arXiv:2301.03192v1 [physics.flu-dyn] 9 Jan 2023

分区大涡模拟的时间精确入口耦合

Marcel Blind[1*]、Johannes Kleinert[1]、Thorsten Lutz[1]和 Andrea Beck[2,1]

[1]斯图加特大学空气动力学与气体动力学研究所，Pfaffenwaldring21，斯图加特，70569，德国。[2]马格德堡奥托·冯·格里克大学流体动力学与热力学研究所，大学广场2号，马格德堡，39106，德国。

*通讯作者。电子邮箱：blind@iag.uni-stuttgart.de；贡献作者：kleinert@iag.uni-stuttgart.de；lutz@iag.uni-stuttgart.de；beck@iag.uni-stuttgart.de；

摘要

在分区大涡模拟（zLES）中，生成湍流入口数据是一项颇具挑战的任务，通常依赖预先定义的直接数值模拟数据来生成统计特性正确的合成湍流。更准确但也更复杂的替代方法，是使用先导仿真的瞬时数据。将瞬时数据用作入流条件，可对飞机等对象的子域开展高保真模拟，涵盖所有非定常事件或罕见事件。本文介绍了一种工具链，能够在采用不同离散格式的流动求解器之间交换高分辨率时空数据。为此，我们采用适用于散乱数据的插值算法进行空间插值；在时间维度上，则针对每个自由度使用一维插值格式。结果表明，我们能够获得稳定的仿真，并将源数据中的所有流动特征映射到新的目标域。因此，该耦合方法能够将任意数据分布和格式映射到新的区域，同时还原并保持湍流结构和尺度。根据目标域所用数值格式的分辨率要求，即可确定所需的时间和空间分辨率要求。

关键词：DGSEM、瞬时入流条件、耦合、分区大涡模拟

1 引言

随着计算性能的提升，大涡模拟（LES）在现代计算流体力学（CFD）研究中日益普及 [1]。然而，许多实际应用仍无法采用这一技术进行详细研究。因此，如今许多仿真都基于所谓的分区方法：这些方法通常依赖预先定义的直接数值模拟（DNS）数据，以生成具有正确统计特性的合成湍流。所谓分区，是指仅使用大涡模拟方法仿真区域中的一小部分，以降低计算成本，而区域中的大部分则采用计算成本低得多的雷诺平均纳维–斯托克斯（RANS）方法求解；或者为该子区域设置合适的边界条件，尤其是尺度分辨来流数据。这种做法在湍流边界层仿真中尤为有用，因为我们无需仿真初始转捩过程，只需以充分发展的边界层作为起始状态。为此，人们开发了许多方法，例如合成涡方法 [2, 3]或回收-重缩放方法 [4, 5] ，从而能够显著缩小计算区域。一个实际例子是对翼型后缘气动噪声进行仿真，此时只需对翼型的一部分进行详细仿真 [6]。上述应用的一个共同点是，它们依赖边界层特性，因此需要文献中的参考数据。

另一个略有不同的例子是研究来流湍流尾流与边界层的相互作用。例如，可以研究飞机的湍流机翼尾流与其水平安定面的相互作用（参见图1）。所述情形颇具挑战性，因为完全能够分辨尺度的代码通常无力计算整架飞机，而能够仿真整架飞机的代码往往又无法对其局部进行高保真模拟。因此，需要以时间精确的方式，将一个仿真的结果映射为针对更精细仿真所需的来流/初始条件；该精细仿真的区域更小，并将映射结果作为来流条件施加。这种方法能够对特别关注的区域进行精细仿真。此外，这些代码之间的耦合还为高效地进一步研究不同物理尺度的湍流相互作用提供了途径。因此，高雷诺数流动的分区模拟成为可能。

