1,...,\mathrm{S}]$ 个状态向量 $\boldsymbol{\varphi}$ 进行串联，即可构造出一个长度为 $\mathrm{Y}=(\mathrm{S}\,\mathrm{L})$ 的局部状态向量 $\boldsymbol{y}$，该向量表示分区 $p\in[1,...,\mathrm{P}]$, 在时刻 $t\in[1,...,\mathrm{T}]$ 的状态，即:

$$\boldsymbol{y}^{(t,p)}=\frac{1}{\mathrm{P}\,\mathrm{L}}\begin{bmatrix}\boldsymbol{\varphi}^{(1,1:\mathrm{L},\mathrm{t},\mathrm{p})}\\ \vdots\\ \boldsymbol{\varphi}^{(\mathrm{S},1:\mathrm{L},\mathrm{t},\mathrm{p})}\end{bmatrix}\qquad\text{with}\qquad\boldsymbol{\varphi}^{(\mathrm{s},1:\mathrm{L},\mathrm{t},\mathrm{p})}=\frac{1}{\varphi^{\mathrm{ref}}}\begin{bmatrix}\varphi^{(\mathrm{s},1,\mathrm{t},\mathrm{p})}\\ \vdots\\ \varphi^{(\mathrm{s},\mathrm{L},\mathrm{t},\mathrm{p})}\end{bmatrix}. \tag{1}$$

此处的 φ^{ref} 指的是一种特定状态参考值，用于对相关状态进行无量纲化，并支持将这些状态组合起来。为此，它充当预条件器，对可能具有显著不同数量级的场变量进行数值协调。此外，所有状态向量都除以离散全局自由度的常数数量（$=\mathrm{P}\,\mathrm{L}$），以便将后续的全局矩阵-向量乘积限制在双精度运算所支持的范围内。分区特定的状态快照向量提供了以下随时间演化的全局状态快照矩阵 $\boldsymbol{Y}\in\mathbb{R}^{\mathrm{N}\times\mathrm{T}}$

$$\boldsymbol{Y}=\begin{bmatrix}\boldsymbol{y}^{(1,1)} & \dots & \boldsymbol{y}^{(\mathrm{T},1)}\\ \vdots & & \vdots\\ \boldsymbol{y}^{(1,\mathrm{P})} & \dots & \boldsymbol{y}^{(\mathrm{T},\mathrm{P})}\end{bmatrix}, \tag{2}$$

其中通常满足 $\mathrm{N}=(\mathrm{SL})\,\mathrm{P}>>\mathrm{T}$，即全局状态矩阵的行数远多于列数。对于第 7 节讨论的应用，网格上涉及 $\mathrm{S}=7$ 个状态变量，网格规模约为 $\mathrm{L}\approx10\,000$

每个分区包含控制体，使用 $\mathrm{P} = 2880$ 个分区。对流场进行分析的时间范围为 $\mathrm{T} = 10\,000$ 个时间步，由此得到的全局状态矩阵规模约为 $\boldsymbol{Y} \in \mathbb{R}^{201\,600\,000\times 10\,000}$。管理如此规模的矩阵十分棘手，甚至可能无法实现，因此高效利用资源的奇异值分解（SVD）技术备受青睐。下一小节介绍标准信息，并为求完整起见，尤其是为了便于理解而予以重述；假定读者已熟悉相关证明。

2.1 奇异值分解

设 $\boldsymbol{Y} \in \mathbb{R}^{\mathrm{N}\times\mathrm{T}}$，则存在正交矩阵 $\boldsymbol{U} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_\mathrm{N}) \in \mathbb{R}^{\mathrm{N}\times\mathrm{N}}$ 和 $\boldsymbol{V} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\mathrm{T}) \in \mathbb{R}^{\mathrm{T}\times\mathrm{T}}$以及矩形对角矩阵 $\boldsymbol{S} = \mathrm{diag}(s_\mathrm{i}) \in \mathbb{R}^{\mathrm{N}\times\mathrm{T}}$，使得

$$\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T = \sum_{i=1}^{d} s_i \boldsymbol{u}_i \boldsymbol{v}_i^T \tag{3}$$

其奇异值 $s_1 \geq \ldots \geq s_\mathrm{d} \geq 0$均不相同，奇异向量为 $\boldsymbol{u}_\mathrm{i}$，$\boldsymbol{v}_\mathrm{j}$ 和 $d = \min(\mathrm{N}, \mathrm{T})$。注意，理论等式 $\boldsymbol{Y} = \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^T$仅在不存在数值误差时成立。由此，我们引入下面将用于调整奇异值分解的两个范数，即

$$||\boldsymbol{Y}||_2 = s_1 \qquad \text{and} \qquad ||\boldsymbol{Y}||_\mathrm{F} = \sqrt{\sum_{i=1}^{\mathrm{N}}\sum_{i=1}^{\mathrm{T}} |y_\mathrm{ij}|^2} = \sqrt{\sum_{i=1}^{d} s_\mathrm{i}^2}\,. \tag{4}$$

在式 (4) 中，$||\cdot||_2$ 和 $||\cdot||_\mathrm{F}$ 分别表示 L2 范数和 Frobenius范数。此外，我们定义$r = \max(k \in \mathbb{N} : s_\mathrm{k} \neq 0)$为非零奇异值的数量，该数量等于 $\mathrm{rank}(\boldsymbol{Y}) = r$。因此，只需前 r 个奇异向量即可确定式 (3) 中的矩阵 $\boldsymbol{Y}$。由此可以定义约化奇异值分解（rSVD）：设 $\mathrm{rank}(\boldsymbol{Y}) = r \leq \min(\mathrm{N}, \mathrm{T})$，使得

$$\boldsymbol{Y} = \boldsymbol{U}_\mathrm{r}\boldsymbol{S}_\mathrm{r}\boldsymbol{V}_\mathrm{r}^\mathrm{T} = \sum_{i=1}^{\mathrm{r}} s_\mathrm{i}\boldsymbol{u}_\mathrm{i}\boldsymbol{v}_\mathrm{i}^\mathrm{T} \tag{5}$$

其中，rSVD矩阵表示为 $\boldsymbol{U}_\mathrm{r} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_\mathrm{r}) \in \mathbb{R}^{\mathrm{N}\times\mathrm{r}}$和 $\boldsymbol{V}_\mathrm{r} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_\mathrm{r}) \in \mathbb{R}^{\mathrm{T}\times\mathrm{r}}$，以及 $\boldsymbol{S}_\mathrm{r} = \mathrm{diag}(s_1, \ldots, s_\mathrm{r}) \in \mathbb{R}^{\mathrm{r}\times\mathrm{r}}$。与 $\boldsymbol{Y}$不同，必须存储矩阵 $\boldsymbol{U}_\mathrm{r}$和 $\boldsymbol{V}_\mathrm{r}$以及对角线 $\boldsymbol{S}$项向量 $\boldsymbol{s} \in \mathbb{R}^\mathrm{r}$。实际上，当矩阵 $\boldsymbol{Y}$具有较高的秩时，尽管可以观察到奇异值快速衰减，这会导致存储开销增加。因此，需要进一步减少计算工作量。

由于其广泛的适用性和多功能性，通常会采用基于POD的模型降阶。该方法旨在实现低秩近似，同时通过根据特定的截断准则截去最小的非零奇异值来保持整体解的行为。根据Kolmogorov N-宽度快速衰减的假设（Kolmogorov（1936）；Benner等（2020b）），与较小奇异值对应的POD模态被认为对近似质量的影响微不足道，从而得到截断奇异值分解（tSVD）。$\boldsymbol{Y}$的秩为q 的tSVD近似，其中 $1 \leq q < r$，定义为

$$\boldsymbol{Y}_\mathrm{q} = \boldsymbol{U}_\mathrm{q}\boldsymbol{S}_\mathrm{q}\boldsymbol{V}_\mathrm{q}^\mathrm{T} = \sum_{\mathrm{i}=1}^{\mathrm{q}} s_\mathrm{i}\boldsymbol{u}_\mathrm{i}\boldsymbol{v}_\mathrm{i}^\mathrm{T}\,. \tag{6}$$

tSVD 矩阵表示为 $\boldsymbol{U}_\mathrm{q} = (\boldsymbol{u}_1, \ldots \boldsymbol{u}_\mathrm{q}) \in \mathbb{R}^{\mathrm{N}\times\mathrm{q}}$、$\boldsymbol{V}_\mathrm{q} = (\boldsymbol{v}_1, \ldots \boldsymbol{v}_\mathrm{q}) \in \mathbb{R}^{\mathrm{T}\times\mathrm{q}}$ 以及 $\boldsymbol{S}_\mathrm{q} = \mathrm{diag}(s_1, \ldots s_\mathrm{q}) \in \mathbb{R}^{\mathrm{q}\times\mathrm{q}}$。$\boldsymbol{Y}_\mathrm{q} \in \mathbb{R}^{\mathrm{N}\times\mathrm{T}}$ 表示由（5）中的 $\boldsymbol{Y}$ 的 q 个最大奇异值所得到的矩阵。没有明确的方法可以先验确定 q。通常，目标是使 q 尽可能小，同时又足够大，以便$\boldsymbol{Y}_\mathrm{q} \approx \boldsymbol{Y}$。尽管秩为 q 的近似在最小二乘意义下是最优的（Eckart 和 Young（1936）），但这并未为秩确定提供明确路径。用于评估 tSVD 质量的一项广泛采用的标准是指保留能量或信息含量 η^q 启发式方法，参见 Gr¨aßle 和 Hinze（2018年）；Gubisch 和 Volkwein（2017年）；Lassila 等（2014年）。后者没有严格的物理意义，但通常定义为 Frobenius 范数平方之比，参见公式（4），即:

$$\eta^\mathrm{q} = \frac{||\boldsymbol{Y}_\mathrm{q}||_\mathrm{F}^2}{||\boldsymbol{Y}||_\mathrm{F}^2} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{q}} s_\mathrm{i}^2}{\sum_{\mathrm{i}=1}^{\mathrm{r}} s_\mathrm{i}^2} = \frac{\sum_{\mathrm{i}=1}^{\mathrm{q}} s_\mathrm{i}^2}{\sum_{\mathrm{i}=1}^{\mathrm{T}} ||\boldsymbol{y}_\mathrm{i}||^2}\,. \tag{7}$$

公式（7）要求获得全部 r 奇异值，这对于技术应用中的大型矩阵而言并不现实。然而，由于公式（4）和 $d \to r$，只要在构造过程中至少有一次获得状态快照向量，就可以直接计算 $||\boldsymbol{Y}||_\mathrm{F}^2$。本文的时间推进框架正是如此，即 $\sum^{\mathrm{T}}{}_{\mathrm{i}=1} ||\boldsymbol{y}_\mathrm{i}||^2$ 可以按照下一节 2.2 所述的增量方式计算，且不会产生任何截断误差。

自适应启发式方法

本文所考虑的应用表明，最小截断秩 o 有利于提高自适应性。原因在于，通常会观察到前几个奇异值占主导地位，这往往会使公式（7）中的能量界趋近于机器精度。因此，结合两步法，对表达式（7）进行启发式调整。本文的启发式方法忽略前 o 个奇异值，以评估并调整奇异值分解的质量，即:

$$\eta^{\mathrm{o}}=\frac{\sum_{\mathrm{i=o}}^{\mathrm{q}}s_{\mathrm{i}}^{2}}{\sum_{\mathrm{i=o}}^{\mathrm{r}}s_{\mathrm{i}}^{2}}=\frac{\sum_{\mathrm{i=o}}^{\mathrm{q}}s_{\mathrm{i}}^{2}}{\sum_{\mathrm{i=1}}^{\mathrm{T}}||\boldsymbol{y}_{\mathrm{i}}||^{2}-\sum_{\mathrm{i=1}}^{\mathrm{o-1}}s_{\mathrm{i}}^{2}}\,. \tag{8}$$

注意，对于 $o=1$，可以恢复公式 (7)。由于该自适应准则在分母中包含精确参考能量，因此后者要减去由–始终可用的–前几个奇异值所贡献的能量。参数 o 用作自适应的起点（步骤 1）。本文所涉及的应用通常采用 $o=\mathcal{O}$（T/100）。尽管无法事先保证，但前 0.5%-1% 个模态通常能够保留相当一部分矩阵能量，即至少 50%。随后（步骤 2），应选择所需的 η^{o} 以启动秩自适应。由于 η^{o} 描述了在整个自适应过程中应恢复的初始遗漏矩阵能量所占的份额，因此这一选择取决于步骤 1。假设步骤 1 保留矩阵能量的 50% [60%、70%、80%、90%]，则选择 $\eta^{o}=0.8$ 最终将提供能量的 90% [92%、94%、96%、98%] 。同样，$\eta^{o}=0.9$ 将得到能量的 95% [96%、97%、98%、99%] 。这些示例数值表明，$\eta^{o}=[0.8,0.9]$ 是返回足够总能量的合理选择；而当步骤 1 低于 66% 时，$\eta^{o}=0.7$ 可能会变得至关重要，因为此时保留的矩阵能量低于 90%。

由上述内容可知，可根据公式 (3)、(5) 或 (6)，通过 (r,t) 奇异值分解近似全局状态矩阵。不过，一次性计算要求全局矩阵可用，而这对于大规模应用似乎并不可行，因此需要在线构造。

2.2 奇异值分解的加法修改

本节介绍一种算法，该算法无需了解底层系统矩阵的其他信息，即可逐步调整已有的奇异值分解。此方法可用于增量更新奇异值分解。为此，本文先简要概述一种通用方法，即对 rSVD 进行加法秩-b 修改，该方法主要由 Brand（2002、2006年）提出。

对于给定矩阵 $\boldsymbol{Y}\in\mathbb{R}^{\mathrm{N\times T}}$，设 $\boldsymbol{Y}=\boldsymbol{USV}^{\mathrm{T}}$ 为其秩为 r 的 rSVD。此外，设 $\boldsymbol{B}\in\mathbb{R}^{\mathrm{N\times b}}$ 和 $\boldsymbol{C}\in\mathbb{R}^{\mathrm{T\times b}}$为任意秩为 b 的矩阵，用于描述对系统矩阵 $\boldsymbol{Y}\rightarrow\boldsymbol{Y}+\boldsymbol{BC}^{\mathrm{T}}$的预期修改。那么，$\boldsymbol{Y}+\boldsymbol{BC}^{\mathrm{T}}=\tilde{\boldsymbol{U}}\tilde{\boldsymbol{S}}\tilde{\boldsymbol{V}}^{\mathrm{T}}$ 的 rSVD 可由以下三个更新后的 SVD 矩阵给出

$$\tilde{\boldsymbol{V}}=\begin{bmatrix}\boldsymbol{V} & \boldsymbol{Q}_{\mathrm{C}}\end{bmatrix}\boldsymbol{V}' \tag{9}$$

$$\tilde{\boldsymbol{S}}=\boldsymbol{S}' \tag{10}$$

$$\tilde{\boldsymbol{U}}=\begin{bmatrix}\boldsymbol{U} & \boldsymbol{Q}_{\mathrm{B}}\end{bmatrix}\boldsymbol{U}' \tag{11}$$

其中，$\boldsymbol{Q}_{\mathrm{B}}\boldsymbol{R}_{\mathrm{B}}=(\boldsymbol{I}-\boldsymbol{UU}^{\mathrm{T}})\boldsymbol{B}$和 $\boldsymbol{Q}_{\mathrm{C}}\boldsymbol{R}_{\mathrm{C}}=(\boldsymbol{I}-\boldsymbol{VV}^{\mathrm{T}})\boldsymbol{C}$分别由两次 QR分解得到。此外，$\boldsymbol{U}'\boldsymbol{S}'\boldsymbol{V}'^{\mathrm{T}}$ 源自矩阵 $\boldsymbol{K}\in\mathbb{R}^{(\mathrm{r+b})\times(\mathrm{r+b})}$ 的 rSVD，其形式为

$$\boldsymbol{K}=\begin{bmatrix}\boldsymbol{S} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}+\begin{bmatrix}\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}\\ \boldsymbol{R}_{\mathrm{B}}\end{bmatrix}\begin{bmatrix}\boldsymbol{V}^{\mathrm{T}}\boldsymbol{C}\\ \boldsymbol{R}_{\mathrm{C}}\end{bmatrix}^{\mathrm{T}}\,. \tag{12}$$

上述方法提供了多种修改系统矩阵的可能方式。除了更新和降级更新之外，还可以修改单个值，或交换行与列，参见 Brand（2002，2006）。所需的 QR分解例如采用改进的Gram-Schmidt过程，其分解得到的矩阵维度为 $\boldsymbol{Q}_{\mathrm{B}}\in\mathbb{R}^{\mathrm{N\times b}},\boldsymbol{Q}_{\mathrm{C}}\in\mathbb{R}^{\mathrm{T\times b}}$ 和 $\boldsymbol{R}_{\mathrm{B}},\boldsymbol{R}_{\mathrm{C}}\in\mathbb{R}^{\mathrm{b\times b}}$。上述修改可能导致秩增加，参见矩阵 $\boldsymbol{K}\in\mathbb{R}^{(\mathrm{r+b})\times(\mathrm{r+b})}$，因此也会增加˜内存消耗。由于式（10）中的 $\boldsymbol{S}=\boldsymbol{S}'$，对加性秩-$b$ 修改进行秩-q 截断，可以通过截断式（12）中的 $\boldsymbol{K}$ 来实现，即

$$\tilde{\boldsymbol{V}}_{\mathrm{q}}=\begin{bmatrix}\boldsymbol{V} & \boldsymbol{Q}_{\mathrm{C}}\end{bmatrix}\boldsymbol{V}'(:,1:q) \tag{13}$$

$$\tilde{\boldsymbol{S}}_{\mathrm{q}}=\boldsymbol{S}'(1:q,1:q) \tag{14}$$

$$\tilde{\boldsymbol{U}}_{\mathrm{q}}=\begin{bmatrix}\boldsymbol{U} & \boldsymbol{Q}_{\mathrm{B}}\end{bmatrix}\boldsymbol{U}'(:,1:q)\,. \tag{15}$$

本文重点讨论时间更新，即系统矩阵 $\boldsymbol{Y}$ 的列扩展。令 $\boldsymbol{C} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^{\mathrm{T}}$，通过矩阵 $\boldsymbol{B}$ 进行列更新的方法是：用 b 个额外的零列扩展 $\boldsymbol{Y}$ 和 $\boldsymbol{V}$，使得 $\begin{bmatrix}\boldsymbol{Y} & \boldsymbol{0}\end{bmatrix} = \boldsymbol{U}\boldsymbol{S}\begin{bmatrix}\boldsymbol{V} & \boldsymbol{0}\end{bmatrix}^{\mathrm{T}} \in \mathbb{R}^{\mathrm{N}\times(\mathrm{T}+\mathrm{b})}$。由于 $\boldsymbol{C}$ 表示一个增广单位矩阵，其 QR分解具有特定结构，因此可以简化先前的秩-b 修改，即 $\boldsymbol{Q}_{\mathrm{C}} = \boldsymbol{I}$ 和 $\boldsymbol{R}_{\mathrm{C}} = \boldsymbol{C}$。此外，$\boldsymbol{C} = \begin{bmatrix}\boldsymbol{0} & \boldsymbol{I}\end{bmatrix}^{\mathrm{T}}$ 中的零元素给出 $\begin{bmatrix}\boldsymbol{V} & \boldsymbol{0}\end{bmatrix}^{\mathrm{T}}\boldsymbol{C} = \boldsymbol{0}$，因此，式（13）至（15）中 $\boldsymbol{Y} + \boldsymbol{B}\boldsymbol{C}^{\mathrm{T}} \approx \tilde{\boldsymbol{U}}_{\mathrm{q}}\tilde{\boldsymbol{S}}_{\mathrm{q}}\tilde{\boldsymbol{V}}^{\mathrm{Tq}}$ 的一个 q奇异值分解现在可表示为

$$\tilde{\boldsymbol{V}}_{\mathrm{q}} = \begin{bmatrix}\boldsymbol{V} & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{I}\end{bmatrix}\boldsymbol{V}'(:, 1:q) \tag{16}$$

$$\tilde{\boldsymbol{S}}_{\mathrm{q}} = \boldsymbol{S}'(1:q, 1:q) \tag{17}$$

$$\tilde{\boldsymbol{U}}_{\mathrm{q}} = \begin{bmatrix}\boldsymbol{U} & \boldsymbol{Q}_{\mathrm{B}}\end{bmatrix}\boldsymbol{U}'(:, 1:q)\,, \tag{18}$$

其中，$\boldsymbol{U}'\boldsymbol{S}'\boldsymbol{V}'^{\mathrm{T}}$ 表示 $\boldsymbol{K}$ 的 rSVD

$$\boldsymbol{K} = \begin{bmatrix}\boldsymbol{S} & \boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}\\ \boldsymbol{0} & \boldsymbol{R}_{\mathrm{B}}\end{bmatrix}. \tag{19}$$

式（16）—（19）构成了本文即时计算或增量计算的截断奇异值分解（itSVD）的基础。

重要的阶段性结论： 在每个时间步，瞬时状态快照向量 $\boldsymbol{y}$ 都会作为附加列与所谓的批次矩阵 $\boldsymbol{B}$对齐；每经过 b 个时间步，当 $\boldsymbol{B}$ 填满后，就会依据式（16）—（19）将后者追加到之前的 itSVD 中。随后，$\boldsymbol{B}$ 被清空，过程重新开始。因此，在 $\boldsymbol{B}$的批量大小或列大小 b 、整体 itSVD 构建时间与所需内存之间存在权衡。可以设想两种极限情况：

（a）批次矩阵的列数达到时间步的最大数量（$b \to \mathrm{T}$），增量截断奇异值分解最终并入仿真结束时执行的经典一次性 SVD。

（b）相比之下，也可以在每个时间步扩展 itSVD，并将矩阵 $\boldsymbol{B}$ 转换为列向量（$b \to 1$）。

选项（a）主要会增加内存需求，而选项（b）则会延长仿真时间。与截断值 q类似，批次矩阵宽度 b应在满足需求的前提下尽可能取大。这个方面主要属于技术层面，但对于可实现的效率仍然至关重要。它主要受所使用硬件的影响——例如最大（随机访问）内存容量或仿真时间等——并将在第 5 节中进一步讨论。

2.3 并行itSVD的评估

itSVD完成后，例如在所关注的时间范围结束时，每一列——因而也是重构状态向量 $\tilde{\boldsymbol{\varphi}}(:) \approx \boldsymbol{\varphi}(:)$的近似——都可以通过随后对 itSVD 的评估得到。为此，将关系式（1）进行变形，可得到分区 p上状态 s 在时间 t 处、长度为 L 的近似局部瞬时快照向量 $\tilde{\boldsymbol{\varphi}}^{(\mathrm{s,1:L,t,p})}$ 的表达式，即

$$\begin{bmatrix}\tilde{\varphi}^{(\mathrm{s,1,t,p})}\\ \vdots\\ \tilde{\varphi}^{(\mathrm{s,L,t,p})}\end{bmatrix} = \varphi^{\mathrm{ref}}\tilde{\boldsymbol{\varphi}}^{(\mathrm{s,1:L,t,p})} \quad \text{with} \quad \begin{bmatrix}\tilde{\boldsymbol{\varphi}}^{(1,1:\mathrm{L,t,p})}\\ \vdots\\ \tilde{\boldsymbol{\varphi}}^{(\mathrm{S},1:\mathrm{L,t,p})}\end{bmatrix} = (\mathrm{P\,L})\,\tilde{\boldsymbol{y}}^{(t,p)}(:) = \tilde{\boldsymbol{U}}_q(:, 1:\tilde{q})\tilde{\boldsymbol{S}}_q(1:\tilde{q}, 1:\tilde{q})\tilde{\boldsymbol{V}}_q^{\mathrm{T}}(t, 1:\tilde{q})\,. \tag{20}$$

需要注意的是，重构（20）过程中所使用的参考量应与构造（1）阶段的参考量一致。关于总体自由度 P L 的常数因子，同样如此。此外，还引入了另一个参数 $\tilde{q} \le q$ ，该参数决定重构秩；为使基于 itSVD 的近似尽可能准确，其应与构造秩 $\tilde{q} \to q$ 一致。不过，在某些情况下，人们可能会关注更低秩的信息，例如只关注前 $\tilde{q}$ 个能量丰富模态。在这种情况下，式（20）中需要执行的矩阵-向量乘法的数值开销会降低。关于 $\tilde{q}$ 的选择（其取值因具体情况而异）对重构质量的影响，将在第 5 节中详细研究。

为简化符号表示并提高可读性，本文其余部分统一假设构造的是秩为 q 的 itSVD，因此，式（16）至（18）中的波浪号标记以及下标 $\tilde{\boldsymbol{U}}_q\tilde{\boldsymbol{S}}_q\tilde{\boldsymbol{V}}_{\mathrm{T}_q} \rightarrow \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}_{\mathrm{T}}$ 不再明确写出。

最后，应通过 itSVD 评估确保物理合理性。然而，数值误差，甚至不准确的近似（例如，由过小的构造秩/截断秩 $q/\tilde{q}$ 导致的近似），可能使重构场不再满足基本原理和条件。这些条件可能包括边界条件、远场条件或耦合条件，而这些条件可能不再得到满足。对于非齐次 Dirichlet 边界条件，例如对变化的流入条件的描述，可以通过引入提升函数来解决潜在缺陷，参见 Ballarin and Rozza (2016)；Nonino 等人 (2021)。在此过程中，从单状态快照中减去非齐次边界值，以生成齐次化对应项。这自然保证了重构单状态满足 Dirichlet 边界条件。然而，对于本研究而言，与边界条件相关的问题并不重要。更为关键的是可能包含物理上不合理内容的重构场，例如负能量或体积浓度，正如论文的应用部分所示，参见第 6 节。此外，当 itSVD 由源自连续性约束（例如）的无散快照构成时，任何重构的降阶解也都是无散的，即连续性方程会自动得到满足，参见 Quarteroni and Rozza (2014)；Caiazzo 等人 (2014)；Ballarin 等人 (2015)。

3 并行化方面

本文的总体目标是针对空间并行、时间串行（SPTS）数据集，增量式地构造截断奇异值分解，即过程上高度分布式的全局状态矩阵 $\boldsymbol{Y}$，参见公式（2）。图 1 示意了全局快照矩阵的并行构造，假设

1. 矩阵行按块以过程分布式方式分布，且各块长度不同（即行数，也就是每个处理器/分区/线程上的空间自由度乘以状态数）：$\mathrm{N1} = \mathrm{S} \cdot \mathrm{L1} = \mathrm{Y1}$，$\mathrm{N2} = \mathrm{S} \cdot \mathrm{L2} = \mathrm{Y2}$，……，$\mathrm{Np} = \mathrm{S} \cdot \mathrm{Lp} = \mathrm{Yp}$，但宽度始终相同（即列数/时间自由度/时间步数）$\mathrm{T1} = \mathrm{T2} = \ldots = \mathrm{Tp} = \mathrm{T}$；并且

2. 列数 T 显著少于行数，即 $\mathrm{N1} + \mathrm{N2} + \ldots + \mathrm{Np} = \mathrm{N} >> \mathrm{T}$，因此，在本文考虑的所有情形中，公式（3）[（5）] 中的 $\mathrm{d} = \mathrm{T}$ [r =T] 均适用于一般的 [约化] 奇异值分解。因此，时间步数决定了 itSVD 的秩。

图 1 特别表明，在所概述的 SPTS 情况下，矩阵 $\boldsymbol{S} \in \mathbb{R}^{\mathrm{T}\times\mathrm{T}}$ 和 $\boldsymbol{V} \in \mathbb{R}^{\mathrm{T}\times\mathrm{T}}$ 具有全局特征，即所有分区上的奇异值必须一致，而只有矩阵 $\boldsymbol{U} \in \mathbb{R}^{\mathrm{N}\times\mathrm{T}}$ 的元素可以在局部有所不同。

本节旨在提出一种适用于分布式数据的局部→全局算法，该算法无需了解底层全局系统矩阵，即可修改已有的全局 tSVD。随后，可以按照上一节所述，利用该方法增量更新秩为 q 的 tSVD。所提出的 SPTS itSVD策略的离散实现方案通过 Algs. 1-5 中的多个伪代码进行了概述。

目前，假设模拟的过去数据已经通过相应的 itSVD 完成降阶。批次矩阵 $\boldsymbol{B}$ 目前已填充完毕，因此现在需要根据第 2.2 节中的说明执行一次更新。因此，在两个位置都需要两个全局矩阵的乘积 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}$：一处是在 $(\boldsymbol{I} - \boldsymbol{U}\boldsymbol{U}_{\mathrm{T}})\boldsymbol{B}$ 的 QR 分解中，另一处是在组装 $\boldsymbol{K}$ 的右上部分时，参见公式（19）。不过，两个当前全局矩阵的内积可以先在线程上分别进行局部计算，然后按照局部→全局操作的方式，在处理器边界之间进行通信，参见 Alg. 3 以及图 3 中的示意图。本文采用 MPI 例程 $ALLREDUCE$ （消息传递接口论坛（2021）），该例程首先合并（即求和）所有进程的 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}$ 值，然后将结果发送回所有发送方，从而确保 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}$ 的全局一致性。随后即可组装 QR 分解并在局部执行，参见 Alg. 3。正交归一化采用一种经过修改的并行 Gram–Schmidt 方法，其中所需的全局内积再次基于 $MPI\text{-}ALLREDUCE$ 指令完成。一旦全局确定的 $\boldsymbol{Q}_{\mathrm{P}}$ 和 $\boldsymbol{R}_{\mathrm{P}}$ 矩阵在各局部均可用，就可以将其代入公式（18）和（19）。随后组装矩阵 $\boldsymbol{K}$，并可根据公式（5）计算其 rSVD。不过需要注意的是，$\boldsymbol{K}$ 主要由先前确定的奇异值 $\boldsymbol{S}$构成，这些奇异值应在所有分区上保持一致。此外，还会加入全局不变的子矩阵 $\boldsymbol{M} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}$ 和 $\boldsymbol{R}_{\mathrm{P}}$，因此 $\boldsymbol{K}$ 实际上指的是一个全局矩阵。因此，$\boldsymbol{K}$ 的 rSVD

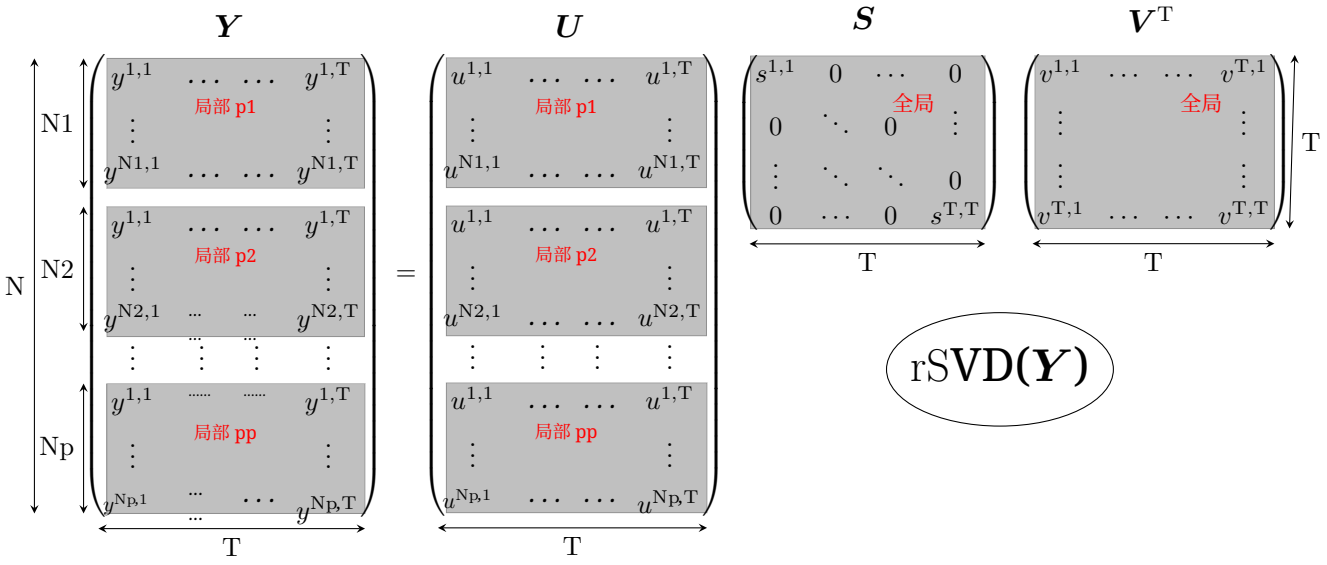


图 1：基于按空间（或按过程）分布的局部状态子矩阵，将全局总状态矩阵 $\boldsymbol{Y}$(N >>T) 进行缩减后的、空间上（行，N）并行/时间上（列，M）串行奇异值分解的示意图，其中局部状态子矩阵被分解为均等分布的局部子矩阵 $\boldsymbol{U}$ 以及两个全局矩阵 $\boldsymbol{S}$和 $\boldsymbol{V}$。

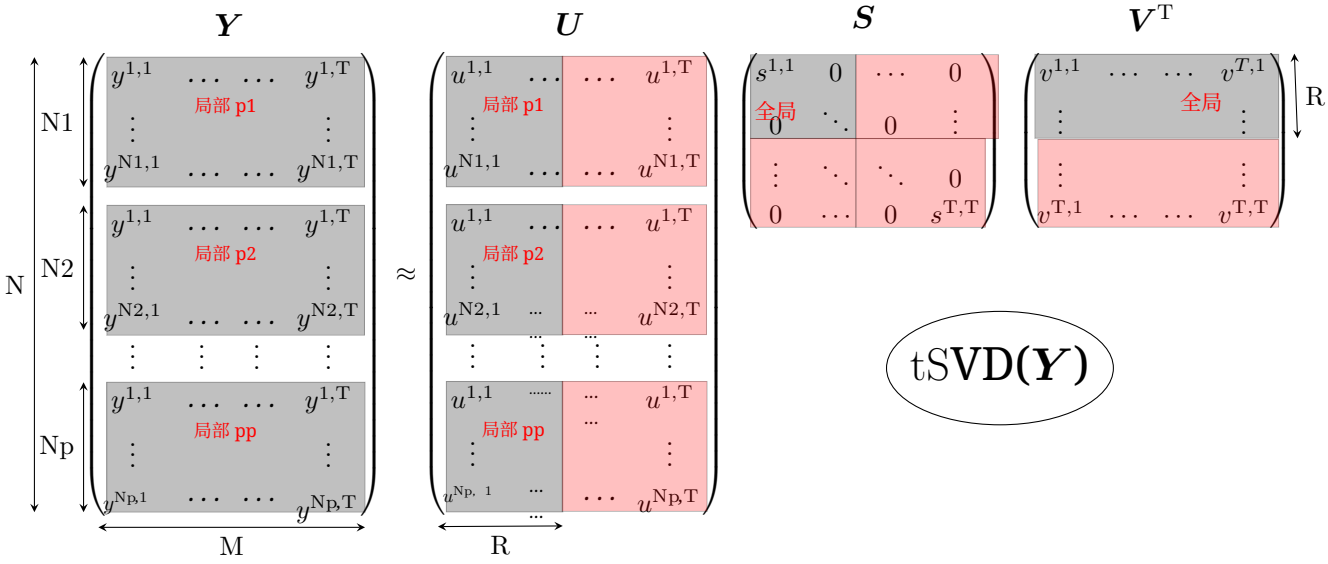


图 2：基于按空间（或按过程）分布的局部状态子矩阵，将全局总状态矩阵 $\boldsymbol{Y}$(N >>T) 进行截断后的、空间上（行，N）并行/时间上（列，M）串行奇异值分解的示意图，其中局部状态子矩阵被分解为均等分布的局部子矩阵 $\boldsymbol{U}$ 以及两个全局矩阵 $\boldsymbol{S}$和 $\boldsymbol{V}$。

$\boldsymbol{K}$ 可以在所有线程上同时计算。然而，只有在不存在数值舍入误差的情况下，才能期待得到完全相同的结果。因此，出于安全考虑，建议——本研究也采用了这种做法——在一个分区上执行 $\boldsymbol{K}$ 的 rSVD，然后将结果传递给其余所有线程，例如使用 *MPI-BCAST* 命令，参见 Alg. 3。本文使用 *LAPACK* 库确定 $\boldsymbol{K}$ 的局部 rSVD，具体而言，采用其内置的 *DGESVD* 例程，参见 Anderson 等人（1999）。一旦所有处理器上都获得了 $\boldsymbol{U}'\boldsymbol{S}'\boldsymbol{V}'$分解 $\boldsymbol{K}$，即可更新 $\boldsymbol{USV}$矩阵，参见公式（16）—（18）；为提高效率，仅将 $\boldsymbol{S} \rightarrow \boldsymbol{s}$ 的对角线存储在一维数组中。另一个技术性观察是：

算法1： 空间并行/时间串行增量更新的构造算法

截断的秩-q 奇异值分解，其中 P、L、Y、T 已知，来源于底层仿真框架，参见第 2 节。

输入： $b,\ q,\ o \in \mathbb{N}$ // Bunch size, truncation and minimum truncation rank

输出： $\boldsymbol{U} \in \mathbb{R}^{\mathrm{Y}\times \mathrm{q}},\ \boldsymbol{S} \in \mathbb{R}^{\mathrm{q}},\ \boldsymbol{V} \in \mathbb{R}^{\mathrm{T},\ \mathrm{q}}$ // SVD matrices/vectors that we are looking for

可选输出： $\boldsymbol{F} \in \mathbb{R}^{\mathrm{T},h}$ // h quantities of interest (e.g. drag/lift coefficient) over time

1 **模块**构造 奇异值分解():

2 **公有** 分配($\boldsymbol{U}$(Y,1),$\boldsymbol{s}$(1),$\boldsymbol{V}$(1,1),$\boldsymbol{B}$(Y,b)) // itSVD-relevant matrices per processor

3 **公有** $e = 0$ // Matrix energy value incrementally updated

4 **私有** $k = 0$ // Bunch matrix counter

5 **对于** t=1,T **do**

6 获取 $\boldsymbol{\varphi}$(:) // Solve for 1 = [1, ..., S] instantaneous state variables

7 $\boldsymbol{y}$(:) $\leftarrow \boldsymbol{\varphi}$(:)/(P L φ^{ref})) // Construct local, non-dimensionalized state vector, cf. Eqn. (1

8 确定 $\boldsymbol{F}(t,:)$ // Determine quantities of interest (i.e. drag/lift coefficient)

9 $e \leftarrow$并行求和($\boldsymbol{y}$(:)$^{\mathrm{T}}\boldsymbol{y}$(:)) // Update the global matrix energy, cf. Eqns. 7-8

10 **如果** $t = 1$ **则**

11 $\boldsymbol{V}(1,\ 1) = 1$ // First r.h.s. itSVD vector

12 $\boldsymbol{s}(1) = \boldsymbol{y}(:)^{\mathrm{T}}\boldsymbol{y}(:)$ 并行求和($\boldsymbol{s}(1)$), $\boldsymbol{s}(1) = \sqrt{\boldsymbol{s}(1)}$ // First singular value

13 $\boldsymbol{U}(:,1) = \boldsymbol{y}(:)/\boldsymbol{s}(1)$ // First l.h.s. itSVD vector

14 **else**

15 $k++$ // Increase the bunch matrix counter

16 $\boldsymbol{B}(:\ k) \leftarrow \boldsymbol{y}(:)$ // Update the bunch matrix

17 **如果** (k=b)**或** (t=T) **则**

18 itSVD 更新() // Update the itSVD

19 $\boldsymbol{B}(:,:) = 0,\ k = 0$ // clear the bunch matrix and its counter

20 **如果** procID=0**则**

21 保存 $\boldsymbol{V}$, $\boldsymbol{s}$, $\boldsymbol{F}$ // Store global data in one file only, e.g. S_svd_0.h5

22 保存 $\boldsymbol{U}$ // Store local data for every processor, e.g. U_svd_<procID>.h5

23 **返回** 0

对于较小的批次矩阵宽度 b，itSVD算法会被频繁调用，而其中涉及的代数子空间旋转可能无法保持正交性。因此，矩阵$[\boldsymbol{U}\quad \boldsymbol{Q}_{\mathrm{P}}]$可能会出现由数值误差引起的正交性丧失，进而导致更新后的 $\boldsymbol{U}$ 可能出现问题。其他作者（Brand（2006）；Fareed等（2018）；Bach等（2019）；Fareed和Singler（2019、2020）；Zhang（2022））也认识到了这一问题，因此建议额外执行一个作用于 $\boldsymbol{Q}$的全局正交归一化步骤。随后，批次矩阵 $\boldsymbol{B}$ 被清空，并继续进行时间积分 b 个时间步，直到执行下一次itSVD 更新。

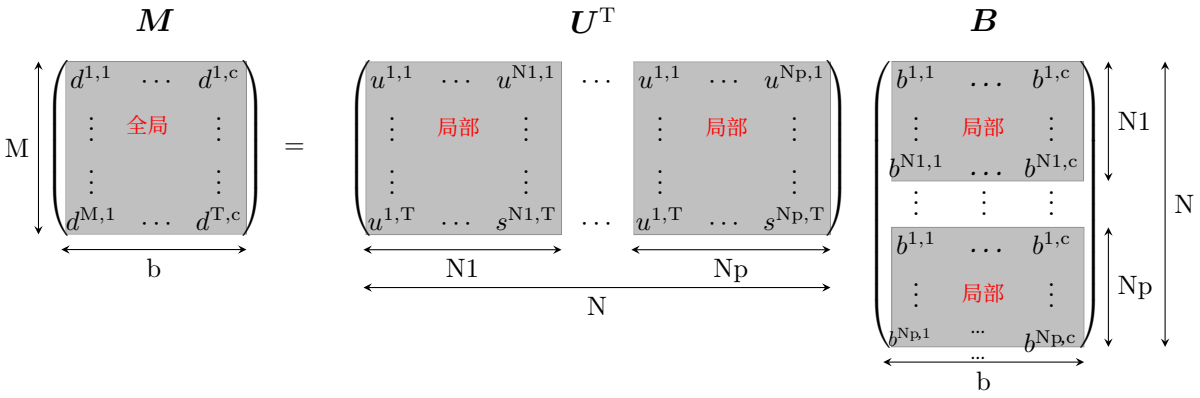


图3：两个分布式矩阵 $\boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}$ 相乘得到全局常量矩阵 $\boldsymbol{M}$ 的示意图，参见算法2第3行。

算法 2： 空间并行/时间串行截断奇异值分解的增量秩 b 更新
分解，其中 Y 已由底层仿真框架知晓，参见第 2 节

子程序itSVD 更新()**：**

$u =$ 大小$(\boldsymbol{U},\ 2)$，$v =$ 大小$(\boldsymbol{V},\ 1)$，$c = u + b$，$d = v + b$，$l =$ 最小值$(q,\ c)$； // Required sizes

$\boldsymbol{M} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}$，并行求和$(\boldsymbol{M})$； // 1) $\boldsymbol{M} \in \mathbb{R}^{u\times b}$ 2) Local to Global Operation. cf. Fig. 3

$\boldsymbol{P} = \boldsymbol{B} - \boldsymbol{U}\boldsymbol{M}$； // $\boldsymbol{P} \in \mathbb{R}^{\mathrm{Y}\times b}$ around Eqn. 12

$[\boldsymbol{Q}_{\mathrm{P}}, \boldsymbol{R}_{\mathrm{P}}] =$ 并行格拉姆-施密特正交化$(\boldsymbol{P})$； // Global QR Decomposition

如果 procID=0**则**

$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{S} \\ \boldsymbol{0} \end{bmatrix}$； // Prepare the SVD input matrix $\boldsymbol{K} \in \mathbb{R}^{c\times c}$, cf. Eqn. 19

$[\boldsymbol{U}', \boldsymbol{s}', \boldsymbol{V}'] =$ 奇异值分解$(\boldsymbol{K})$； // Local SVD yields $\boldsymbol{U}'$, $\boldsymbol{V}' \in \mathbb{R}^{c\times c}$ and $\boldsymbol{s}' \in \mathbb{R}^{c}$

如果自适应**则**

q = 自适应截断$(\boldsymbol{s}')$； // Adapt the truncation rank

并行广播$(\boldsymbol{U}',\ \boldsymbol{s}',\ \boldsymbol{V}',\ q)$； // Globalize the results

$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{V} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$； // Prepare $\boldsymbol{V}$ Update via $\boldsymbol{R} \in \mathbb{R}^{d\times l}$, cf. Eqn. 13

$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{U} & \boldsymbol{Q}_{\mathrm{P}} \end{bmatrix}$； // Prepare $\boldsymbol{U}$ Update via $\boldsymbol{Q} \in \mathbb{R}^{\mathrm{Y}\times l}$, cf. Eqn. 15

$[\boldsymbol{Q}_{\mathrm{Q}}, \cdot] =$ 并行格拉姆-施密特正交化$(\boldsymbol{Q})$； // Global QR Decomposition

释放分配$(\boldsymbol{U},\ \boldsymbol{s},\ \boldsymbol{V})$，分配$(\boldsymbol{U}(\mathrm{Y}\ l)\ \boldsymbol{s}(l)\ \boldsymbol{V}(d,\ l))$； // Increase the $\boldsymbol{U}$, $\boldsymbol{S}$, and $\boldsymbol{V}$ sizes

$\boldsymbol{V} = \boldsymbol{R}\,\boldsymbol{V}'(:,1{:}l)$； // Update $\boldsymbol{V}$, cf. Eqn. 13

$\boldsymbol{s} = \boldsymbol{s}'(1{:}l)$； // Update $\boldsymbol{s}$, cf. Eqn. 14

$\boldsymbol{U} = \boldsymbol{Q}_{Q}\,\boldsymbol{U}'(:,\ 1{:}l)$； // Update $\boldsymbol{U}$, cf. Eqn. 15

计算工作包括以下三项操作。首先，用于获得$\boldsymbol{Q}_{\mathrm{P}}$、 $\boldsymbol{R}_{\mathrm{P}}$ 和 $\boldsymbol{Q}_{\mathrm{Q}}$ 的QR分解分别需要 $\mathcal{O}(N\,q^2)$和 $\mathcal{O}(N\,(u+b)^2)$。其次，公式（19）中 $\boldsymbol{K}$ 的奇异值分解复杂度为 $\mathcal{O}((u+b)^3)$。第三，公式（13）和（15）中的子空间旋转需要 $\mathcal{O}((N+T)(u+b)q)$次运算（Brand（2002））。由itSVD算法2带来的额外计算工作将在第7节后续应用研究中进行分析。

当先前的仿真没有 rSVD 信息，且所考虑的时间历史仅限于一个时间步时，会出现一种特殊情况。在这种情况下，全局秩-1 rSVD 只需读取 $\boldsymbol{y} = \boldsymbol{u}\,s\,v$，其中 $\boldsymbol{u} = \boldsymbol{y}/s$、$s = ||\boldsymbol{y}||_2$ 和 $v = 1$ 可使用前文提到的 *ALLREDUCE* 例程快速获得。尽管从原理上讲并非必要，本文提出的 itSVD算法始终在第一个时间步之后从秩-1 rSVD 开始。随后，可将批次矩阵 $\boldsymbol{B}$ 构建为任意大的秩 b。后者可以根据第 6 节所述的底层仿真硬件进行调整。

在分布式内存环境中，如果每个线程存储其局部矩阵 $\boldsymbol{U}$，而只有一个分区存储全局向量 $\boldsymbol{S} \to \boldsymbol{s}$ 和矩阵 $\boldsymbol{V}$，则可以节省大量内存。对于包含数百至数千个分区的大规模并行应用而言，这一点至关重要，并且可以显著降低总内存开销。在本文中，使用 HDF5（The HDF集团（1997-NNNN））库对 itSVD 数据以及用于基准测试和比较的所有瞬时快照向量进行压缩和存储。

当时间步多于离散自由度时（即 N <T；不过，这种情况并未出现在本研究中），由于 itSVD 更新中的舍入误差，算法2中概述的过程可能会引发数值不稳定性。因此，通过增加一次矩阵乘法来正则化 $\boldsymbol{K} \to \boldsymbol{R}_{Q}\boldsymbol{K}$ 矩阵，从而扩展该过程。调整后的伪代码见附录中的算法6，其中与算法2的差异已突出显示。此外，还提供了示例 Matlab© 代码，参见 K¨uhl（2023）。通过对 $\boldsymbol{Q}$ 上的第二次 QR 分解进行适当的必要性检查，可以提高算法的效率，参见 Zhang（2022）；Li等（2022）。

4 控制物理方程与数值框架 k

本文的数值研究考察了两种互不相溶的惰性流体的流场（a，b），其体密度（ρ^{a}，ρ^{b}）和体黏度（μ^{a}，μ^{b}）均为常数，其中 b 需与批次矩阵大小区分。流体a 称为前景流体，流体 b 称为背景流体。即将进行的单相模拟

算法 3： 用于：1）通过基于格拉姆-施密特的 QR分解 对分布式行数据进行矩阵正交化；2）将不同局部值求和为相同的全局值；以及 3）将值从一个处理器广播给所有其他处理器的辅助函数和例程。

函数并行格拉姆-施密特正交化(A)：
  对于 j=1,size(A,2)**执行**
    $v(:) = A(:, j)$
    对于 i=1,j-1，**执行**
      $R(i, j) = Q(:, i)^{\mathrm{T}} v(:)$，并行求和($R(i, j)$)
      $v(:) = v(:) - R(i, j) Q(:, i)$
    $R(j, j) = v(:)^{\mathrm{T}} v(:)$，并行求和($R(j, j)$)，$R(j, j) = \sqrt{R(j, j)}$
    $Q(:, j) = v(:) / R(j, j)$
  return Q，R

子程序并行求和(vIn)：
  MPI_全归约(vIn)

子程序并行广播(vIn)：
  MPI_广播(vIn)

算法 4： 秩 $\tilde{q}$ 评估 ($\tilde{q} \leq q$) 的空间并行/时间串行截断 Sin-奇异值分解算法，其中 P、L 和 Y 已知于底层仿真框架，参见第 2 节