尝试耦合不同的数值代码时，我们会遇到若干关于如何处理的问题：

1. 是否必须采用双向耦合，还是单向耦合就足够？2. 我们
是否在运行时对代码进行耦合？3. 底层数值方法是否兼容？

第一个问题——结合上述场景，很容易回答。由于我们只关注来流对目标域的影响，因此单向耦合就足够了。我们注意到，根据方程组的不同，通过预设的狄利克雷边界实现单向耦合

由于信息传输仅限于单一方向，因此容易产生误差。不过，假设我们提取的是远离壁面的尾流区域中的流动（对于可压缩纳维–斯托克斯方程而言），便可以证明这种简化是合理的。此外，这种简化消除了大量复杂性，从而能够高效耦合不同的求解器和实验；采用双向耦合则无法实现这一点。

因此，我们可以直接回答第二个问题。让两套代码分别运行，使我们能够在分区模拟的预处理步骤中执行映射，从而消除运行时的瓶颈。此外，这也避免了必须解决高性能计算（HPC）问题所带来的复杂性：两套可能差异很大的代码需要同步运行，并建立高效的并行通信模式。如前所述，这种耦合旨在对某个子域进行详细仿真，也就是说，特别关注的区域也包含在全域模拟中，因此假定该区域的表征足够充分，能够捕捉所有必要的物理现象。我们还必须考虑到，输入的物理信息可能会被截断。因此，我们将研究不同分辨率组合的影响，以量化这一误差。

第三个问题较难回答，因为我们不仅要考虑有限体积法、有限差分法和有限元方法等空间离散方式，还要处理时间离散；在大多数应用中，时间离散采用隐式或显式方法。通常，将解在两种异构表示之间进行映射有两种可选方案：投影方法和插值方法。前者具有（近似）守恒性，但要在时空中的任意网格上确保这一性质，过程繁琐、代价高昂（因为需要执行非局部运算），且灵活性不足。插值方法则放宽了这一限制，使映射过程具有很强的通用性，并可扩展算法，以任意（x，y，z）数据作为输入，并将其映射为兼容的数据格式。因此，映射算法必须能够处理由网格间距和每个自由度的数值效率造成的分辨率差异、任意点以及不一致的时间步。由此看来，插值算法似乎是实现时空中这些特性的不错选择。

我们需要解决的另一个问题是如何处理大型数据集。尽管我们选择了离线耦合，从而无需在原位传输数据，但时间分辨的表面数据或体数据非常占用内存。因此，必须考虑内存管理算法，包括并行化、负载均衡和数据缩减，以保持较低的计算成本。

本文旨在展示，将德国航空航天中心有限体积代码 TAU 的瞬时数据映射到高阶谱元代码 FLEXI 是可行的，并且能够对各子域进行详细仿真。因此，我们希望回答以下研究问题：

1. 是否可以获得一种数据映射，以实现数值稳定的耦合模拟?

2. 需要多大的时间分辨率？3. 空间分辨率（网格和每个自由度的数值效率）差异会如何影响映射误差？

2 数值方法

映射算法的一个重要方面是了解底层数值方法及其尺度解析特性，我们将在下一节中对此进行更详细的评估。

2.1 代码框架

目标数值格式是开源间断伽辽金框架 FLEXI，该框架由斯图加特大学开发，需为其准备数据 [7]。在此方案中，区域被划分为互不重叠的非结构六面体单元。我们可以为单元选择任意多项式次数。这也意味着，每个单元中的解都以多项式表示。

与谱元代码不同，源数据通常是点数据，来自实验或有限体积代码。本文提出的映射算法旨在具有普适性，但针对德国航空航天中心的有限体积代码 TAU 进行了优化。在图 1 中展示了 TAU 的一个典型应用。TAU 采用混合 RANS/LES 方法，例如在存在分离流的情形下，以 RANS模式处理附着边界层，并以 LES模式解析脱体尾流区域。因此，在 TAU 中很难研究机翼尾流对 HTP 边界层的影响。另一方面，FLEXI 能够解析边界层，因此可以量化机翼尾流的影响。因此，能够在 FLEXI 中进行模拟的感兴趣区域在图 1中标出。稍后还将使用 TAU 验证第 4 节中的结果。