输入： $U \in \mathbb{R}^{\mathrm{Y} \times \mathrm{q}}$，$s \in \mathbb{R}^{\mathrm{q}}$，$V \in \mathbb{R}^{\mathrm{T},\ \mathrm{q}}$，$\tilde{q} \in \mathbb{N}$ // itSVD matrices/vectors and evaluation rank (note: $\tilde{q} \leq q$)
可选输出： $\tilde{F} \in \mathbb{R}^{\mathrm{T},\ h}$ // h quantities of interest (e.g. drag/lift coefficient) over time

函数计算 奇异值分解()：
  如果 procID=0**则**
    读取 V，s // Read global data in one file only, e.g. S_svd_0.h5
    并行广播(V，s) // Broadcast to all processors
  读取 U // Read local data for every processor, e.g. U_svd_<procID>.h5
  对于 t=1,T **do**
    $f = V(t,\ 1{:}\ \tilde{q}) s(1{:}\ \tilde{q})$ // $f \in \mathbb{R}^{\tilde{q}}$ first $\tilde{q}$ global modes
    $\tilde{y}(:) = U(:,\ 1{:}\ \tilde{q}) f(1{:}\ \tilde{q})$ // Compute the (approximated) local, non-dimensional state vector
    $\tilde{\varphi}(:) \leftarrow (\mathrm{P\ L}\ \varphi^{\mathrm{ref}}) \tilde{y}(:)$ // Extract the $s = [1, \ldots, \mathrm{S}]$ instantaneous state variables, cf. Eqn. (20)
    确保物理可行性
    确定 $\tilde{F}(t,\ :)$ // Determine (approximated) quantities of interest
  save $\tilde{F}$ // Store quantities of interest
  返回0

验证与确认第 5 节液滴流体 a. 在全文中，前景流体指空气，背景流体指水。根据 Hirt 和 Nichols（1981）提出的流体体积分数（VoF）方法，假定两种流体共享同一个运动学场。欧拉浓度场描述流体的空间分布，其中 $c = c^{\mathrm{a}} = V^{\mathrm{a}}/V$ 表示前景流体的体积浓度，而背景流体所占的体积分数表示为 $c^{\mathrm{b}} = V_{\mathrm{b}}/V = (V - V_{\mathrm{a}})/V = (1 - c)$。

控制方程主要指混合物的动量方程和连续性方程，以及关于前景相体积浓度和湍流量的附加输运方程。以下方程组需要求解压力 p、流体速度 v_i、浓度 c、湍流动能（TKE） k及其耗散率 ε，其中 $\mathrm{D}^{(\cdot)}$、$\mathrm{P}^{(\cdot)}$ 以及 $\mathrm{R}^{(\cdot)}$分别指它们的

算法5： 基于最小秩 o 和最大修正保留能量 η^{o}的自适应截断秩估计。

```
函数自适应截断(s)：
    如果 length(s) ≤ o 则
        q = length(s)                                   // Use all singular values
    else
        对于q=o,length(s) 执行
            如果 [s^T(o : q)s(o : q)]/[e − s^T(1 : o − 1)s(1 : o − 1)] ≥ η^o 则
                exit                                    // Increase q and leave when the energetic bound is met, cf. Eqn. 8
    返回 q
```

分别为扩散、生成和耗散，即：

$$\mathrm{R}^{\mathrm{p}} = \frac{\partial v_{\mathrm{k}}}{\partial x_{\mathrm{k}}} = 0 \tag{21}$$

$$\mathrm{R}^{\mathrm{c}} = \frac{\partial c}{\partial t} + \frac{\partial v_{\mathrm{k}} c}{\partial x_{\mathrm{k}}} = 0 \tag{22}$$

$$\mathrm{R}_{\mathrm{i}}^{\mathrm{v_i}} = \frac{\partial \rho v_{\mathrm{i}}}{\partial t} + \frac{\partial v_{\mathrm{k}} \rho v_{\mathrm{i}}}{\partial x_{\mathrm{k}}} + \frac{\partial}{\partial x_{\mathrm{k}}}\left[p^{\mathrm{eff}}\delta_{\mathrm{ik}} - 2\mu^{\mathrm{eff}} S_{\mathrm{ik}}\right] - \rho g_{\mathrm{i}} = 0 \tag{23}$$

$$\mathrm{R}^{\mathrm{k}} = \frac{\partial \rho k}{\partial t} + \frac{\partial v_{\mathrm{k}} \rho k}{\partial x_{\mathrm{k}}} - \mathrm{D}^{\mathrm{k}} - \mathrm{P}^{\mathrm{k}} + \mathrm{R}^{\mathrm{k}} = 0 \tag{24}$$

$$\mathrm{R}^{\varepsilon} = \frac{\partial \rho \varepsilon}{\partial t} + \frac{\partial v_{\mathrm{k}} \rho \varepsilon}{\partial x_{\mathrm{k}}} - \mathrm{D}^{\varepsilon} - \mathrm{P}^{\varepsilon} + \mathrm{R}^{\varepsilon} = 0\,. \tag{25}$$

其中，$\mu^{\mathrm{eff}} = \mu + \mu^{\mathrm{t}}$ 和 $p^{\mathrm{eff}} = p + p^{\mathrm{t}}$ 分别表示有效黏度和压力。单位坐标和应变率张量分别表示为克罗内克δ函数 δ_{ik} 和 $S_{\mathrm{ik}} = 1/2(\nabla_{\mathrm{k}} v_{\mathrm{i}} + \nabla_{\mathrm{i}} v_{\mathrm{k}})$。湍流平衡方程的源项和汇项是局部材料性质、速度梯度和湍流量的模型相关函数，例如，对于湍动能生成，可表示为 $P_{\mathrm{k}} := P^{\mathrm{k}}(\rho,\ \nabla_{\mathrm{k}} v_{\mathrm{i}},\ k,\ q)$，详细信息可参见相关文献，例如 Wilcox （1998）。前景相浓度 $c \in [c^{\mathrm{a}}, c^{\mathrm{b}}]$ 用于测量相和流体性质。该浓度范围从前景状态（$c^{\mathrm{a}},\ \rho^{\mathrm{a}},\ \mu^{\mathrm{a}}$）到背景状态（$c^{\mathrm{b}},\ \rho^{\mathrm{b}},\ \mu^{\mathrm{b}}$），而式（23）—（25）中的局部材料性质则由两个状态方程（EoS）得出，即：

$$\rho = m\rho^{\Delta} + \rho^{\mathrm{b}}, \qquad \text{and} \qquad \mu = m\mu^{\Delta} + \mu^{\mathrm{b}}, \tag{26}$$

其中，$\rho^{\Delta} = \rho^{\mathrm{a}} - \rho^{\mathrm{b}}$ 和 $\mu^{\Delta} = \mu^{\mathrm{a}} - \mu^{\mathrm{b}}$ 标示体密度和黏度的差异。标准状态方程对应于极限状态之间的简单线性插值，即 $m = c$，参见 Kr¨oger 等（2018年）；K¨uhl 等（2021a）

用于近似偏微分方程 (21)–(25) 的数值框架采用常规的、基于压力的二阶精度有限体积 (FV) 格式。可以使用基于任意多面体单元且可能包含悬挂节点的非结构网格，参见 Rung 等 (2009)。该顺序过程采用强守恒形式，并对所有输运性质使用以单元为中心、共点的存储布局。该框架专用于分布式内存并行CPU机器上的单指令多数据 (SIMD) 实现，可支持数千个进程；所采用的区域分解方法基于 (par)METIS算法 (Karypis (2011); Karypis 和 Schloegel (2011)) 以及 MPI 通信协议 (Yakubov 等 (2013)；消息传递接口论坛 (2021))。内部过程 FresCo^{+} 所采用的算法见 Rung 等 (2009)、Yakubov 等 (2013) 以及 K¨uhl (2021)。这些算法建立在通用欧拉输运方程的积分形式之上，该方程带有残差 R^{φ}，即方程 (21)–(25)，用于描述标量场 $\varphi(x_i,\ t)$；该标量场受到可能的非线性源项 $\mathcal{S}^{\varphi}$、建模的（非线性）梯度扩散项及其扩散率 Γ^{eff} 的影响，其中控制体 Ω 的边界为表面Γ，即：

$$\int \mathrm{R}^{\varphi}\mathrm{d}\Omega = 0\,, \qquad \rightarrow \qquad \int \left[\frac{\partial \varphi}{\partial t} - \mathcal{S}^{\varphi}\right]\mathrm{d}\Omega + \oint \left[u_{\mathrm{i}}\varphi - \Gamma^{\mathrm{eff}}\frac{\partial \varphi}{\partial x_{\mathrm{i}}}\right] n_{\mathrm{i}}\,\mathrm{d}\Gamma = 0\,. \tag{27}$$

此处， x_{i} 表示笛卡尔空间坐标，而 n_{i} 和 t 分别表示表面法向量分量和物理时间。因此，所采用的有限体积近似产生了一个规模为 P L $\times$ P L 的离散系统，其中 P L 表示控制体 (CV) 或离散自由度的数量，即一个状态快照向量的长度（参见第 2 节）。每一行对应于特定控制体的一个平衡方程，即:

$$A^{\varphi,\mathrm{CV}}\varphi^{\mathrm{CV}} - \sum_{\mathrm{NB(CV)}} A^{\varphi,\mathrm{NB}}\varphi^{\mathrm{NB}} = S^{\varphi,\mathrm{CV}}\,, \tag{28}$$

其中， $A^{\varphi,\mathrm{CV}}$、 $A^{\varphi,\mathrm{NB}}$ 和 $S^{\varphi,\mathrm{CV}}$ 分别表示一般源项形式下的主对角系数、其相邻项以及右端项。采用增强型压力修正格式迭代求解，直至收敛。通过适当的 Rhie-Chow 插值格式抑制压力与速度分量之间的奇偶解耦，参见 Rhie 和 Chow（1983年）；Yakubov等（2015年）；K¨uhl 和 Rung（2022年）。数值积分采用二阶中点法，其中离散扩散通量通过二阶精度中心差分计算，离散对流通量则采用高阶迎风偏置插值公式。时间导数采用隐式三时间层格式（ITTL），并结合子循环策略，以高效满足基于最大 Courant 数条件的稳定性准则；在细化的自由表面区域，Co $= \mathcal{O}(0.5)$，参见 Manzke等（2012年）；K¨uhl等（2021b年）。动量和湍流参数的对流输运采用对流运动学二次迎风插值（QUICK）格式。对于两相流，采用压缩型高分辨率界面捕捉（HRIC）方法对体积分数场进行对流输运。随后结合显式界面锐化（EIS）方法，确保流体-气体界面尽可能锐利，并将其控制在最多 1-2 个网格单元范围内（Manzke（2018年））。依据线性（Airy）波理论，通过调整浓度场方程组，在朝向内部的外边界处、缓冲区域内施加可能的时空波浪边界条件 c，即 $A^{c,\mathrm{P}}$、$A^{c,\mathrm{NB}}$ 和 $S^{c,\mathrm{P}}$ 见式（28）；也就是说，计算域的非线性 Navier-Stokes 解向边界方向逐渐过渡为线性波理论解，参见 W¨ ockner等（2010年）；Luo-Theilen 和 Rung（2017、2019年）。在本文考虑的所有情形中，缓冲区向计算域内延伸一个波长，其调节强度按二次方衰减。由于数据结构通常是非结构化的，因此采用适用于对称与非对称系统的预条件迭代稀疏矩阵求解器，例如广义最小残差法、双共轭梯度法、准最小残差法、共轭梯度平方法或稳定双共轭梯度法。

5 验证与确认

本节评估并验证所提出的 itSVD SPTS 方法的可信度。调查针对静止的二维浸没圆柱周围的层流单相流展开。研究考察局部质量指标，例如（$\tilde{\varphi} - \varphi$）$/\varphi^{\mathrm{ref}}$，以及通过阻力系数和升力系数获得的全局积分力，即:

$$c_{\mathrm{d}} = \frac{2F_i\delta_{i1}}{\rho A V_1} \qquad \text{and} \qquad c_{\mathrm{l}} = \frac{2F_i\delta_{i2}}{\rho A V_1} \qquad\qquad \text{with} \qquad\qquad F_i = \int_{\Gamma^{\mathrm{C}}} \left[p\delta_{\mathrm{ik}} - \mu 2S_{ik}\right] n_{\mathrm{k}}\,\mathrm{d}\Gamma\,. \tag{29}$$

此处 Γ^{C} 和 n_{k} 分别表示边界及其法向量，而 A 以及 V_1 分别指参考面积和主体入口速度。重力处于启用状态，并基于重力加速度 G、运动黏度 ν 以及预期的涡脱落频率 $f \approx 0.2$，在 $\mathrm{Re_D} = \rho^{\mathrm{b}} V_1 D/\mu^{\mathrm{b}} = 200$、$\mathrm{Fn} = V_1/\sqrt{G\,D} = 0.2828$ 和 $\mathrm{St} = Df/V_1 \approx 0.2$条件下进行瞬态研究。所采用的二维计算域如图 4 所示。其长度和高度分别为 $40\,D$和 $20\,D$，其中入口 [底部]边界位于 10 [9。距圆柱原点5] 个直径处。在入口处施加均匀的单向（水平）主体流 $v_{\mathrm{i}} = V_1\delta_{i1}$ 。顶部和底部边界采用滑移壁面，并在下游设置出口，同时采用静水压力边界。初始值取自静水压力场以及均匀的水平流场。所使用的非结构化数值网格如图 5 所示，其中标示了所采用的 P $= 96$ 个分区。该网格包含约 75 000 个控制体，圆柱形状沿圆周由 770 个表面单元离散。第一网格层的无量纲壁面法向距离为$y^+ \approx 0.01$。为捕捉涡街，在尺寸为 $[20\,D \times 5\,D]$ 的区域内部对网格进行加密，间距为 $\Delta x_1/D = \Delta x_2/D = 1/20$。时间积分采用恒定的时间增量 $\Delta t\,f = \mathrm{St}/50$ 或$\Delta t/\left(D/V_1\right) = 1/50$，即一个圆柱通过时间步长被离散为 50 个时间步。

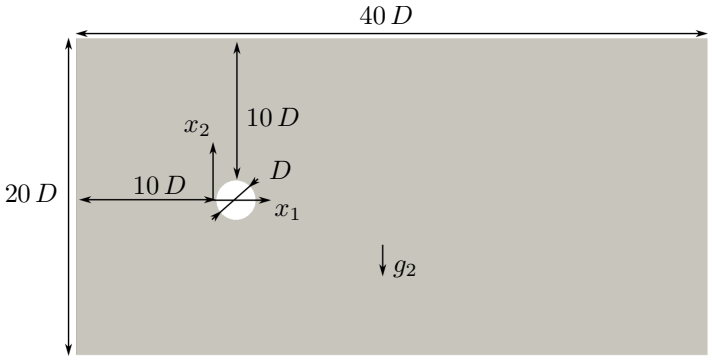
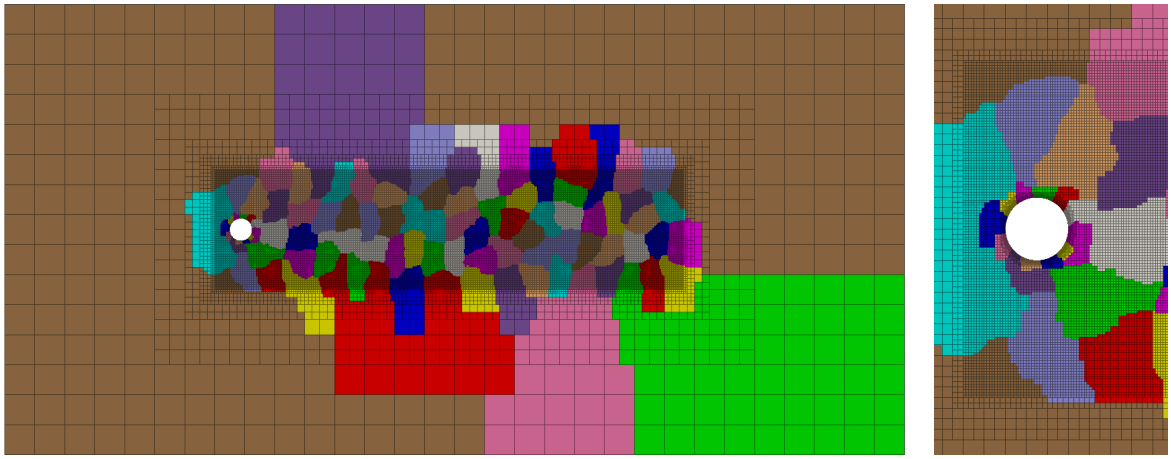


图 4：示意　　图中浸没圆柱案例配置示意图（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$　　）.



(a)　　(b)

图 5：（a）所采用的浸没圆柱案例非结构化数值网格（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$），包含 $\mathrm{P} = 96$ 个分区；（b）圆柱附近的局部放大图。

5.1 并行与串行执行

我们首先验证所提出的算法与并行化 P 及批大小 b无关。为此，从头开始对时长为 $\Delta T/(D/V_1) = 100$ 个圆柱通过时间步长的时间区间进行模拟，以建立具有涡街的周期流场。流场由三个状态变量 S 组成，即压力p、水平 v_1 和垂直 v_2 速度；公式 (1)、(20) 中的参考量设为 $\varphi^v = V_1$ 和$\varphi^p = \rho^{\mathrm{b}}\, G\, D$。由于未分区的计算域由约 $\mathrm{L} \approx 75\,000$ 个单元离散，因此全局系统矩阵的空间尺寸（长度）为 $\mathrm{N} \approx 225\,000$。时间步数描述了该矩阵的宽度，其值为 $\mathrm{T} = 5000$，即 $\boldsymbol{Y} \in \mathbb{R}^{225\,000 \times 5000}$。为评估分解的影响，我们采用$q = \mathrm{T}/10 = 500$ 个时间步，对应于十次等效的流场通过。

通过考察20种不同情形，评估算法1-4的并行化。为此，采用十种不同的空间分区方法 $\mathrm{P} = [1, 2, 3, 4, 5, 6, 12, 24, 48, 96]$，分别进行（a）完全并行和（b）串行 itSVD 构造。在后一种情况下，忽略构造过程中算法1（第9、12、20行）、Algs. 1（第3、6、11行）和算法3（第5、7行）中与并行化相关的行，以及用于评估的算法4（第2、4行），因此每种情形（b）都会进行 P 次相互独立的 itSVD 构造。根据式（5），由于完整、未截断的 rSVD 具有特征等式 $\boldsymbol{A} = \boldsymbol{U}_r \boldsymbol{S}_r \boldsymbol{V}_r^{\mathrm{T}}$，并行方法与串行方法之间预计不会有差异，图1和图2的示意图也证实了这一点。然而，如果大幅降低截断秩（即 $\tilde{q} << q$），则无法再保证重构场变量在分区边界处连续，因为保留矩阵能量 η^{q} 在局部可能存在显著差异，参见式（7）。图6和图7更详细地考察了并行方法和串行方法下的两种分区情形，即粗粒度分区（P=12；左）和细粒度分区（P=96；右）。

准确性

图6展示了时刻 $t/\mathrm{T} = 0.8$ 的多个水平速度场，其中包括（a）并行方法和（b）串行方法。图中显示了 $\tilde{q} = [1, 2, 4, 500]$ 重构秩，其数值从上到下逐渐增加。在所有并行研究中，重构场均保持平滑，不受分区方式影响。能量最强的主体模态（$\tilde{q} = 1$）似乎形成了稳态对称流。重构时加入后续三个奇异值后，则会出现涡街。比较 $\tilde{q} = 4$ 和$\tilde{q} = 500$ 时所显示的差异，通常小于 $\tilde{q} = 2$ 和 $\tilde{q} = 4$之间的变化；第6节报告的自适应性研究将进一步考察这一点。与并行框架不同，串行方法在评估秩较小（$\tilde{q} \leq 4$）时表现出对分区方式的依赖，参见图5。对于 $\tilde{q} = 2$，分区边界处的差异尤为明显。随着评估秩和保留能量增加，重构场的对齐度（平滑度）也有所改善。对于 $\tilde{q} = 500$，串行框架下的速度场也较为平滑，且似乎与并行方法一致。图7展示了全部16种情形下重构原始速度 $|\tilde{v}_1 - v_1|/V_1$ 的局部误差幅值。该图还揭示了采用串行方法时，小 $\tilde{q}$ 情况下分区方式的影响。除了对重构原始速度进行视觉比较外，图中还更详细地展示了与重构尤为相关的奇异值。图8（左）以实线表示分区#0中串行计算所得的第一（黑色）、第三（蓝色）和第五（绿色）个奇异值，并将其与并行方法所得的对应奇异值进行比较；后者以橙色符号和虚线表示。图中的横坐标表示分区方式。可以看出，并行方法所得的奇异值不受分区方式影响。相比之下，在所有串行情形中，分区规模较小时奇异值增大 [减小] ；分区规模较小 [较大] 时则会减小。细粒度分区下串行计算所得奇异值减小是预期之中的，因为空间自由度的数量——也就是每个线程的状态快照向量长度及其幅值——会随之减小，参见算法1第10行。图8（右）展示了根据式（7）预测的不同空间分解下分区#0的保留能量 η^{q} 。该图的横坐标表示截断秩 q。无论分区方式如何，所有曲线最终都收敛到 $\eta^{\mathrm{q}} \to 1$。然而，对于分区较粗[较细] 的情形，收敛开始得较晚 [较早] 。这是因为串行方法覆盖的空间区域较小。请注意，分区越粗，每个分区中的单元数越多；当 $\mathrm{P} = 1$ 时，最终与并行奇异值分解相同，这与左侧子图中的串行奇异值一致。此外，这也支持了先前的观察，即在秩较小（ $\tilde{q}$）时，保留矩阵能量显著依赖于分区方式。不过，当 $\tilde{q} \geq 20$时，这种影响会明显消失。

存储

~下面讨论相应 itSVD 构造过程结束时所占用的磁盘空间 d 。为此，测量算法1（第20—22行）的内存需求；其中，串行 itSVD 方法省略第20行，并在每个分区中除局部矩阵 $\boldsymbol{U}$ 外，还存储 $\boldsymbol{V}$ 矩阵和 $\boldsymbol{s}$ 向量。存储需求如表1所示。表中数据以完整存储方法 $\tilde{d}^{\mathrm{full}}$ 为基准进行归一化，该方法在每个时间步之后存储状态向量。并行框架始终仅需完整存储磁盘用量的7.5%，即 $\tilde{d}_{\mathrm{svd,\ par.}}/\tilde{d}^{\mathrm{full}} \approx 0.075$。相比之下，串行情况下所需的内存 $\tilde{d}_{\mathrm{svd,\ ser.}}/\tilde{d}^{\mathrm{full}}$会随着分区数量的增加而增加。该表还给出了串行与并行内存开销之比：在串行情况下该比值等于1（$\mathrm{P} = 1$），而对于$\mathrm{P} = 96$ 个分区时，该比值约增加至原来的三倍。

表 1： $q/\mathrm{T} = 1/10$（$\mathrm{Re_D} = 200$，$\mathrm{Fn} = 0.2828$，$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）单相圆柱绕流。串行和并行itSVD的内存开销 $\tilde{d}$，以完整存储方法 $\tilde{d}^{\mathrm{full}}$ 的开销为基准进行归一化；该方法在每个时间步之后存储状态向量。

分区 P [-]	96	48	24	12	6	4	2	1
$\tilde{d}^{\mathrm{svd,\ 串行.}}/\tilde{d}^{\mathrm{full}}$	0.2293	0.1512	0.1127	0.0931	0.0832	0.0799	0.0767	0.0751
$\tilde{d}^{\mathrm{svd,\ 并行.}}/\tilde{d}^{\mathrm{完整}}$	0.0751	0.0751	0.0751	0.0751	0.0751	0.0751	0.0751	0.0751
比率 [-]	3.0581	2.0183	1.498	1.238	1.108	1.065	1.021	1.0

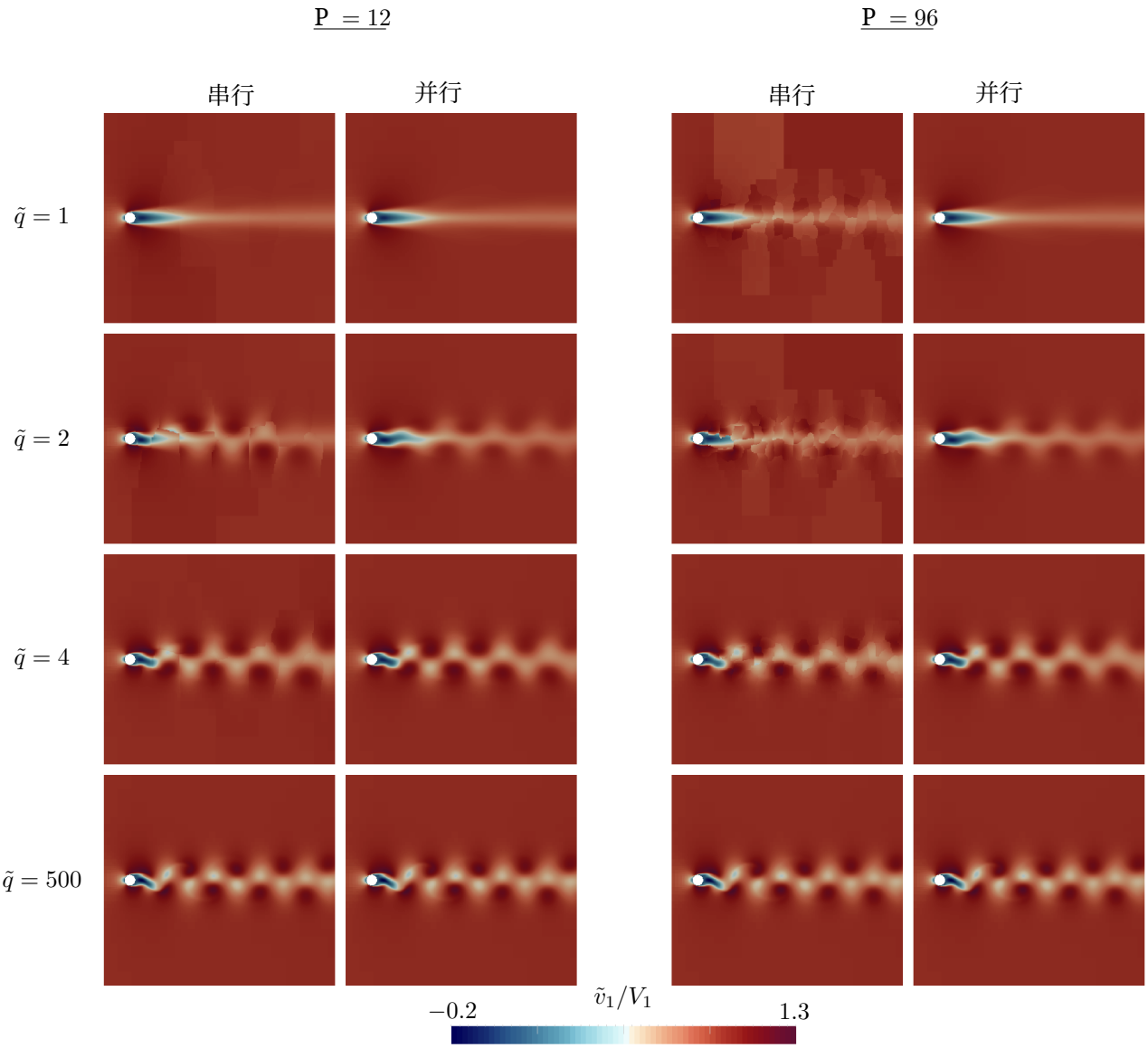


图6： $\tilde{v}_1/V_1$ 在 $q/\mathrm{T} = 1/10$ （$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）条件下单相圆柱绕流的重构原始速度。基于 Alg. 4，在不同重构秩（行）和两种划分方式（列）下，于 $t/\mathrm{T} = 0.8$ 重构流场。每项研究均采用并行方法和串行方法进行，后者忽略构造过程中的 Algs. 1（第9、12、20行）、2（第3、5、6、11行）和 Alg. 3（第5、7行）中的所有并行语句，以及评估阶段 Alg. 4（第2、4行）中的所有并行语句。

扩展性

并行 itSVD算法的可扩展性通过强扩展性测试进行评估。为此，采用包含PL $\approx 1\,000\,000$ 个控制体的更细网格；与图5所示网格相比，该网格在每个空间维度上约细化了四倍。可扩展性测试基于T $= 500$ 个时间步进行，使用 $q/\mathrm{T} = 1$ 和$b/q = 0.2$。所使用的处理器数量依次翻倍，即P $= [1, 2, 4, 8, 16, 32, 64, 128, 256]$，并根据算法2—3所需的墙钟时间确定加速比 $\tilde{t}^{\mathrm{svd}}(1)/\tilde{t}^{\mathrm{svd}}(P)$，其中 $\tilde{t}^{\mathrm{svd}}(1)$对应于串行 itSVD 构造的开销。为消除潜在读写延迟对结果的影响，每种场景的墙钟时间测量数据均取三次数值实验的平均值。所安装的硬件为每个节点配备96个 Intel Cascade Lake© Platinum 9242 处理器，并通过 Intel Omni-Path© 连接至 NHR G¨ottingen，参见 www.hlrn.de。图9展示了测得的加速比（左）和效率（右）结果。结果表明，在P $= 16$ 之前，扩展性最初呈现超线性扩展，这可能是缓存效应所致；随后直到P $= 128$时，呈现出较为理想的线性扩展。结果还显示，处理器负载上限约为

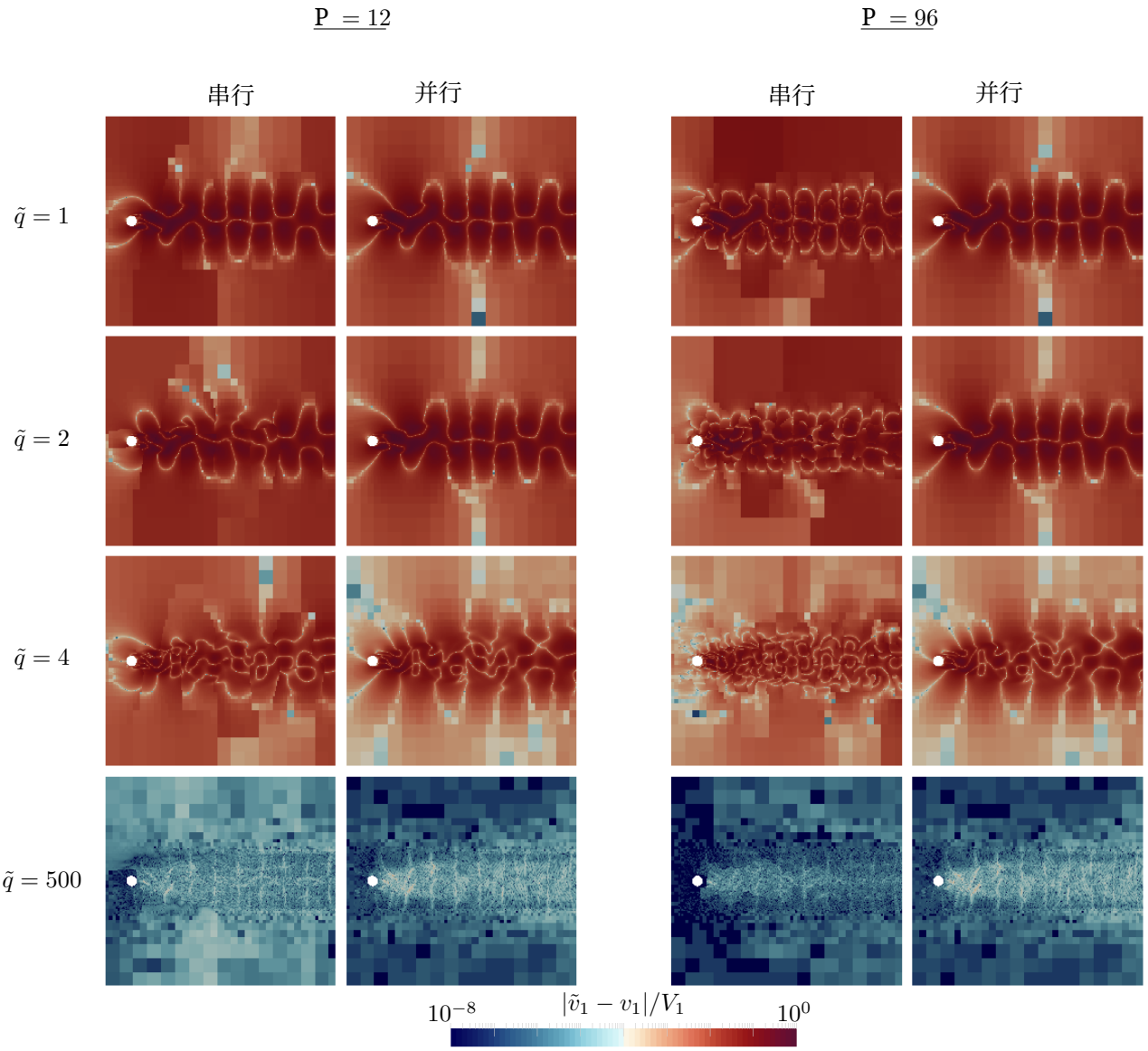


图7： $|\tilde{v}_1 - v_1|/V_1$ 单相圆柱绕流在 $q/\mathrm{T} = 1/10$ （$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）下的重构误差。基于 Alg. 4，在4种不同重构秩（行）和两种划分方式（列）下，于 $t/\mathrm{T} = 0.8$ 处的相对误差。每项研究均采用并行方法和串行方法，后者忽略构造过程中的 Algs. 1（第9、12、20行）、2（第3、5、6、11行）和 Alg. 3（第5、7行）中的所有并行语句，以及评估过程中的 Alg. 4（第2、4行）。

$\mathrm{N/P} = \mathrm{SL} \approx 15000$，在该点以下，所提出的算法达到其扩展性极限。对于不同的分组大小 b/q，也得到了类似结果。

5.2 并行itSVD的开销和准确性

本文其余部分仅讨论并行itSVD版本，本小节中的所有案例均采用图5所示的细粒度划分 $\mathrm{P} = 96$ 。第5.1节中讨论的存储开销以所需的硬盘占用 $\tilde{d}^{\mathrm{svd}}$来衡量，而 CPU 开销则通过额外的 CPU 墙钟时间 $\tilde{t}^{\mathrm{svd}}$来表征。后者还使用全存储方法所对应的数据进行归一化；该方法在构造阶段的每个时间步之后存储状态快照向量，并得到归一化参数 $\tilde{t}^{\mathrm{full}}$。需要注意的是——为了提高效率——全存储方法的输出也可以按照itSVD构造所采用的流程进行批处理。不过，这里有意不将其作为参考。对于最大构造秩 $q/\mathrm{T} = 1$，计算结果与重构结果之间的误差

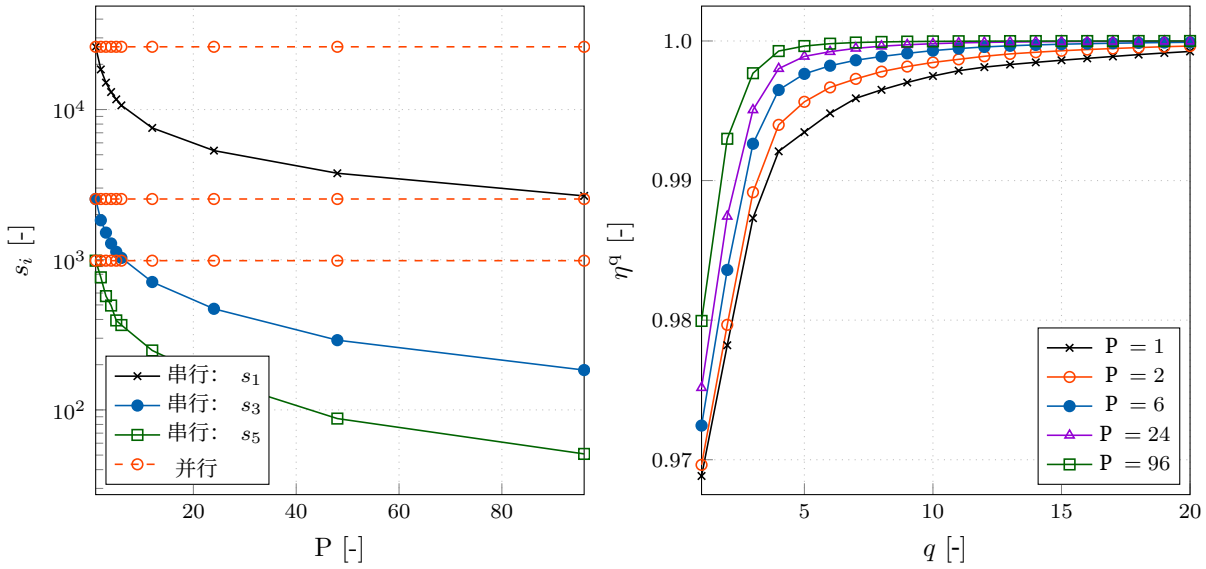


图8：具有 $q/\mathrm{T} = 1/10$ 的单相圆柱绕流（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）。左：针对不同空间划分方式，串行确定的第一阶（黑色）、第三阶（蓝色）和第五阶（绿色）奇异值的表示。并行方法对应的奇异值与划分方式无关，如水平橙色线所示。右：对于不同空间分区，在分区#0上前20个模态根据公式（7）得到的保留能量 η^q 。

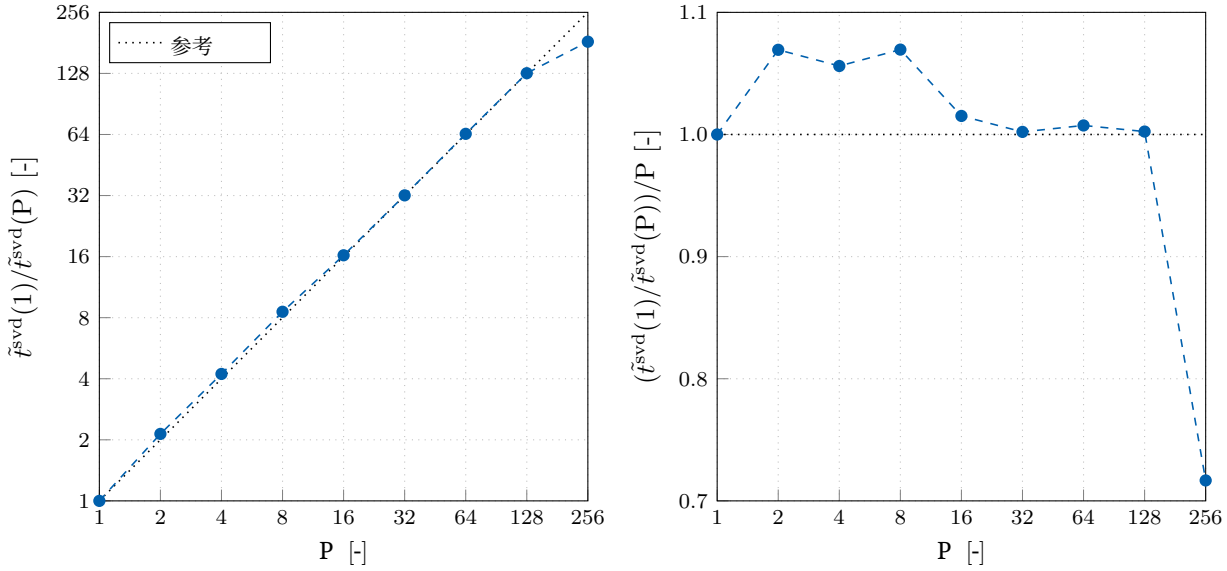


图9：具有 $q/\mathrm{T} = 1$ 和$b/q = 0.2$ 的单相圆柱绕流（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 500$）的并行itSVD方法可扩展性测试。细网格（$\mathrm{N} \approx 3\,000\,000$）下itSVD算法的加速比（左）和效率（右）。给出的数值对应于每种场景运行三次所得的平均值。

状态应在 $|\tilde{\varphi} - \varphi|/\varphi^{\mathrm{ref}} \to 0\,\mathrm{T}$ $\tilde{q} \to q =$时消失。为验证这一点，持续增大上一小节中的构造秩 $q = 500$，即 $\tilde{q}/q = 1$ 取 $q/\mathrm{T} = [1/50, 1/10, 1/5, 1/2, 1]$。此外，在每种情况下还考察五种不同的批大小 $b/q = [1, 1/5, 1/10, 1/25, 1/50]$，共得到25项研究。未截断的一次性随机奇异值分解指的是 $b/q = 1$ 以及 $q/\mathrm{T} = 1$，仍可在适中的

当前二维圆柱绕流的开销。

图10和图11左侧分别展示了计算得到的阻力（c_d）和升力（c_l）系数的变化历史。经过约4000个时间步 t 后，流场达到周期状态。图10和图11中间的图放大显示了时间区间 $3000 \leq t \leq 4000$ ，并对比了重构流场（符号， $\tilde{c}$）与原始流场（曲线， c）得到的力数据。两个力系数 $\tilde{c}_\mathrm{d}$， $\tilde{c}_\mathrm{l}$ 均由重构流场使用批量大小为 $b/q = 1$ 、 $\tilde{q}/\mathrm{T} = 1$的设置计算得到。重构力与原始数据及其完整存储对应结果均吻合。图10和图11右侧的图展示了误差历史及其对应的时间平均值（以水平线标出），进一步支持了这一结论。完整存储方法的误差以蓝色表示。所有误差无一例外地都相当小：阻力系数的误差低于 10^{-9} ，升力系数的误差低于 10^{-11} 。

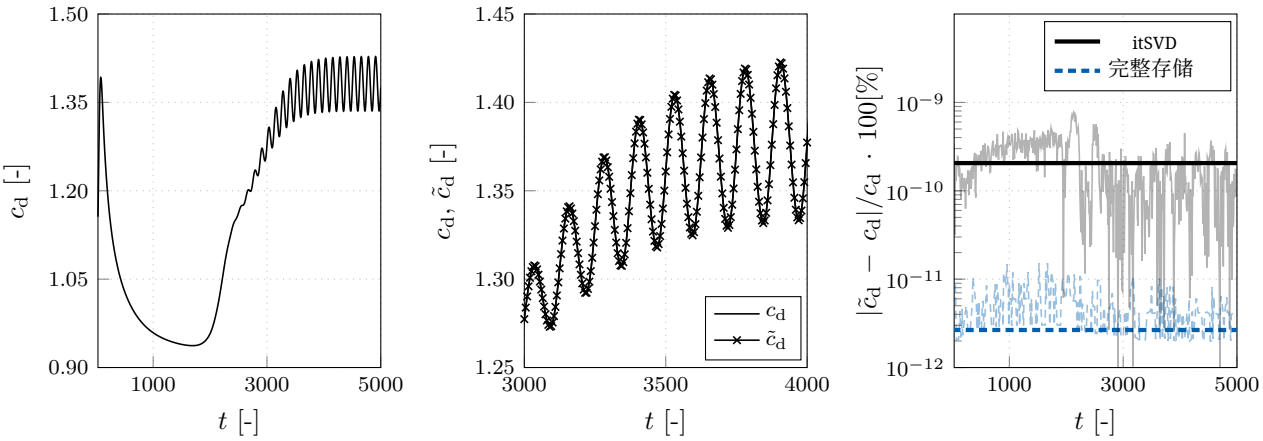


图 10： $q/\mathrm{T} = 1 = \tilde{q}/\mathrm{T}$ 的单相圆柱绕流（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）。左：计算得到的阻力系数 c_d 的时间演化。中：时间步区间$3000 \leq t \leq 4000$ 的放大图，并附有根据重构流场得到的阻力系数 $\tilde{c}_\mathrm{d}$ 。右：两者之间的相对误差（黑色），以及完整存储方法返回的误差（蓝色）和相应的平均值（水平线）。所有仿真均采用 P=96 个分区并行进行。

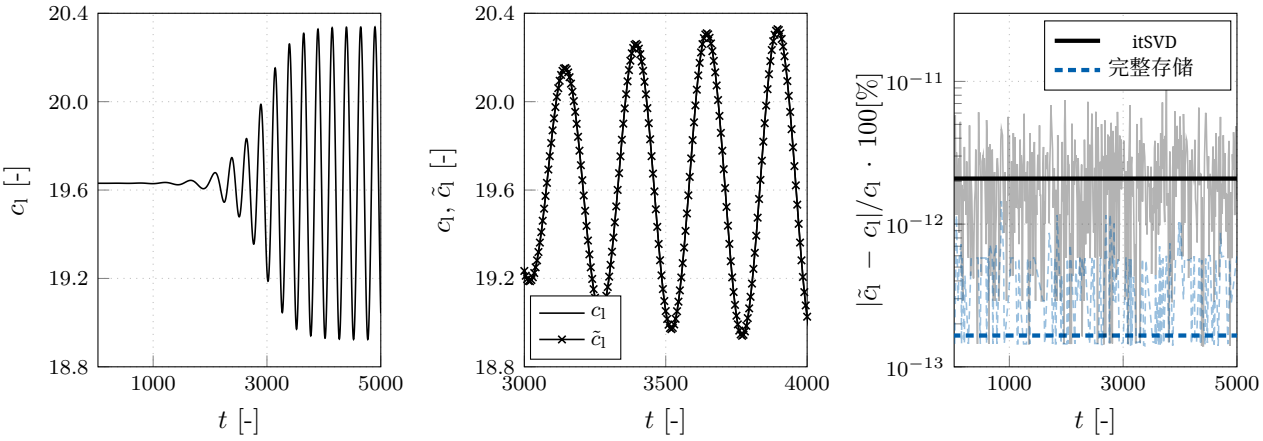


图 11： $q/\mathrm{T} = 1 = \tilde{q}/\mathrm{T}$ 的单相圆柱绕流（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）。左：升力系数 c_l 的时间演化。中：时间步区间 $3000 \leq t \leq 4000$supplemented with 由重构流场获得的升力系数 $\tilde{c}_\mathrm{l}$ 的放大图。右：两者之间的相对误差以黑色表示，完整存储方法返回的误差以蓝色表示，并给出相应的平均值（水平线）。所有仿真均使用 P=96 个分区并行执行。

对于 $q/\mathrm{T} = 1 = \tilde{q}/\mathrm{T}$，图10和图11所示的准确性不会随批大小而改变；也就是说，$b/q = [1/5, 1/10, 1/25, 1/50]$ 得到的结果几乎完全相同。虽然批大小 b 不会影响结果质量和存储开销，却会显著影响 itSVD 构建时间 $\tilde{t}^{\mathrm{svd}}$。表2列出了所有考察情况的归一化墙钟开销（$\tilde{t}^{\mathrm{svd}}/\tilde{t}_{\text{完整}}$）。总体而言，无论构造秩如何，减小 b都会增加 itSVD 构建时间。这是预期之中的，因为较小的 b 需要进行更多次 itSVD 更新，参见算法1。当最大秩 $q/\mathrm{T} = 1$ 与最小$b/q = 1/50$结合时，构建时间的增幅尤其显著（$\tilde{t}^{svd}/\tilde{t}^{\mathrm{full}} \approx 94$）；其他作者在类似应用中也观察到了这一现象，参见 Vezyris等（2019）；Margetis等（2021、2022）。与此同时，itSVD 构建时间自然会随 q/T减小而缩短；当 $q/\mathrm{T} = 1/10$时，墙钟开销降至完整存储开销的约132%。对于所考察的圆柱绕流，只有在显著降低截断秩（$q/\mathrm{T} > 1/10$）且同时采用较大的批次大小（$b/q \leq 1/10$）时，itSVD 才优于全存储方法。

表2：单相圆柱绕流（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）。不同重构秩 $q/\mathrm{T} = [1, 1/2, 1/5, 1/10, 1/50]$ 和分组大小 $b/q = [1, 1/5, 1/10, 1/25, 1/50]$ 下测得的归一化墙钟时间 $\tilde{t}^{\mathrm{svd}}$（参考数据对应于完整存储墙钟时间 $\tilde{t}^{\mathrm{full}}$）。给定数值为使用 P=96 个分区进行并行计算时，每种情景运行三次所得的平均值。

q/T	$\frac{\tilde{t}^{\text{奇异值分解}}}{\tilde{t}^{\text{完整}}}\left(\frac{b}{q}=1\right)$	$\frac{\tilde{t}^{\text{奇异值分解}}}{\tilde{t}^{\text{完整}}}\left(\frac{b}{q}=\frac{1}{5}\right)$	$\frac{\tilde{t}^{\text{奇异值分解}}}{\tilde{t}^{\text{完整}}}\left(\frac{b}{q}=\frac{1}{10}\right)$	$\frac{\tilde{t}^{\text{奇异值分解}}}{\tilde{t}^{\text{完整}}}\left(\frac{b}{q}=\frac{1}{25}\right)$	$\frac{\tilde{t}^{\text{奇异值分解}}}{\tilde{t}^{\text{完整}}}\left(\frac{b}{q}=\frac{1}{50}\right)$
1	10.941	13.796	22.369	49.3725	94.0163
1/2	9.0980	11.487	17.400	38.0972	75.3642
1/5	3.0029	4.1213	6.6093	14.5674	27.7240
1/10	1.3224	1.9041	3.0843	6.70971	12.9726
1/50	0.2047	0.2715	0.4372	0.93983	1.83124

表3概述了所考虑的5种构造秩所使用的归一化磁盘空间，该空间与所采用的批大小 b 无关。正如预期，所需内存大致随 itSVD 的秩缩放；例如，当降幅达到十倍时，$q/\mathrm{T} = 1/10$ 仅需约 7.5% 的完整存储内存，尽管该秩下的结果与完整存储数据吻合良好，参见图6和图7的最下行。

表3：单相圆柱绕流（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）。归一化内存开销 $\tilde{d}^{\mathrm{svd}}$随重构秩的扩展性（对于 $q/\mathrm{T} = [1, 1/2, 1/5, 1/10, 1/50]$，参考数据指完整存储方法 $\tilde{d}_{\mathrm{full}}$的磁盘使用量）。

q/T	1	1/2	1/5	1/10	1/50
$\tilde{d}^{\text{奇异值分解}}/\tilde{d}^{\text{完整}}$	0.998	0.375	0.152	0.0751	0.0169

最后，对所研究的5个截断秩 q 的准确性影响进行了评估。图12展示了重构阻力（左）和升力（右）系数相应的平均值（黑色）、最大值（橙色）和最小值（蓝色）相对误差。全存储方法对应的误差以水平虚线表示。正如预期，随着秩的降低，由于保留矩阵能量减少，误差会增大。后者由图13（上部）中前40个奇异值对应的黑线表示，表明当考虑4个或更多奇异值时，保留的矩阵能量超过99%。如图中下部所示，第一个奇异值处于 $\mathcal{O}(10^4)$ 范围内，并在超过 $q/\mathrm{T} = 0.1$ 后迅速降低约5个数量级，参见图13下部。请注意，这些数据是从头生成的，并且初始瞬态带来了显著的（不利）影响，如图13中黑色曲线与橙色曲线之间的差异所示。

橙色曲线表示使用重启机制为时间步 $5000 \leq t \leq 10000$ 编制的模拟数据。在这种情况下，不会出现初始瞬态，保留能量在大约七个模态后趋于收敛，参见图 13（底部）。奇异值比通过重构带有初始瞬态效应的仿真结果所获得的相应值低一个数量级。图 12 的误差分析针对完全周期情形进行了重复，如图 14 所示。升力值可以说不太容易受到均匀

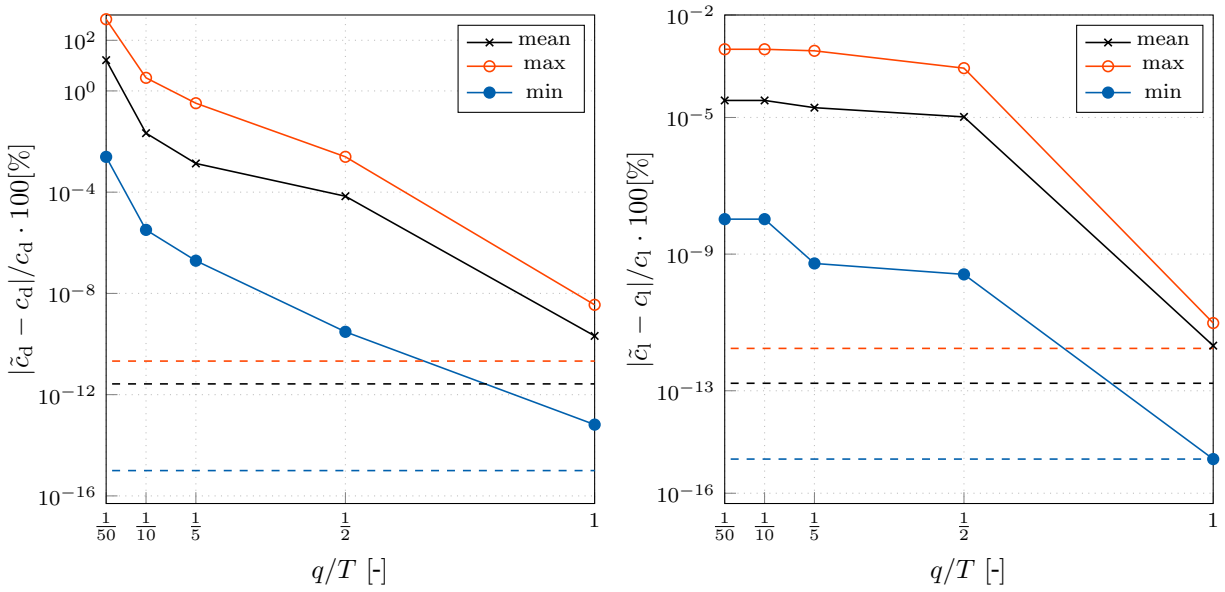


图12： $q/\mathrm{T} = [0.02, 0.1, 0.2, 0.5, 1]$ （$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）下的单相圆柱绕流。左 [右]：重构阻力[升力] 系数 $\tilde{c}_\mathrm{d}$ [$\tilde{c}_\mathrm{l}$] 的平均值（黑色）、最大值（橙色）和最小值（蓝色）相对误差，其中采用了 itSVD 降阶（$1 \leq t \leq \mathrm{T}$）。

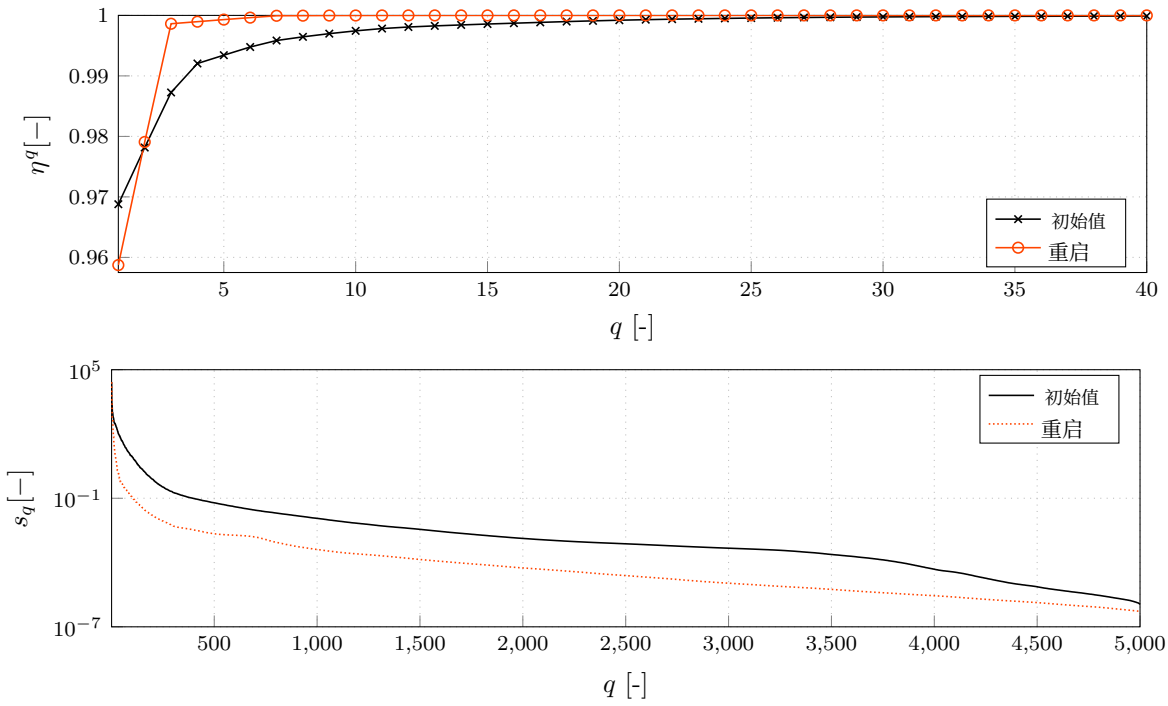


图13： $q/\mathrm{T} = 1$ （$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} = 5000$）下的单相圆柱绕流。上图：按降序排列的全部奇异值。下图： $1 \leq t \leq \mathrm{T}$（黑色）和 $\mathrm{T} \leq t \leq 2\mathrm{T}$（橙色）对应的保留能量。

流动初始化的影响，且升力系数的误差行为非常相似。相比之下，只有在重构值较大时，阻力系数误差的量级才相近。

总结而言 一致的并行实现可能至关重要，尤其是对于低秩近似而言。秩会线性影响磁盘使用量，并以非线性方式影响准确性。然而，

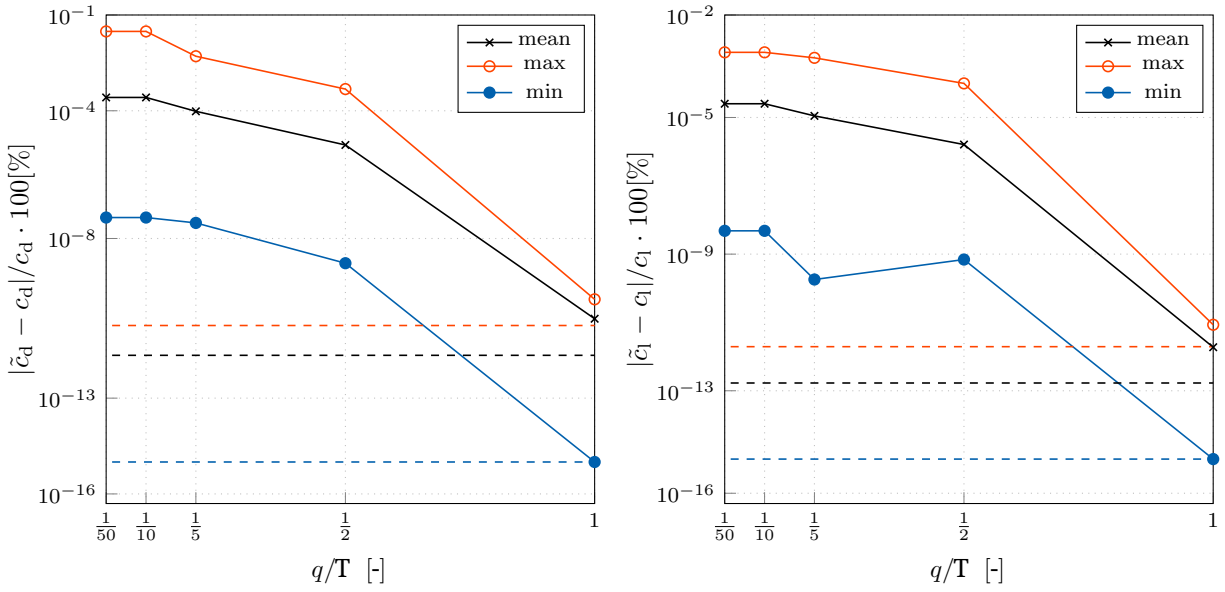


图14： $q/\mathrm{T} = [0.02, 0.1, 0.2, 0.5, 1]$ 的单相圆柱绕流（$\mathrm{Re_D} = 200$、$\mathrm{Fn} = 0.2828$、$\mathrm{St} = 0.2$，$\mathrm{T} =$5000）。左 [右]：重构阻力[升力] 系数 $\tilde{c}_\mathrm{d}$ [$\tilde{c}_\mathrm{l}$] 的平均值（黑色）、最大值（橙色）和最小值（蓝色）相对误差，以及重启模拟从$\mathrm{T} \leq t \leq 2\mathrm{T}$进行itSVD降阶的结果。

准确性要求取决于计算物理量。因此，保留能量提供了一种合理的质量衡量指标。出于效率方面的考虑，批次大小应选择得足够大，同时在设计区域分解时必须考虑可分配内存的最大值。最后，引入自适应元素可以实现itSVD过程的自动化。下一节将研究这些方面。

6 自适应性与可实现性

本节介绍并行itSVD的自适应实现，旨在确定合适的秩和批大小。此外，还评估了重构场的可实现性。为此，研究了 5° 入射角下浸没式NACA0012水翼周围的时变二维湍流两相流。该案例参考了Duncan（1981年、1983年）开展的实验，如图15所示。与实验不同，仿真并非在（稳态）静水条件下进行，而是在周期波上进行；同时，雷诺数提高到符合完全湍流条件。水翼前缘处的弦长L^c 至浸没深度 S^w 之比为 $L^c/S^w = 7/9$。仿真基于重力加速度 $G = |g_2|$、来流速度 $V_1 = |v_1|$ 以及水相的密度 ρ^b 和动力黏度 μ^b ，在$\mathrm{Re} = \rho^\mathrm{b}\, V_1\, L^c/\mu^\mathrm{b} = 750\,000$和$\mathrm{Fn} = V_1/\sqrt{G\, S^w} = 0.62$条件下进行。入射波的长度、高度和周期分别为 $L^w/L^c = 4$、 $H^w/L^c = 1/10$ 和 $T^w (V_1/L^c) = 4$。预计浸没式水翼将在水翼下游诱导出长度为 $\lambda^c/S^w = 2\,\pi\, \mathrm{Fn}^2 = 2.416$ 的波场。

计算网格由约530 000个控制体组成，并划分为$\mathrm{P} = 288$个分区，参见图16。其在水平（x_1）方向延伸 $34L^c$ ，在垂直方向（x_2）延伸 $23L^c$ 。自由表面采用尺寸为 $\Delta x_1/\lambda^c = 1/400$ 和 $\Delta x_2/H^w = 1/50$的单元进行加密，加密区域位于 $-2.75\,(H^w/2) \leq x_2 - S^w \leq 4\,(H^w/2)$ 之间，其中原点位于水翼的前缘。为满足随时间保持不变的静水出口条件，水平分辨率朝出口方向逐渐降低。水翼前缘分别距离入口和底部 $11L^c$ 和 $11.5L^c$ 。完全湍流仿真采用壁面函数，其中第一网格层沿壁面法向的厚度为 $y^+ \approx \mathcal{O}(50)$ 。在入口处，将线性波场叠加到水平主体流中，同时设置湍流强度为 $I = 5\%$和归一化涡黏度为 $\mu^\mathrm{t}/\mu = 1$。顶部/底部边界采用滑移壁面，出口处指定静水压力边界。与单相流相比，采用更小的时间步，以满足压缩近似格式的稳定性条件，参见Manzke等（2012）；K¨uhl等（2021b），流场随时间推进，时间步为

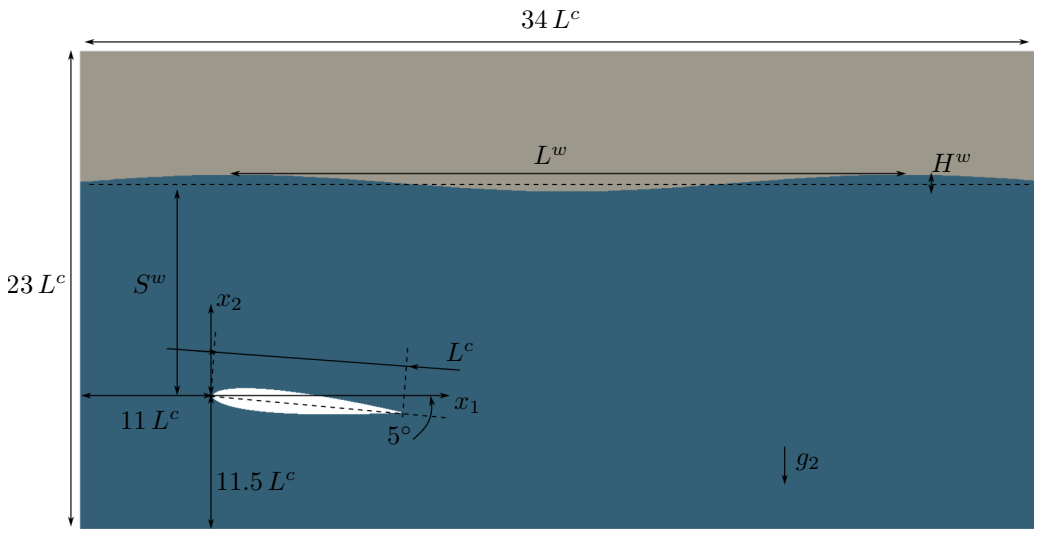
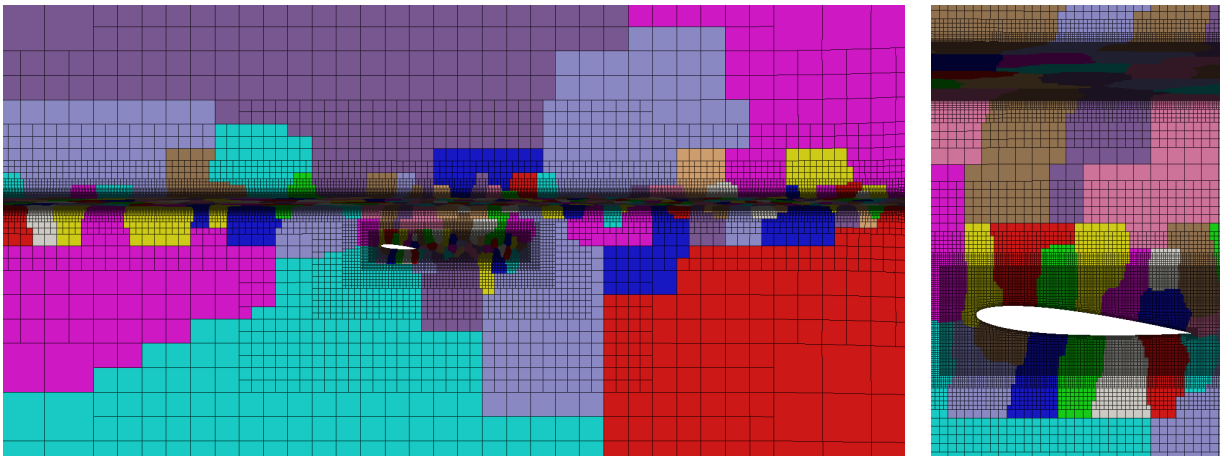


图15：二维水翼周围两相流的初始配置示意图（$\mathrm{Re_L} = 750\,000$，$\mathrm{Fn} = 0.62$，$L^w/L^c = 8$，$H^w/L^c = 0.1$，T^w（V_1/L^c）$= 4$，$\mathrm{T} = 10000$）。



(a) (b)

图16：二维水翼周围两相流所采用的非结构网格（$\mathrm{Re_L} = 750\,000$，$\mathrm{Fn} = 0.62$，$L^w/L^c = 8$，$H^w/L^c = 0.1$，T^w（V_1/L^c）$= 4$，$\mathrm{T} = 10000$）。划分为288个分区（左）以及水翼区域的放大图（右）。

时间步 $\Delta t\, V_1/L^c = 0.01$（$\Delta t/T^W = 0.0025$）。

6.1 截断秩的自适应估计

保留的奇异值数量对itSVD的准确性和计算成本都至关重要。因此，自适应秩是可取的，本小节将使用$\mathrm{T} = 10\,000$ 个时间步对其进行评估。

流场由 $\mathrm{S} = 6$ 个状态变量组成。相浓度为无量纲并经过归一化处理。根据公式（1），其他场量分别采用 $\varphi^v = V_1$、$\varphi^p = \rho\, G\, L^c$、$\varphi^k = u_\tau^2$和$\varphi^\varepsilon = u_\tau^4$ 对速度、压力、湍动能及其耗散进行归一化。摩擦速度u_τ 依据经验平板关系得到，即 $u_\tau = \sqrt{\tau_\mathrm{w}/\rho}$，其中 $\tau_\mathrm{w} = c_\mathrm{f}\rho V_1^2/2$~~和 $c_\mathrm{f} = 0.026/\mathrm{Re}^{(1/7)}$~~。与之前的层流单相示例相比，离散化工作量有所增加。因此，全局矩阵长度为 $\mathrm{N} = \mathrm{Y\,P} = (\mathrm{S\,L})\,\mathrm{P} \approx 3\,180\,000$，全局状态矩阵规模为 $\boldsymbol{Y} \in \mathbb{R}^{3\,180\,000\times 10\,000}$。

首先，执行了 8 种具有不同秩 $q/\mathrm{T} = [0.001, 0.005, 0.01, 0.025, 0.05, 0.1, 0.2, 0.4]$ 的非自适应增量截断奇异值分解构造，并将其作为基准。由于复杂度增加以及先前的研究结果，仅确定所有可能奇异值的 0.1%–40%。将评估秩设置为相应的构造秩，即 $q/$，所得重构预测的阻力

和升力系数的平均值（黑色）、最大值（橙色）和最小值（蓝色）误差，在图 17 左侧 [右侧]中以标记的实线表示。与第 5 节的圆柱研究类似，平均误差位于由最小值和最大值误差界定的范围内，并且随着秩的增加而降低约 3 个数量级。其次，为进一步研究构造秩（q）与评估秩（$\tilde{q}$）之间的相互作用，对后者进行变化

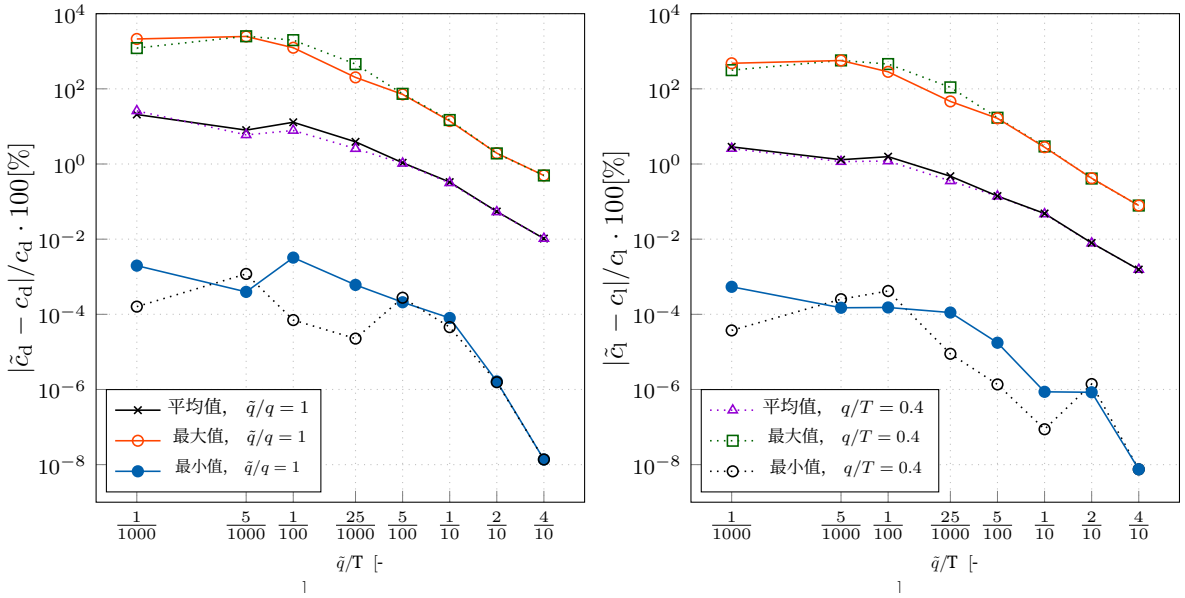


图 17：对于具有 $\tilde{q}$/T $= [0.001, 0.005, 0.01, 0.025, 0.05, 0.1, 0.2, 0.4]$（$\mathrm{Re_L} = 750\,000$、Fn $= 0.62$、 $L^w/L^c = 8$、$H^w/L^c = 0.1$、 T^w（V_1/L^c） $= 4$）的二维水翼周围两相流，重构阻力（左）和升力（右）的平均值（黑色）、最大值（橙色）和最小值（蓝色）相对误差。实线表示 $\tilde{q}/q = q$/T 调查结果。虚线标记 $\tilde{q}$/T $\leq q$/T、 q/T $= 0.4$ 研究的结果。

而前者被赋予其最大值，即 $\tilde{q}$/T $\leq q$/T，且 q/T $= 0.4$。所得的平均值（紫色）、最大值（绿色）和最小值（黑色）误差以点线形式添加到先前的结果中。这些误差与 $\tilde{q}/q = 1$ 研究的误差略有偏差，而平均结果的一致性最佳。

图18（左）展示了保留能量随构造秩 q/T 的变化，该变化由公式（7）描述。图中表明，前十个模态占据主导地位，包含约99%的总能量。考虑更多模态——因此在固定 q/T $= 0.4$的情况下也会增加 $\tilde{q}$/T——被认为对总体结果影响较小。除规定固定构造秩外，还针对自适应秩和 $\tilde{q}/q = 1$开展了四项研究。它们的区别在于最小构造秩 o/T $\cdot 100 = [0.01, 0.1, 1, 2.5]$，并分别与13种不同规定的保留矩阵能级 $\eta^o \cdot 100 = [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 99]$,结合使用，参见公式（8）和算法 5。这些研究的结果如图19所示，分别对应 $o = 1$ （黑色）、 $o = 10$ （橙色）、 $o = 100$ （蓝色）和 $o = 250$ （紫色）。所示数据指的是升力和阻力系数平均误差。提高最小构造秩通常会降低误差，并且对阻力和升力表现出定性相似的行为。在几乎所有情形下，提高保留能量也会降低误差，但 $o = 1$情形除外，参见公式（7、8）。对于自适应方法的这一经典起点，误差在几乎整个能量谱范围内保持不变，这归因于前面提到的第一个模态（或前几个模态）极高的能量贡献，参见图18（左）。因此，只有在对保留能量提出非常高的要求时，该情形才会进入自适应过程；图18右图也证实了这一点。其中，针对每个指定的保留能量值测量所得的构造秩。要求越宽松，所得秩越接近规定的最小值（$o = 1$ 黑色、 $o = 10$橙色、 $o = 100$ 蓝色和 $o = 250$ 紫色）；只有当达到 $\eta^{o=1} \geq 95\%$ 时，秩才会从 $q = 1$ 增加到 $q = 4$。图19（左）中的阻力系数也呈现出类似现象：在较低的保留能级下，$\eta^o \cdot 100 \leq 10$ %时的误差有时低于 $o = 10$，参见 $o = 1$ 。这归因于所采用的误差评估方式，该方式仅限于全局平均偏差。为便于说明，图18（中）将高质量的阻力重构（蓝色）与两种中等计算开销的重构（$\eta^o \cdot 100 = 1\%$）进行了比较，后两者分别使用 $o = 1$（黑色）和 $o = 10$ （橙色）。对于 $o = 1$，近乎恒定的模态会产生显著更小的偏差。对于 $o = 10$，至少还会考虑九个模态，从而引发更高的偏差和平均误差。对于

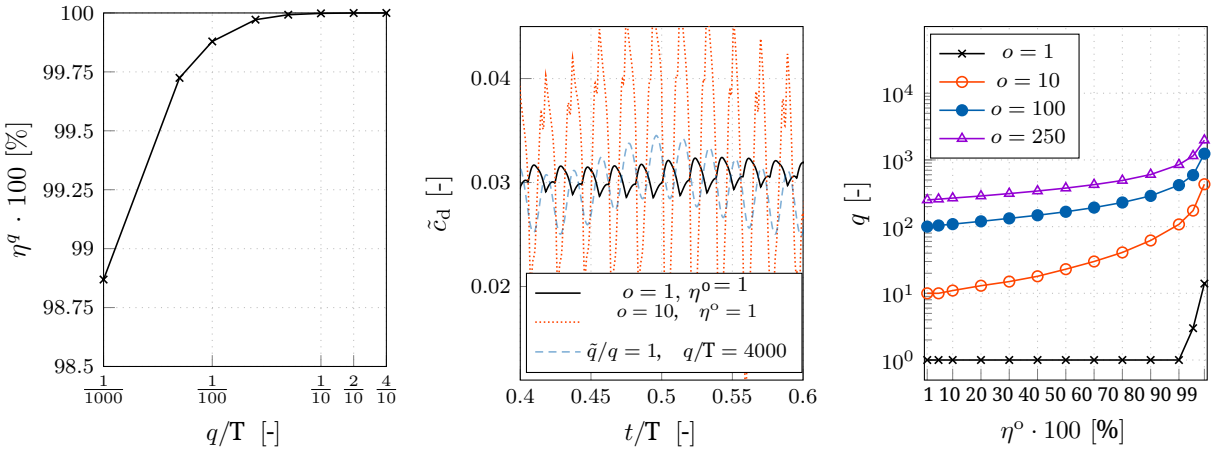


图18：两相水翼流动（$\mathrm{Re_L} = 750\,000$，$\mathrm{Fn} = 0.62$、$L^w/L^c = 8$、$H^w/L^c = 0.1$、T^w（V_1/L^c）=4）。左：保留能量 η^q 与构造秩 q/T的关系。中：由不同重构得到的阻力系数。右：对于4个最小秩，根据保留矩阵能量 η^o 估计所需构造秩。

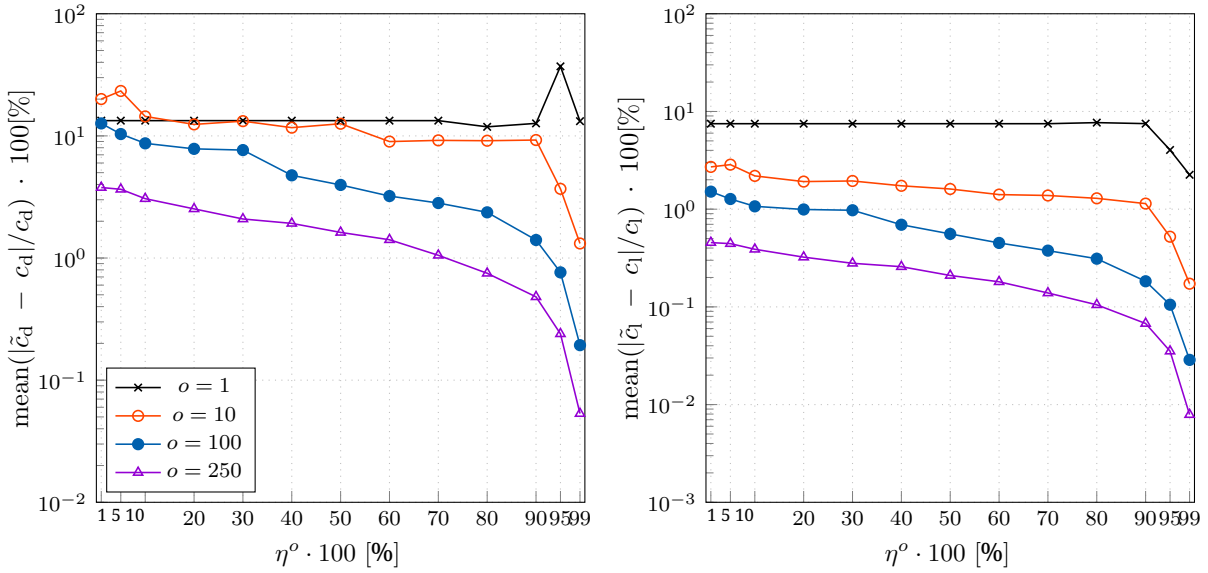


图 19：采用具有不同最小构造秩 $\eta^o \cdot 100 = [1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 99]$ 的自适应 itSVD，对二维水翼周围两相流（$\mathrm{Re_L} = 750\,000$、$\mathrm{Fn} = 0.62$, $L^w/L^c = 8$, $H^w/L^c = 0.1$, $T^w(V_1/L^c) = 4$）重构阻力（左）和升力（右）的平均相对误差，其中 $o = 1$（黑色）、$o = 10$（橙色）、$o = 100$（蓝色）、$o = 250$（紫色）。

本文所考虑的流动而言，此类现象仅限于前 $\mathcal{O}(10^1)$ 个模态，而构造秩的显著增加会使平均近似误差持续降低。

由此得出，自适应 itSVD 程序通过指定最小构造秩得到了显著改进。然而，最小秩 o 的确定依赖于经验判断，例如可依据第 2 节中介绍的启发式方法。

6.2 批大小的估计

批大小 b 应在满足需要的前提下尽可能小，同时尽可能大。理想情况下，它将接近单次处理方法中的最大时间步数，即 $b \to \mathrm{T}$。通过评估硬件特定的可用内存，可以估算出合理的批大小 b。所需内存的近似值为

根据主导行大小 N 确定，参见公式（1）。因此，在 itSVD 构造开始阶段，可以通过逐步增大批大小 b 来模拟 itSVD 更新，直至达到最大可分配内存限制。需要注意的是，除了批次矩阵 $\boldsymbol{B}$之外，在更新过程中还必须分配其他全局矩阵，其中一些矩阵长度相同（例如， $\boldsymbol{P}$、 $\boldsymbol{Q}$ ，见算法2），批大小估计器应将此因素纳入考虑。例如，本节中的二维流动在 N $\approx 3\,180\,000$ 和 $b = 500$ 的情况下，以 8 字节双精度环境存储时（可能采用分布式存储），其结果为 (N $\times$b $\times$ 8)/$10^9 \approx 12.72$ 吉字节。最后，在估算出的最大允许 $b_{\max}$ 值上施加安全系数，以确保在当前 SIMD概念下仿真的稳定性；本文始终将该系数设为 $\approx$ 75%（即 $b = \mathcal{O}(0.75\, b_{\max})$）。

6.3 重构场的可实现性

低秩近似通过依赖快照的约化基构造线性组合。它们继承了底层快照的性质，例如，对于与 $\partial v_k/\partial x_k = 0$一致的快照，无散速度场通常能够得到良好保持。然而，阈值的物理约束（例如有界体积分数 $c \in [0,1]$ 或固有的正湍动能 $k \geq 0$）不一定能够得到遵守。图 20 展示了在时刻 $t/\mathrm{T} = 0.5$ ，使用具有 $o = 10$ 和 $\eta^o = 90\%$（$q = 108 = \tilde{q}$）的自适应 itSVD 算例得到的重构浓度场 $\tilde{c}$ （上图）所对应的波形，以及归一化湍动能 $\tilde{k}/u_\tau^2$（TKE，下图）。非物理负值区域以透明绿色突出显示。

在“层流”区域会出现略微为负的 TKE 值，但在翼型附近的高剪切率区域、其尾流和自由表面处，没有观察到伪空洞区域的污染。同样，不可实现的浓度值似乎也得到了有利的“组织”，即过低的值嵌入水相中，而过高的值出现在气相内部。后者在剧烈流动中可能更加棘手，尤其是因为超出可实现浓度区间会导致流体性质完全错误，例如出现负的流体密度。

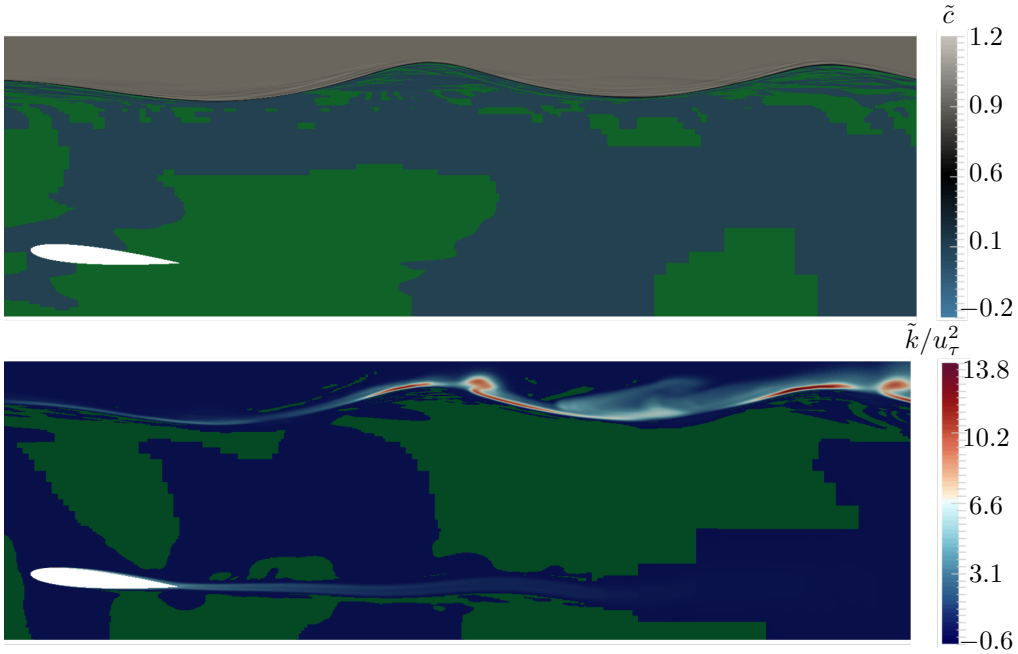


图 20：时刻 $t/\mathrm{T} = 0.5$ （$\mathrm{Re_L} = 750\,000$、Fn $= 0.62$、 $L^w/L^c = 8$、 $H^w/L^c = 0.1$、 $T^w V_1/L^c = 4$）二维水翼周围重构的两相流。浓度场（上）和归一化湍动能（下），其中绿色区域表示非物理负值区域。

为恢复重构场的可实现性，需纳入相关要求。通常会有意抑制或截断负能量或浓度值，即：

$$\tilde{c} = \max(\alpha, \min(\tilde{c}, 1)), \qquad \text{and} \qquad \tilde{k} = \max(\alpha, \tilde{k}), \quad \tilde{\varepsilon} = \max(\alpha, \tilde{\varepsilon})\,, \tag{30}$$

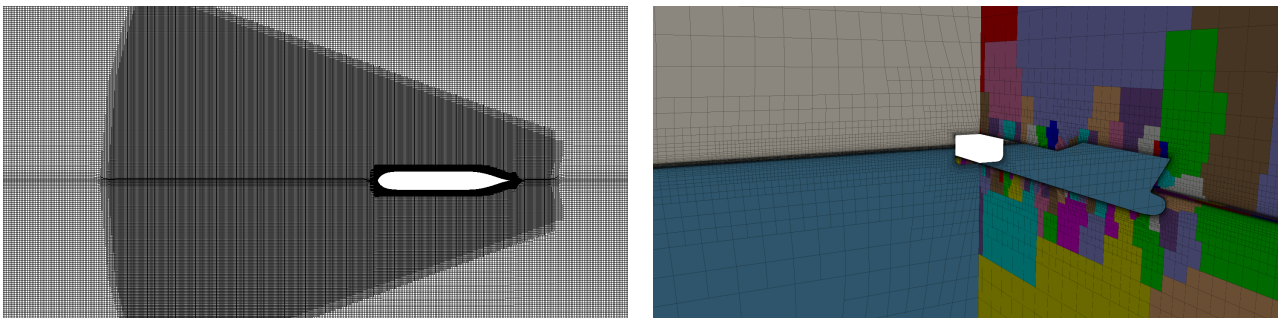
其中， $\alpha = 10^{-16}$ 指的是可表示的最小值，以避免数值冲突。请注意，对于可实现和不可实现的场重构，仅观察到积分阻力和升力系数存在细微差异。

7 应用

本应用案例研究了三个具有实际意义的问题：首先，再次考察模型降阶对全局积分数据（如船舶总阻力）和局部量（此处为波面升高）的影响。此外，还分析了仿真过程中因 itSVD 构造产生的计算开销，以识别所述算法中的潜在瓶颈。

三维应用涉及谐和迎浪中无附体 Kriso集装箱船（KCS）船体周围的完全湍流。所研究的 1:31.6 比例模型提供了大量在静水条件下获得的数值和实验数据，例如 Banks 等（2010年）；Larsson 等（2013年）；Kr¨oger 等（2018年），以及迎浪条件下的数据，例如 Carrica 等（2011年）；Simonsen 等（2013年）；Shen 等（2015年）。船体模型艉垂线与艏垂线之间的距离作为参考长度 $L = L^{\mathrm{pp}}$。其他参考属性包括重力加速度 $G = |g_2|$、来流速度大小 $V_1 = |v_1|$ 以及水的运动黏度 $\nu^{\mathrm{b}} = \mu^{\mathrm{b}}/\rho^{\mathrm{b}}$。计算在雷诺数和弗劳德数 $\mathrm{Re} = V_1 L/\nu^{\mathrm{b}} = 1.4 \cdot 10^7$ 和 $Fn = V_1/\sqrt{GL} = 0.26$下进行。仿真过程中，船舶运动和推进均受到抑制，初始吃水为 $d^{\mathrm{h}}/L^{\mathrm{pp}} = 0.04696$。迎浪的长度、高度和周期分别为 $L^{\mathrm{w}}/L^{\mathrm{pp}} = 1.3$、 $H^{\mathrm{w}}/L^{\mathrm{pp}} = 0.075$ 和 T^{w} （V_1/L^{pp}） $= 1.3$。除入射波外，预计船体还将诱导产生长度为 $\lambda^{\mathrm{w}}/L^{\mathrm{pp}} = 2\,\pi\,\mathrm{Fn}^2 = 0.4247$ 的波场。

所采用的数值网格由约 3,000 万个非结构计算单元组成，如图 21 所示。计算域在水平（x_1）、横向（x_2）和垂直（x_3）方向分别延伸至 $8\,L^{\mathrm{pp}}$、 $6\,L^{\mathrm{pp}}$ 和 $5\,L^{\mathrm{pp}}$。整个流场均得到解析，即未施加对称条件，且入口（左舷） [下方] 边界位于 $x_1/L^{\mathrm{pp}} = 4$（$x_2/L^{\mathrm{pp}} = 3$） $[x_3/L^{\mathrm{pp}} = 3]$。船体表面采用约 595,000 个表面单元进行离散。法向壁面分辨率对应于无量纲壁面距离$y^+ = \mathcal{O}$（50），因此可以使用壁面函数。自由表面区域的垂向分辨率在整个计算域内保持不变，以便用 20 个单元解析给定的波幅 $2\,H^{\mathrm{w}}$，即 $\Delta x_2/H^{\mathrm{w}} = 1/20$。自由表面的总体切向分辨率由 $\Delta x_1/L^{\mathrm{w}} = 1/40$ 解析。在 Kelvin 楔形区域内对其进行加密，以约 50 个单元捕捉船体诱导波形，即 $\Delta x_1/\lambda^{\mathrm{w}} = 1/50$。在入口处，



(a) (b)

图 21：用于 Kriso集装箱船仿真的非结构网格（$\mathrm{Re}_{\mathrm{L}} = 14\,000\,000$、$\mathrm{Fn} = 0.26$、$d^{\mathrm{h}}/L^{\mathrm{pp}} = 0.04696$、 $L^{\mathrm{w}}/L^{\mathrm{pp}} = 1.3$、$H^{\mathrm{w}}/L^{\mathrm{pp}} = 0.075$、 T^{w}（V_1/L^{pp}） $= 1.3$）。(a) 静水面的剖切图和 (b) 标示 $P = 2880$ 分区的三维视图。

水平整体速度与线性波理论叠加，并与相应的浓度场一同显示。与第 6 节中的水翼研究一致，湍流入口量由 $I = 5\%$ 的湍流强度和 $\nu_t/\nu = 1$ 得出。出口处指定静水压力边界，侧向和垂向边界采用滑移壁面。流场以 $\Delta t\, V_1/\lambda^{\mathrm{w}} = 0.0015$的时间步长进行时间积分。

S $= 7$ 状态变量的归一化分别采用 $\varphi^{v} = V_1$、 $\varphi^{p} = \rho\, G\, L^{\mathrm{pp}}$、 $\varphi^{k} = u_\tau^2$和 $\varphi^{\varepsilon} = u_\tau^4$，对应速度、压力、湍动能及其耗散。摩擦速度 u_τ同样根据湍流平板公式得出，即$u_\tau = \sqrt{\tau_{\mathrm{w}}/\rho}$，其中 $\tau_{\mathrm{w}} = c_{\mathrm{f}} \rho V_1^2/2$ 和 $c_{\mathrm{f}} = 0.026/\mathrm{Re}^{(1/7)}$。与之前的二维研究相比，离散化工作量大幅增加。系统的长度

矩阵为 N = Y P = (S L) P $\approx 210\,000\,000$，该矩阵分布在 $P = 2880$ 个分区上。六个波浪周期对应于 T = 10000 个时间步，通过 itSVD策略进行压缩，全局状态矩阵为 $\boldsymbol{Y} \in \mathbb{R}^{210\,000\,000 \times 10\,000}$。

为避免初始瞬态效应，流场模拟了 T = 50000个时间步，并在最后一个时间区间 $0.8 < t/\mathrm{T} < 1.0$ 内（共 10000 个时间步）构建 itSVD。使用至少 $o = 100$ 个奇异值（即最后一个时间区间的 1%）来启动秩自适应，同时要求保留 $\eta^o = 90\%$ 的初始遗漏矩阵能量，如公式（8）所示。对于底层硬件（≈ 320GB/96 个 CPU），估算可接受的批大小为 $b_{\max} \leq 620$，因此得到 $b = 500$。

相关时间区间内所考虑的奇异值数量变化如图 22（左）所示。每隔 $b = 500$个时间步调整一次，可以清楚地观察到 itSVD 更新，并显示所考虑的奇异值数量持续增加。每次更新约增加 10–15 个奇异值，最终秩在 $t/\mathrm{T} = 1$时达到 $q = 422$。所考虑奇异值的分布如图 22（右）所示，其中同样以最前面的 10–30 个奇异值为主，它们超过 $\mathcal{O}(10^1)$，比所考虑的最小奇异值高出约两个数量级。

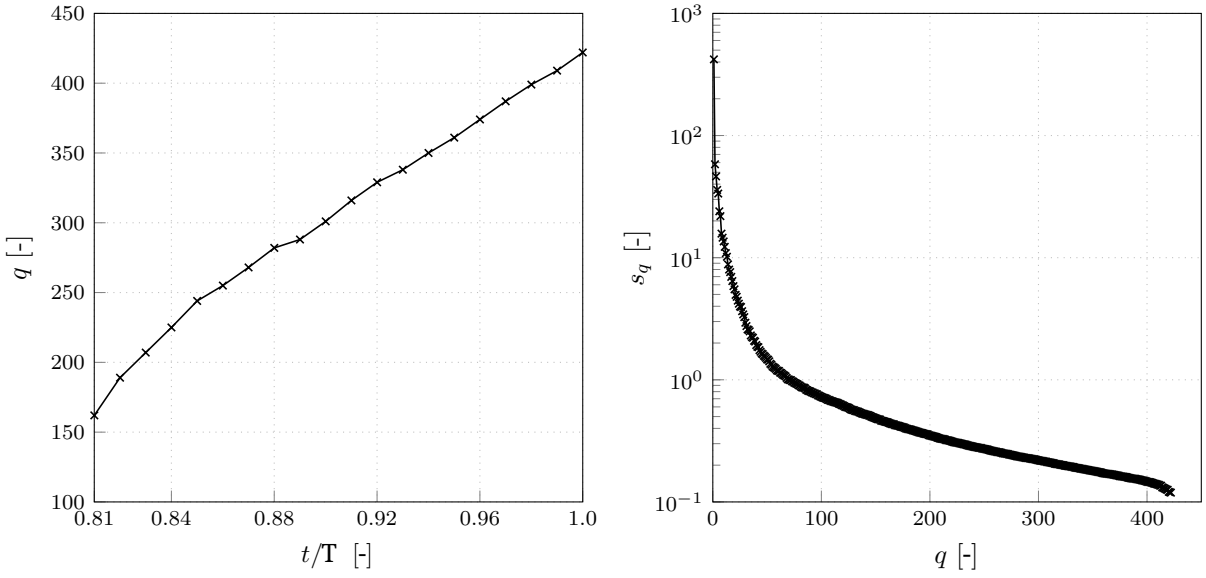


图 22：Kriso集装箱船案例（$\mathrm{Re_L} = 14\,000\,000$，Fn $= 0.26$、$d^{\mathrm{h}}/L^{\mathrm{PP}} = 0.04696$、$L^{\mathrm{w}}/L^{\mathrm{PP}} = 1.3$、$H^{\mathrm{w}}/L^{\mathrm{PP}}$ =0。075，T^w（V_1/L^{PP}）$= 1.3$）。左图：自最小秩 $o = 100$ 开始，为保留 $\eta^o = 90\%$ 的矩阵能量而自适应估计的秩 q 的演化。右图：在 $t/\mathrm{T} = 1$ 时所有 $q = 422$ 个最终奇异值的分布。

基于自适应生成的 itSVD，进行了六项评估研究，改变用于重构的奇异值数量，即 $\tilde{q} = [1, 2, 4, 8, 16, 422$（$= q$）$]$。图 23 展示了积分结果的比较，其中给出了模拟阻力（黑色）以及通过使用 $\tilde{q} = 1$（橙色）、$\tilde{q} = 2$（蓝色）和 $\tilde{q} = 4$（紫色）进行评估、基于重构场获得的对应数值。该图区分了压力、摩擦和总阻力。摩擦力始终会增加阻力。尽管压力阻力会周期性地变为正值，但总阻力不会发生符号变化。对于两个最小评估秩 $\tilde{q} = [1, 2]$，在所有图中，重构贡献与模拟值之间都出现了明显偏差。特别是对于 $\tilde{q} = 4$，压力以及摩擦力的幅值能够得到更好的重构。

图 24 描绘了在 itSVD 相关时间范围内重构阻力系数 $|\tilde{c}_{\mathrm{d}} - c_{\mathrm{d}}|/c_{\mathrm{d}} \cdot 100\%$ 的误差。同样，压力、摩擦阻力和总阻力贡献分别进行了区分。所有贡献均显示出误差从约 $\mathcal{O}(10^2)$（$\tilde{q} = 1$）降低至 $\mathcal{O}(10^{-2})$（$\tilde{q} = 422$）。图 25 比较了最终时刻 $t/\mathrm{T} = 1$ 的波面升高场。每幅图均将模拟数据（上）与相应的重构结果（下）进行比较，分别对应 $\tilde{q} = 1$（a）、$\tilde{q} = 2$（b）、$\tilde{q} = 4$（c）、$\tilde{q} = 8$（d）、$\tilde{q} = 16$（e）、$\tilde{q} = 422$（f）。对于秩非常低的重构结果 $\tilde{q} = [1, 2]$，波幅显著减小，并出现了明显的相位偏移。后者在 $x_2/L^{\mathrm{PP}} = 0.0741$（上）、$x_2/L^{\mathrm{PP}} = 0.1509$（中）、$x_2/L^{\mathrm{PP}} = 0.4224$（下）处的三条波面截线上也清晰可见，这些截线显示于图 27（$\tilde{q} = 1$，红色）中，其相应的重构误差则见图 27。对于 $\tilde{q} = 4$，波场的重构显著改善，这

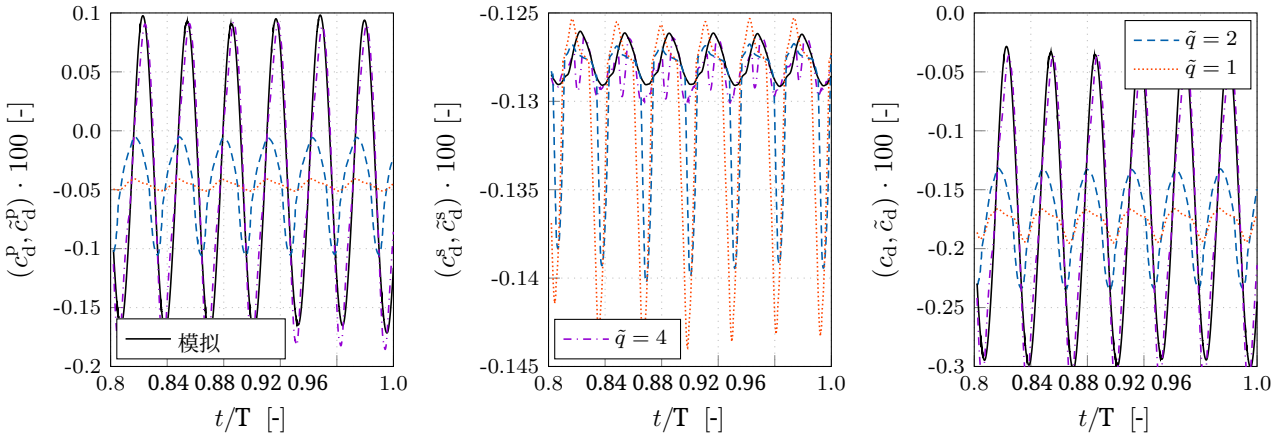


图 23：Kriso集装箱船案例（$\mathrm{Re_L} = 14\,000\,000$，$\mathrm{Fn} = 0.26$、$d^\mathrm{h}/L^\mathrm{pp} = 0.04696$、$L^\mathrm{w}/L^\mathrm{pp} = 1.3$、$H^\mathrm{w}/L^\mathrm{pp} = 0.075$，$T^w$（$V_1/L^\mathrm{pp}$）$= 1.3$）。基于 $\tilde{q} = 1$（橙色）、$\tilde{q} = 2$（蓝色）和$\tilde{q} = 4$（紫色）个奇异值，对压力（左）、摩擦阻力（中）以及总阻力（右）进行模拟（黑色）和重构的阻力。

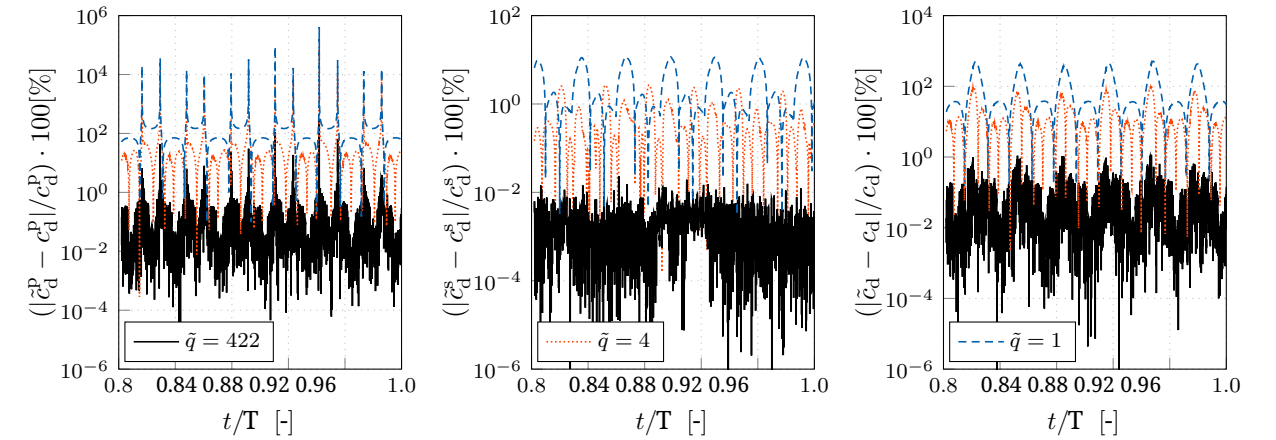


图 24：Kriso集装箱船案例（$\mathrm{Re_L} = 14\,000\,000$，$\mathrm{Fn} = 0.26$、$d^\mathrm{h}/L^\mathrm{pp} = 0.04696$、$L^\mathrm{w}/L^\mathrm{pp} = 1.3$、$H^\mathrm{w}/L^\mathrm{pp}$ =0.075，T^w（V_1/L^pp）$= 1.3$）。使用较低和较高的秩近似，对压力（左）、摩擦阻力（中）以及总阻力（右）重构的 $\tilde{c}_\mathrm{d}$ 阻力系数的相对误差。

与图 23 和图 24 中观察到的阻力重构可接受的一致性相符。使用 $\tilde{q} = 16$ 个奇异值时，尾流中仅显示出细微差异。在图 25 的 $\tilde{q} = 422$ 中，这些差异消失了，且图 27（紫色）中的波面截线误差降至 $\mathcal{O}(10^{-2})$。

7.1 开销

要充分重构局部数据和积分阻力量，大约需要 40–80 个奇异值（参见图 23–27），这使总计 $\mathrm{T} = 10000$ 时间步的存储开销降至百分之一以下。当前的 itSVD 嵌入时间积分循环中，因此会产生额外的运行时间开销。后者可以根据算法2中的每一行提取。对于所展示的自适应船舶流动 itSVD，使用$q = 422$ 个奇异值覆盖 $\mathrm{T} = 10000$ 个时间步，且批大小为 $b = 500$，表2给出了预期额外开销的估计值。使用 $q/\mathrm{T} \approx 25$ 和 $b/q \approx 1$ 时，计算额外开销应约为 $\mathcal{O}$（10%）；当批大小降至 $b \approx 15$时，该开销将增加约 5 倍。

图28以蓝色柱状图呈现了测得的开销，其依据是总仿真运行时间的相对占比 $\tilde{t}^\mathrm{line}/\tilde{t}^\mathrm{NS} \cdot 100$，其中 $\tilde{t}^\mathrm{RANS}$ 仅指RANS系统的时间积分开销。13.07%的总开销被认为是可以接受的。总开销中相当可观的一部分来自

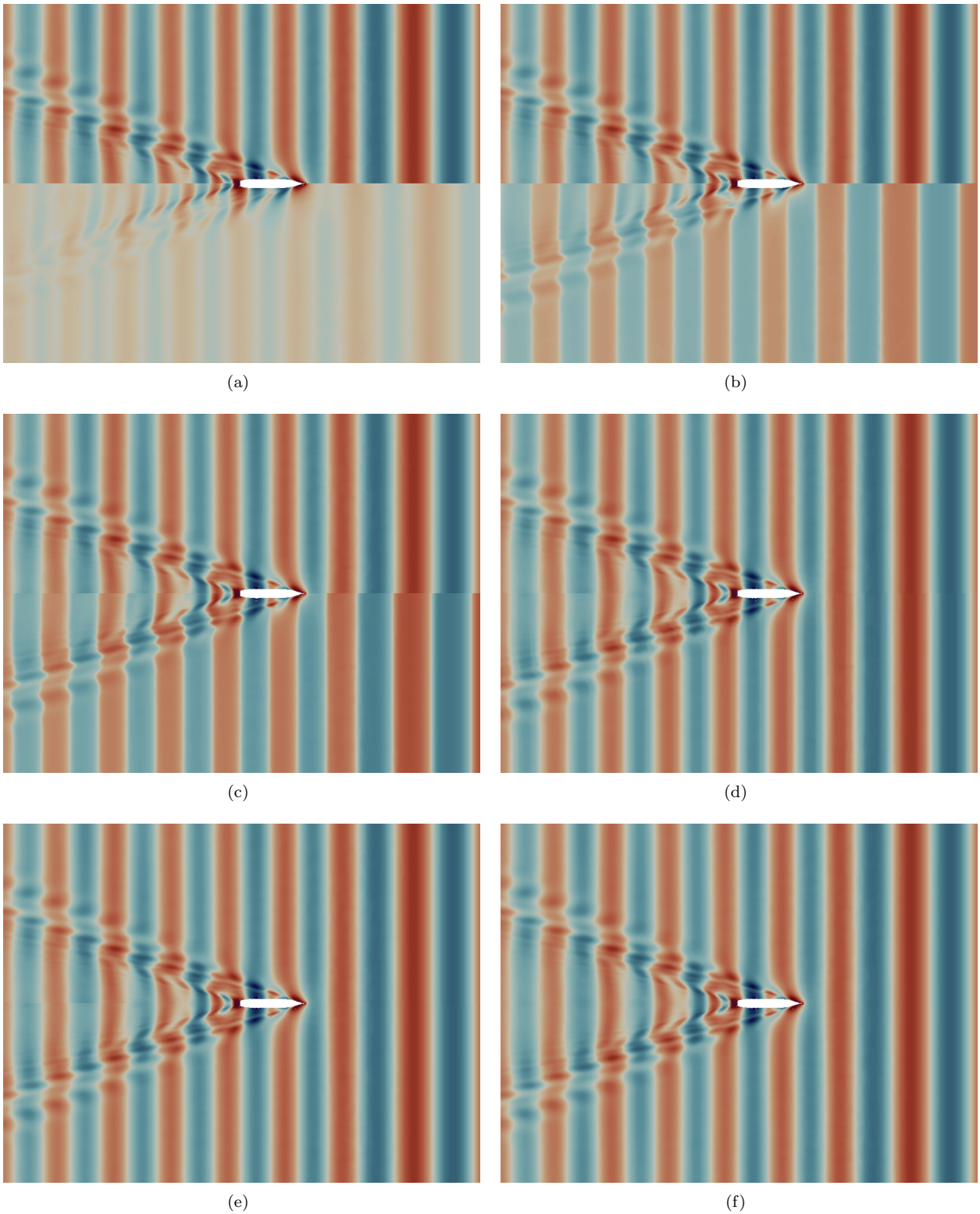


图25：Kriso集装箱船案例（$\mathrm{Re_L} = 14\,000\,000$、Fn $= 0.26$、 $d^{\mathrm{h}}/L^{\mathrm{pp}} = 0.04696$、 $L^{\mathrm{w}}/L^{\mathrm{pp}} = 1.3$、 $H^{\mathrm{w}}/L^{\mathrm{pp}}$ =0.075、 T^{w} （V_1/L^{pp}） $= 1.3$）： $t/\mathrm{T} = 1.0$ 处的模拟波面升高（上）及重构波面升高（下），分别对应$\tilde{q} = 1$ （a）、 $\tilde{q} = 2$ （b）、 $\tilde{q} = 4$ （c）、 $\tilde{q} = 8$ （d）、 $\tilde{q} = 16$ （e）和 $\tilde{q} = 422 = q$ （f）。

全局QR分解（第14行，占5.49%），这凸显了Brand（2006）；Fareed等（2018）；Zhang（2022）；Li等（2022）中对其必要性展开的积极讨论。第5行中的第二次QR分解贡献了

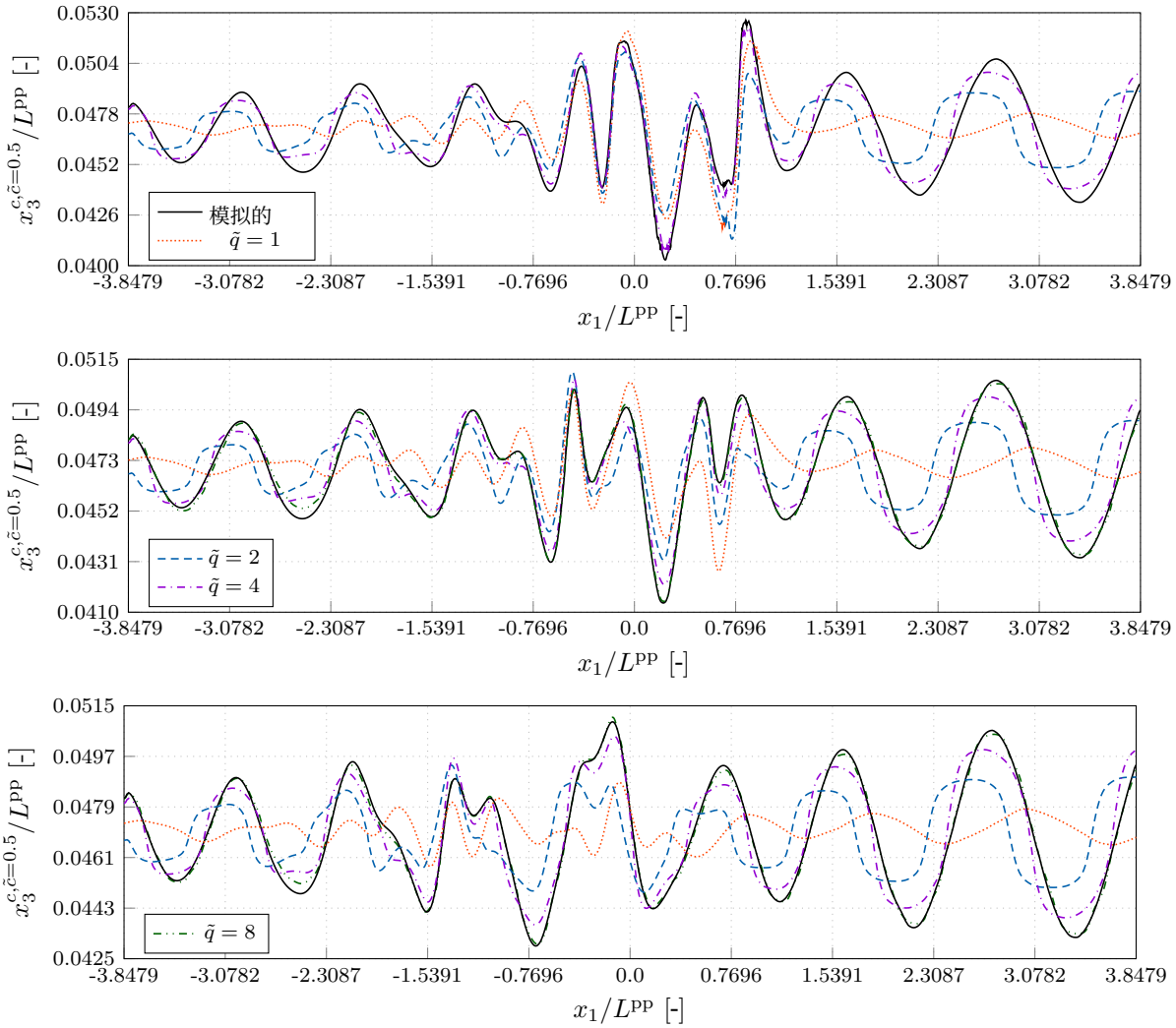


图26：Kriso集装箱船案例（$\mathrm{Re_L}=14\,000\,000$、Fn $=0.26$、 $d^{\mathrm{h}}/L^{\mathrm{pp}}=0.04696$、 $L^{\mathrm{w}}/L^{\mathrm{pp}}=1.3$、 $H^{\mathrm{w}}/L^{\mathrm{pp}}$ =0.075、 T^{w} （V_1/L^{pp}） $=1.3$）：模拟结果（黑色）与图25中四个重构场景在 $t/\mathrm{T}=1$时的 $x_2/L^{\mathrm{pp}}=0.0741$ （顶部）、 $x_2/L^{\mathrm{pp}}=0.1509$ （中部）和 $x_2/L^{\mathrm{pp}}=0.4224$（底部）波浪剖线

.

占比为 2.14%，第 16 行和第 18 行中的两次子空间旋转还会额外占用 2.5%。出人意料的是，第 8 行中局部奇异值分解求解的额外工作量相对较低，仅为 ≈0.02%。与第 3 行和第 4 行中的全局——即随 N 扩展的——矩阵-向量运算相关的工作量则进一步显著增加。其余所有算法行的工作量均低于 0.01%，因此未在图 28 中列出。从效率角度来看，所提出的 itSVD算法主要得益于将各个快照分批处理，参见第 5 节。为突出这一点，采用依次减半的批次大小$b=[250,125,62,31,15]$，重新执行船舶流场的 itSVD 构造。对于更小的批次大小，相对计算时间出现了明显的非线性增长：从 13.58%（$b=250$）增加到 16.94%（$b=125$）、25.75%（$b=62$）和 39.27%（$b=31$），而最小批次大小下则达到 60.28%（$b=15$）。减小分批处理大小时所观察到的开销增加，证实了二维圆柱绕流示例在表 2 中的发现。由此产生的仿真开销分别以红色、浅绿色、紫色、橙色和深绿色添加到图 28 中，对应于 $b=250$、 $b=125$、 $b=62$、 $b=31$ 和 $b=15$ 。正如表 3 中对圆柱流动研究所展示的那样，将批宽减小至 $b=1$ 这一极限情况，即使将 $q/\mathrm{T}\cdot 100=4.22\%$大幅削减，也会使 itSVD 构建时间进入与实际时间积分工作量相当的范围。最后，使用 Matlab©曲线拟合工具箱（参见 The MathWorks (2019a)）将数据拟合为形如 $\tilde{t}(b)=r\,e^{s\,b}+t\,e^{u\,b}$的指数函数，其中 $r=73.79$、 $s=-0.03681$、 $t=17.45$和 $u=-0.0006676$，从而揭示出外推开销为 $\tilde{t}(b=1)\approx 89\%$。

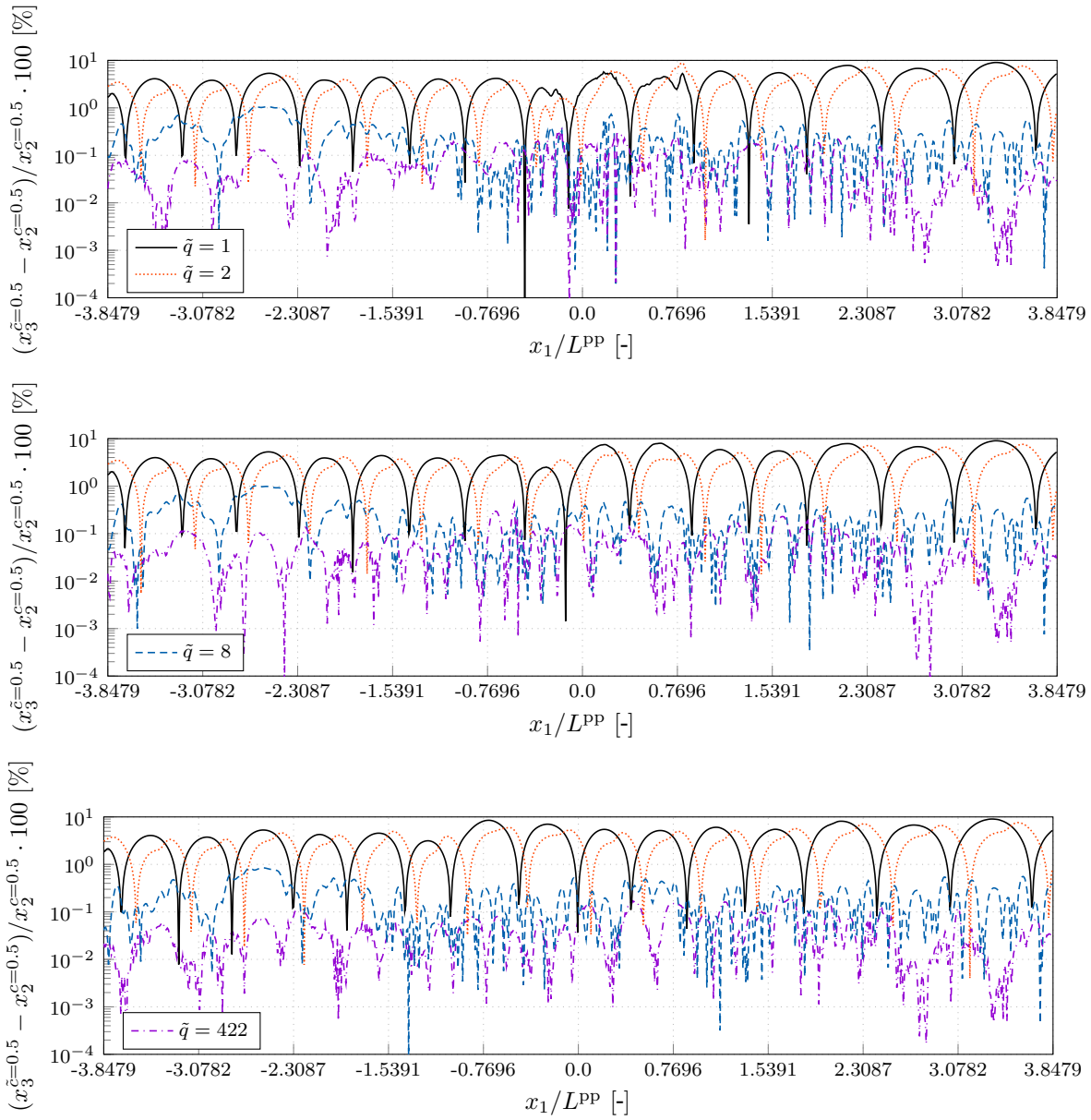


图 27 $_{\text{L}} = 14\,000\,000$ $= 0.26$ $d^{\text{h}}/L^{\text{pp}} = 0.04696$ $L^{\text{w}}/L^{\text{pp}} = 1.3$ $H^{\text{w}}/L^{\text{pp}} =$：Kriso集装箱船案例（Re、Fn、 0.075 T^w $V_1/L^{\text{pp}} = 1.3$ $x_2^{\tilde{c}=0.5} - x_2^{c=0.5}\,/x_2^{c=0.5} \cdot 100$ $x_2/L^{\text{pp}} = 0.0741$ 、 、 、 （ ））：四个重构场景中（顶部）、 $x_2/L^{\text{pp}} = 0.1509$ （中部）、 $x_2/L^{\text{pp}} = 0.4224$ （底部）处的波面升高误差（ ），来自图 25，在 $t/$ T $= 1$。

8 结论与展望

该论文介绍了一种并行增量奇异值分解方法，用于在运行时对大型空间划分网格上获得的随时间演化的偏微分方程结果进行高效压缩，即规约。此类数据规约对于避免后续数据处理步骤中的高数据传输速率，或作为数据科学框架下深入分析的基础工具，可能具有相当重要的意义。该过程独立于区域分解，并已在最多 2900 个进程上进行了测试。它能够达到以往引入的奇异值分解方法的准确性，并可结合基于精确能量含量的启发式质量度量，以调整可达到的准确性。在介绍研究动机之后，最关键的问题涉及相关的存储和运行时间开销。从 $\mathcal{O}$(1%T) 个奇异值出发——其中 T 表示时间数据大小——在墙钟时间方面实现了约 $\mathcal{O}$(10%) 的计算额外开销，同时存储开销仅为$\mathcal{O}$(5%)。如本文所建议，将增量奇异值分解更新进行分批处理，尤其有助于降低运行时间开销，并且符合大规模并行应用的需求。未来的研究方向将着眼于保留

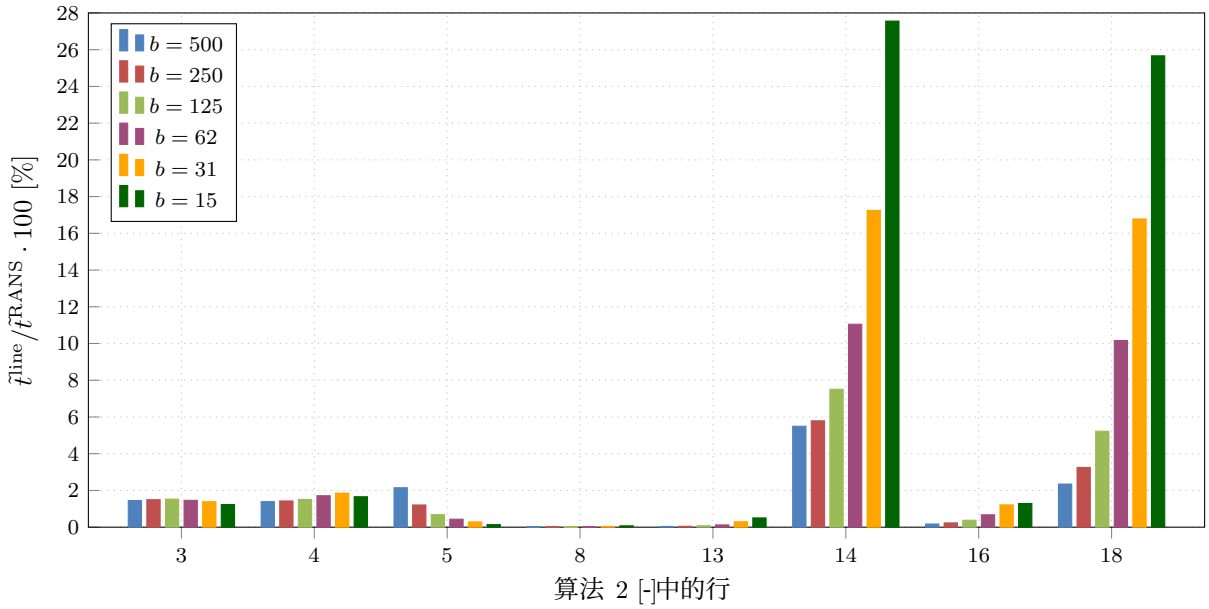


图28：Kriso集装箱船案例（$\text{Re}_{\text{L}} = 14\,000\,000$、Fn $= 0.26$, $d^{\text{h}}/L^{\text{pp}} = 0.04696$, $L^{\text{w}}/L^{\text{pp}} = 1.3$, $H^{\text{w}}/L^{\text{pp}}$ =0.075, T^{w}(V_1/L^{pp}) $= 1.3$）：算法 2 $\tilde{t}^{\text{line}}$每行的itSVD 构建时间总和，相对于RANS模拟实际经过时间 $\tilde{t}^{\text{RANS}}$进行归一化。

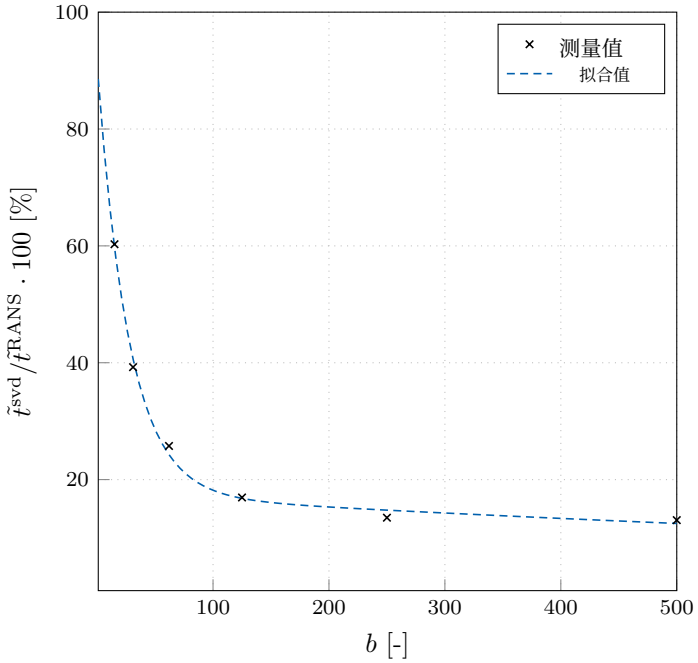


图29：Kriso集装箱船案例（$\text{Re}_{\text{L}} = 14\,000\,000$、Fn $= 0.26$, $d^{\text{h}}/L^{\text{pp}} = 0.04696$, $L^{\text{w}}/L^{\text{pp}} = 1.3$, $H^{\text{w}}/L^{\text{pp}} =$ 0.075, T^{w}(V_1/L^{pp}) $= 1.3$）：测得的itSVD 构建时间（算法2各行之和，参见图28） $\tilde{t}^{\text{svd}}$，相对于积分RANS系统所需的仿真实际经过时间 $\tilde{t}^{\text{RANS}}$进行归一化。此外，将计算开销拟合为指数函数。

在解决该过程算法效率问题的同时，确保重构数据的可实现性。对于每次 itSVD 更新，所采用的正交化步骤可能并非必要，并且可以利用适当的非正交性度量加以控制。此外，该方法还可以扩展到空间非不变网格，例如自适应可细化/粗化网格；同时也可以研究替代性的预条件概念，例如通过局部单元体积进行预条件处理。同样，对于相应字段，也可以采用不同的参数处理方式。

CRediT 作者贡献声明

N.K.：经费获取、概念化、形式分析、调查、方法论、软件、验证、可视化、撰写初稿、审阅与编辑。**H.F.**：方法论、软件、形式分析、审阅与编辑。**M.H.**：经费获取、方法论、审阅与编辑。**T.R.**：经费获取、项目管理、资源、方法论、撰写初稿、审阅与编辑。

利益冲突声明

作者声明，他们不存在任何已知的竞争性经济利益或个人关系，可能会被认为影响了本文所报告的工作。

致谢

本文是协作研究中心 TRR181“ 大气与海洋中的能量传递”项目 M6 和 T4 的合作成果。该项目由德国研究基金会（DFG）根据资助编号 274762653 资助，N.K.、M.H. 和 T.R. 对此表示感谢。第二作者（H.F.）感谢 DFG 在国际研究培训组 GRK 2657“高维计算力学技术”框架下，根据资助编号 433082294 提供的资助。作者衷心感谢资源分配委员会授予的计算时间，以及 NHR@ZIB 和 NHR@G¨ottingen 的 Lise 和 Emmy 超级计算机作为 NHR 基础设施的一部分所提供的计算资源。本研究的计算工作使用了项目 hhi00033（“船舶流体动力阻力最小化”）和 hhi00037（“海气界面能通量”）下的计算资源。

参考文献

L. Agostini。使用自动编码器技术探索和预测流体动力系统。Physics of Fluids，32(6):067103，2020年。数字对象标识符:10.1063/5.0012906。E. Anderson、Z. Bai、C. Bischof、S. Blackford、J. Demmel、J. Dongarra、J. Du Croz、A. Greenbaum、S. Hammar- ling、A. McKenney 和 D. Sorensen。LAPACK 用户指南。工业与应用数学学会，宾夕法尼亚州费城，第三版，1999年。ISBN 0-89871-447-8（平装）。W. Arrighi、G. Oxberry、T. Vassilevska 和 K. Chand。libROM 用户指南与设计，2015年7月。C. Bach、D. Ceglia、L. Song 和 F. Duddeck。用于大型非线性动力学问题基于投影的模型降阶的随机低秩近似方法。International Journal for Numerical Methods in Engineering，118(4):209–241，2019年。数字对象标识符:10.1002/nme.6009。F. Ballarin 和 G. Rozza。参数化流固耦合问题的 POD–Galerkin 单体降阶模型。International Journal for Numerical Methods in Fluids，82(12):1010–1034，2016。数字对象标识符:10.1002/fld.4252。F. Ballarin、A. Manzoni、A. Quarteroni 和 G. Rozza。参数化稳态不可压缩 Navier–Stokes 方程 POD–Galerkin 近似的 Supremizer 稳定化。International Journal for Numerical Methods in Engineering，102(5):1136–1161，2015年。数字对象标识符:10.1002/nme.4772。J. Banks、A. Phillips 和 S. Turnock。KCS 船舶阻力分量的自由表面 CFD 预测。载于第13届数值拖曳水池研讨会，德国杜伊斯堡，第6页，2010年。P. Benner、S. Gugercin 和 K. Willcox。参数化动力系统基于投影的模型降阶方法综述。SIAM review，57(4):483–531，2015年。数字对象标识符:10.1137/130932715。P. Benner、S. Dolgov、A. Onwunta 和 M. Stoll。随机 Navier–Stokes 方程约束的最优控制问题的低秩解。International Journal for Numerical Methods in Fluids，92(11):1653–1678，2020a。数字对象标识符:10.1002/fld.4843。

P. Benner、W. Schilders、S. Grivet-Talocia、A. Quarteroni、G. Rozza 和 L. Miguel Sil- veira。模型降阶：第2卷：基于快照的方法与算法。De Gruyter，2020b。数字对象标识符:10.1515/9783110671490。M. Brand。含缺失值不确定数据的增量奇异值分解。载于欧洲计算机视觉会议，第707–720页。Springer，2002年。数字对象标识符:10.1007/3-540-47969-4 47。M. Brand。薄奇异值分解的快速低秩修正。Linear Algebra and its Applications，415(1):20–30，2006年。数字对象标识符:10.1016/j.laa.2005.07.021。A. B¨unger、S. Dolgov 和 M. Stoll。结合等几何分析的偏微分方程约束优化低秩张量方法。SIAM Journal on Scientific Computing，42(1):A140–A161，2020年。数字对象标识符:10.1137/18M1227238。A. Caiazzo、T. Iliescu、V. John 和 S. Schyschlowa。不可压缩流速度–压力降阶模型的数值调查。Journal of Computational Physics，259:598–616，2014年。数字对象标识符:10.1016/j.jcp.2013.12.004。P.M. Carrica、H. Fu 和 F. Stern。KRISO 集装箱船（KCS）模型的自航、自由下沉与纵倾、以及迎浪运动的计算。Applied Ocean Research，33(4):309–320，2011年。数字对象标识符:10.1016/j.apor.2011.07.003。S. Dolgov 和 M. Stoll。Navier–Stokes 方程约束优化问题的低秩解。SIAM Journal on Scientific Computing，39(1):A255–A280，2017年。数字对象标识符:10.1137/15M1040414。J.H. Duncan。拖曳水翼产生的破碎波实验研究。Proceedings of the Royal Society of London A，377(1770):331–348，1981年。数字对象标识符:10.1098/rspa.1981.0127。J.H. Duncan。二维水翼的破碎波与非破碎波阻力。Journal of Fluid Mechanics，126:507–520，1983年。数字对象标识符:10.1017/S0022112083000294。C. Eckart 和 G. Young。用另一个更低秩的矩阵近似一个矩阵。Psychometrika，1(3): 211–218，1936年。数字对象标识符:10.1007/BF02288367。H. Fareed 和 J.R. Singler。关于连续时间数据增量POD算法的注记。Applied Numerical Mathematics，144:223–233，2019年。数字对象标识符:10.1016/j.apnum.2019.04.020。H. Fareed 和 J.R. Singler。偏微分方程仿真数据增量本征正交分解算法的误差分析。Journal of Computational and Applied Mathematics，368:112525，2020年。数字对象标识符:10.1016/j.cam.2019.112525。H. Fareed、J.R. Singler、Y. Zhang 和 J. Shen。偏微分方程仿真数据的增量本征正交分解。Computers & Mathematics with Applications，75(6):1942–1960，2018年。数字对象标识符:10.1016/j.camwa.2017.09.012。C. Gr¨aßle 和 M. Hinze。利用任意有限元离散的演化方程 POD 降阶建模。Advances in Computational Mathematics，44(6):1941–1978，2018年。数字对象标识符:10.1007/s10444- 018-9620-x。C. Gr¨aßle、M. Hinze、J. Lang 和 S. Ullmann。采用空间自适应快照的不可压缩流 POD 模型降阶。Advances in Computational Mathematics，45(5):2401–2428，2019年。数字对象标识符:10.1007/s10444- 019-09716-7。M. Gubisch 和 S. Volkwein。线性二次最优控制的本征正交分解。模型降阶与近似：理论与算法，15(1)，2017年。数字对象标识符:10.1137/1.9781611974829.ch1。J.S. Hesthaven 和 S. Ubbiali。利用神经网络对非线性问题进行非侵入式降阶建模。Journal of Computational Physics，363:55–78，2018年。数字对象标识符:10.1016/j.jcp.2018.02.037。S. Hijazi、G. Stabile、A. Mola 和 G. Rozza。湍流的数据驱动 POD-Galerkin 降阶模型。Journal of Computational Physics，416:109513，2020年。数字对象标识符:10.1016/j.jcp.2020.109513。C.W. Hirt 和 B.D. Nichols。用于自由边界动力学的流体体积法（VoF）。Journal of Computational Physics，39(1):201–225，1981年。数字对象标识符:10.1016/0021-9991(81)90145-5。

M.A. Iwen 和 B.W. Ong。用于大型网络聚类数据分析的分布式增量奇异值分解算法。SIAM矩阵分析与应用杂志，37(4):1699–1718，2016。数字对象标识符:10.1137/16M1058467。G. Karypis。METIS：用于非结构化图分区、网格分区以及稀疏矩阵减少填充排序的软件包，第5.0版。计算机科学与工程系，明尼苏达大学，明尼阿波利斯，MN 55455，2011年。URL https://www.lrz.de/services/ software/mathematik/metis/metis_5_0.pdf。G. Karypis 和 K. Schloegel。PARMETIS：并行图划分和稀疏矩阵排序库，第4.0版。计算机科学与工程系，明尼苏达大学，明尼阿波利斯，MN 55455，2011年。URL https://www.lrz.de/services/software/mathematik/metis/parmetis_4_0.pdf。A. Kodakkal、B. Keith、U. Khristenko、A. Apostolatos、K.-U. Bletzinger、B. Wohlmuth 和 R. W¨uchner。不确定风况下高层建筑的风险规避设计。《计算机方法在应用力学与工程中的应用》，402:115371，2022年。¨A. Kolmogoroff。关于给定函数类中函数的最佳近似。《数学年刊》，页码107–110，1936年。数字对象标识符:10.2307/1968691。J. Kr¨oger、N. K¨uhl 和 T. Rung。用于船舶流体动力学优化的伴随流体体积法。《船舶技术研究》，65(1):47–68，2018年1月。数字对象标识符:10.1080/09377255.2017.1411001。N. K¨uhl。湍流两相Navier-Stokes系统约束下的基于伴随的形状优化。博士论文，汉堡工业大学，2021年。N. K¨uhl。增量奇异值分解示例。MATLAB Central文件交换平台，2023年。URL https://www.mathworks.com/matlabcentral/fileexchange/124815-incremental-singular-value-decomposition-example。检索日期：2023年2月15日。N. K¨uhl 和 T. Rung。离散伴随动量加权插值策略。Journal of Computational Physics，467:111474，2022年。数字对象标识符:10.1016/j.jcp.2022.111474。N. K¨uhl、M. Hinze 和 T. Rung。用于海洋自由表面流的Cahn–Hilliard Navier–Stokes模拟。《实验与计算多相流》，2021a。数字对象标识符:10.1007/s42757-020-0101-3。N. K¨uhl、J. Kr¨oger、M. Siebenborn、M. Hinze 和 T. Rung。不相溶流的流体体积法伴随补充方法。Journal of Computational Physics，440:110411，2021b年。数字对象标识符:10.1016/j.jcp.2021.110411。K. Kunisch 和 S. Volkwein。流体动力学中一般方程的Galerkin本征正交分解方法。SIAM数值分析杂志，40(2):492–515，2002年。数字对象标识符:10.1137/S0036142900382612。L. Larsson、F. Stern 和 M. Visonneau。数值船舶流体动力学：2010年哥德堡研讨会评估。Springer，2013年。T. Lassila、A. Manzoni、A. Quarteroni 和 G. Rozza。流体动力学中的模型降阶：挑战与展望。《降阶方法与计算约简建模》，页码235–273，2014年。数字对象标识符:10.1007/978-3-319-02090-7 9。X. Li、S. Hulshoff 和 S. Hickel。利用降阶原始解实现面向伴随的网格细化以进行大涡模拟：初步一维Burgers研究。《计算机方法在应用力学与工程中的应用》，379:113733，2021年。X. Li、S. Hulshoff 和 S. Hickel。用于非定常模拟的在线本征正交分解及其并行化的增强算法。Computers & Mathematics with Applications，126:43–59，2022年。L. Lin 和 Y. Tong。具有长程成对相互作用的张量网络算子的低秩表示。SIAM Journal on Scientific Computing，43(1):A164–A192，2021年。数字对象标识符:10.1137/19M1287067。S. Lorenzi、A. Cammi、L. Luzzi 和 G. Rozza。用于Navier–Stokes方程和RANS方程有限体积近似的POD-Galerkin方法。《计算机方法在应用力学与工程中的应用》，311: 151–179

，2016。数字对象标识符：/j.cma.2016.08.006.
10.1016

X. Luo-Theilen 和 T. Rung。《航道中机械耦合体的计算》。《船舶技术研究》，64(3)：129–143，2017年。数字对象标识符：10.1080/09377255.2017.1348654。X. Luo-Theilen 和 T. Rung。《海洋结构物安装过程的数值分析》。《海洋工程》，179：116–127，2019年。数字对象标识符：10.1016/j.oceaneng.2019.03.004。L. Mainini 和 K. Willcox。《支持实时结构评估与决策制定的代理建模方法》。《AIAA期刊》，53(6)：1612–1626，2015年。数字对象标识符：10.2514/1.J053464。M. Manzke。《采用动态目标导向局部网格自适应的流动高效仿真可扩展方法的开发》。《汉堡工业大学》博士论文，2018年。M. Manzke、J.P. Voss 和 T. Rung。《海上两相流的子循环策略》。载于《数值流体力学与多学科设计笔记》，第120卷，第237–251页。Springer，2012年。数字对象标识符：10.1007/978-3-642-33221-0 14。A.-S.I. Margetis、E.M. Papoutsis-Kiachagias 和 K.C. Giannakoglou。《支持二维/三维非结构网格非定常伴随的有损压缩技术》。《计算机方法在应用力学与工程中的应用》，387：114152，2021年。数字对象标识符：10.1016/j.cma.2021.114152。A.-S.I. Margetis、E.M. Papoutsis-Kiachagias 和 K.C. Giannakoglou。《通过协同使用检查点与压缩技术降低非定常伴随的内存需求》。《国际流体数值方法杂志》，2022年。数字对象标识符：10.1002/fld.5136。N. Mastronardi、E.E. Tyrtyshnikov 和 P. Van Dooren。《一种更新和缩减主导核主成分的快速算法》。《SIAM矩阵分析与应用杂志》，31(5)：2376–2399，2010年。数字对象标识符：10.1137/090774422。消息传递接口论坛。《MPI：消息传递接口标准4.0版》，2021年6月。URL https://www.mpi-forum.org/docs/mpi-4.0/mpi40-report.pdf。T. Murata、K. Fukami 和 K. Fukagata。《利用卷积神经网络进行流体动力学的非线性模态分解》。《Journal of Fluid Mechanics》，882，2020年。数字对象标识符：10.1017/jfm.2019.822。H. Nobis、P. Schlatter、E. Wadbro、M. Berggren 和 D.S. Henningson。《通过超疏水表面拓扑优化延迟模态层流–湍流转捩》。《计算机方法在应用力学与工程中的应用》，403：115721，2023年。M. Nonino、F. Ballarin 和 G. Rozza。《用于流固耦合问题的整体式与分区式降阶基方法》。《流体》，6(6)：229，2021年。数字对象标识符：10.3390/fluids6060229。R. Pache 和 T. Rung。《集装箱船上层建筑气动力的数据驱动代理建模》。《计算流体力学工程应用》，16(1)：746–763，2022年。数字对象标识符：10.1080/19942060.2022.2044383。A. Quarteroni 和 G. Rozza。《用于建模与计算约简的降阶方法》，第9卷。Springer，2014年。C.M. Rhie 和 W. L. Chow。《带尾缘分离翼型后方湍流流场的数值研究》。《AIAA期刊》，21(11)：1525–1532，1983年。数字对象标识符：10.2514/3.8284。T. Rung、K. W¨ockner、M. Manzke、J. Brunswig、C. Ulrich 和 A. St¨uck。《海事CFD应用的挑战与展望》。《德国造船技术学会年鉴》，103：127–39，2009年。Z. Shen、D. Wan 和 P.M. Carrica。《OpenFOAM中的动态重叠网格及其在KCS自推进与操纵中的应用》。《海洋工程》，108：287–306，2015年。数字对象标识符：10.1016/j.oceaneng.2015.07.035。C.D. Simonsen、J.F. Otzen、S. Joncquez 和 F. Stern。《规则迎浪中KCS升沉与纵摇的实验流体动力学和计算流体力学研究》。《海洋科学与技术杂志》，18(4)：435–459，2013年。数字对象标识符：10.1007/s00773-013-0219-0。L. Sirovich。《湍流与相干结构的动力学。I. 相干结构》。《应用数学季刊》，45(3)：561–571，1987年。数字对象标识符：10.1090/qam/910462。

G. Stabile 和 G. Rozza。参数化不可压缩纳维–斯托克斯方程的有限体积POD-Galerkin稳定化降阶方法。《计算机与流体》，173:273–284，2018年。数字对象标识符:10.1016/j.compfluid.2018.01.035。M. Stoll 和 T. Breiten。偏微分方程约束优化的时间低秩方法。《SIAM Journal on Scientific Computing》，37(1):B1–B29，2015年。数字对象标识符:10.1137/130926365。R. Swischuk、L. Mainini、B. Peherstorfer 和 K. Willcox。基于投影的模型降阶：基于物理的机器学习的公式。《计算机与流体》，179:704–717，2019年。数字对象标识符:10.1016/j.compfluid.2018.07.021。HDF集团。分层数据格式，第5版，1997-NNNN。URL https://www.hdfgroup.org/HDF5/。The MathWorks公司。曲线拟合工具箱。马萨诸塞州纳蒂克，美国，2019a年。URL https: //de.mathworks.com/help/curvefit/curve-fitting.html。C. Vezyris、E.M. Papoutsis-Kiachagias 和 K.C. Giannakoglou。关于支持非定常伴随优化的增量奇异值分解方法。《国际流体数值方法杂志》，91(7):315–331，2019年。数字对象标识符:10.1002/fld.4755。S. Walton、O. Hassan 和 K. Morgan。使用本征正交分解和径向基函数对非定常流动进行降阶建模。《应用数学建模》，37(20-21):8930–8945，2013年。数字对象标识符:10.1016/j.apm.2013.04.025。Z. Wang、I. Akhtar、J. Borggaard 和 T. Iliescu。湍流的本征正交分解闭合模型：数值比较。《计算机方法在应用力学与工程中的应用》，237:10–26，2012年。数字对象标识符:10.1016/j.cma.2012.04.015。D.C. Wilcox。CFD湍流建模，第2卷。DCW工业公司，拉加拿大，1998年。K. Willcox 和 J. Peraire。通过本征正交分解实现平衡模型降阶。《AIAA期刊》，40(11):2323–2330，2002年。数字对象标识符:10.2514/2.1570。K. W¨ockner、W. Drazyk 和 T. Rung。用于捕捉复杂海况的高效基于VoF的RANS方法。载于国际海洋工程与北极工程会议，第49149卷，第755–763页，2010年。S. Yakubov、B. Cankurt、M. Abdel-Maksoud 和 T. Rung。针对空化建模的欧拉–拉格朗日方法的MPI/OpenMP混合并行化。《计算机与流体》，80:365–371，2013年。数字对象标识符:10.1016/j.compfluid.2012.01.020。S. Yakubov、T. Maquil 和 T. Rung。使用基于压力的CFD方法进行空化流的欧拉–欧拉模拟的经验。《计算机与流体》，111:91–104，2015年。数字对象标识符:10.1016/j.compfluid.2015.01.008。Y. Zhang。对增量奇异值分解中一个开放问题的回答。arXiv预印本arXiv:2204.05398，2022年。

增强型itSVD算法

算法6： 空间并行/时间串行截断
奇异值分解的改进增量秩 {v3}更新，其中突出显示了算法2的增强部分。

子程序itSVD 更新()**：**

$u =$ 大小$(\boldsymbol{U},\ 2)$, $v =$ 大小$(\boldsymbol{V},\ 1)$, $c = u + b$、 $d = v + b$、 $l =$ 最小值$(q,\ c)$; // Required sizes

$\boldsymbol{M} = \boldsymbol{U}^{\mathrm{T}}\boldsymbol{B}$, 并行求和$(\boldsymbol{M})$; // 1) $\boldsymbol{M} \in \mathbb{R}^{u \times b}$ 2) Local to Global Operation. cf. Fig. 3

$\boldsymbol{P} = \boldsymbol{B} - \boldsymbol{U}\boldsymbol{M}$; // $\boldsymbol{P} \in \mathbb{R}^{\mathrm{Y} \times b}$ around Eqn. 12

$[\boldsymbol{Q}_{\mathrm{P}}, \boldsymbol{R}_{\mathrm{P}}] =$ 并行格拉姆-施密特正交化$(\boldsymbol{P})$; // Global QR Decomposition

$\boldsymbol{Q} = [\boldsymbol{U} \quad \boldsymbol{Q}_{\mathrm{P}}]$; // Prepare $\boldsymbol{U}$ Update via $\boldsymbol{Q} \in \mathbb{R}^{\mathrm{Y} \times l}$, cf. Eqn. 15

$[\boldsymbol{Q}_{\mathrm{Q}}, \boldsymbol{R}_{\mathrm{Q}}] =$ 并行格拉姆-施密特正交化$(\boldsymbol{Q})$; // Global QR Decomposition

如果 进程ID=0**则**

$\boldsymbol{K} = \boldsymbol{R}_{\mathrm{Q}} \begin{bmatrix} \boldsymbol{S} & \boldsymbol{R}_{\mathrm{P}} \\ \boldsymbol{0} & \end{bmatrix}^{M}$; // Prepare the SVD input matrix $\boldsymbol{K} \in \mathbb{R}^{c \times c}$, cf. Eqn. 19

$[\boldsymbol{U}', \boldsymbol{s}', \boldsymbol{V}'] =$ 奇异值分解$(\boldsymbol{K})$; // Local SVD yields $\boldsymbol{U}'$, $\boldsymbol{V}' \in \mathbb{R}^{c \times c}$ and $\boldsymbol{s}' \in \mathbb{R}^{c}$

如果 自适应**则**

q $=$ 自适应截断$(\boldsymbol{s}')$; // Adapt the truncation rank

并行广播$(\boldsymbol{U}', \boldsymbol{s}', \boldsymbol{V}', q)$; // Globalize the results

$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{V} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix}$; // Prepare $\boldsymbol{V}$ Update via $\boldsymbol{R} \in \mathbb{R}^{d \times l}$, cf. Eqn. 13

释放分配$(\boldsymbol{U}, \boldsymbol{s}, \boldsymbol{V})$, 分配$(\boldsymbol{U}(\mathrm{Y}\ \ l)\ \ \boldsymbol{s}(l)\ \ \boldsymbol{V}(d, l))$; // Increase the $\boldsymbol{U}$, $\boldsymbol{S}$, and $\boldsymbol{V}$ sizes

$\boldsymbol{V} = \boldsymbol{R}\ \boldsymbol{V}'$(:,1:$l$); // Update $\boldsymbol{V}$, cf. Eqn. 13

$\boldsymbol{s} = \boldsymbol{s}'$(1:l); // Update $\boldsymbol{s}$, cf. Eqn. 14

$\boldsymbol{U} = \boldsymbol{Q}_{Q}\ \boldsymbol{U}'$(:, 1:$l$); // Update $\boldsymbol{U}$, cf. Eqn. 15