2.1.1 TAU

有限体积求解器 TAU 由德国航空航天中心（DLR）开发，并在航空工业中得到广泛应用 [8]。它可在结构网格和非结构网格上求解欧拉方程或雷诺平均纳维–斯托克斯方程。该求解器实现了多种单方程和双方程模型以及雷诺应力模型，用于湍流建模。此外，还可进行大涡模拟或混合RANS/LES模拟。主网格的单元支持六面体、四面体、三棱柱和金字塔单元。计算单元边界处的数值通量时，可使用不同的迎风格式和中心近似。时间积分可选择显式或隐式格式。所得线性系统采用对称高斯–赛德尔格式或下对称高斯–赛德尔格式求解。为加速收敛，使用局部时间步进、残差平滑和多重网格方法。通过区域分解实现并行化，并利用消息传递接口（MPI）进行通信。

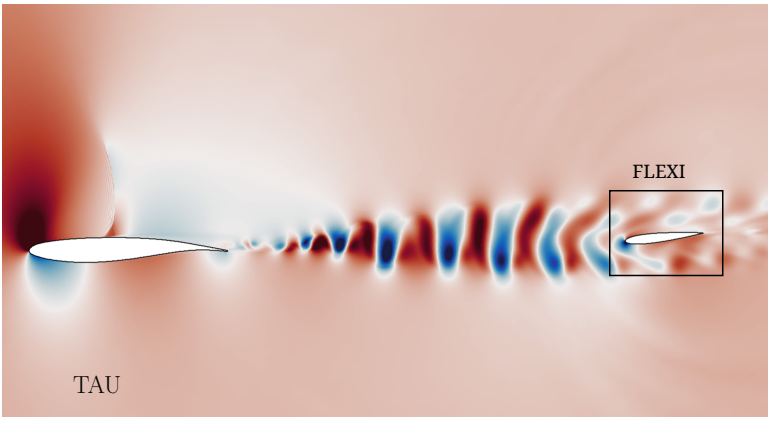


图 1：与 HTP 中仿真的尾流相互作用的串列机翼构型及其可视化机翼尾流。TAU 中的关注区域在 FLEXI 中进行详细仿真，并已突出显示。

2.1.2 FLEXI

FLEXI 是一款基于间断伽辽金谱元方法（DGSEM）的高性能开源计算流体力学求解器。它采用六面体张量积单元，每个单元的多项式次数均可任意设定。由于间断伽辽金方法是结合有限元和有限体积方法的混合格式，我们在单元之间的通量计算中采用耗散最小的 Roe黎曼求解器 [9]。此外，我们在非等距的勒让德–高斯或勒让德–高斯–洛巴托点集上表示多项式解。由于 FLEXI 是一个框架，因此实现了多种方程系统。本文主要使用可压缩纳维–斯托克斯方程。验证时采用线性标量平流系统。应用部分的结果采用可压缩纳维–斯托克斯方程计算，其实现形式为斜对称分裂形式近似，以尽可能减少混叠不稳定性 [10]。边界条件通常以弱形式施加。这意味着我们不规定单元中相应解点处的状态，而是规定数值通量。FLEXI 使用 MPI 实现并行化，并已成功在多达 $\mathcal{O}(10^5)$ 个核心 [11]。

2.1.3 代码框架的比较

间断伽辽金方法是常用的高阶格式。相比之下，有限体积法用于非结构网格时通常仅限于二阶。众所周知，在自由度数量相同的情况下，高阶方法能够达到更低的误差，并且只需更少的解点即可解析相同的结构。这被称为每波长点数 n_{PPW}准则 [1, 12]。高阶方法能够达到 $n_{PPW} \approx 4$，而二阶有限

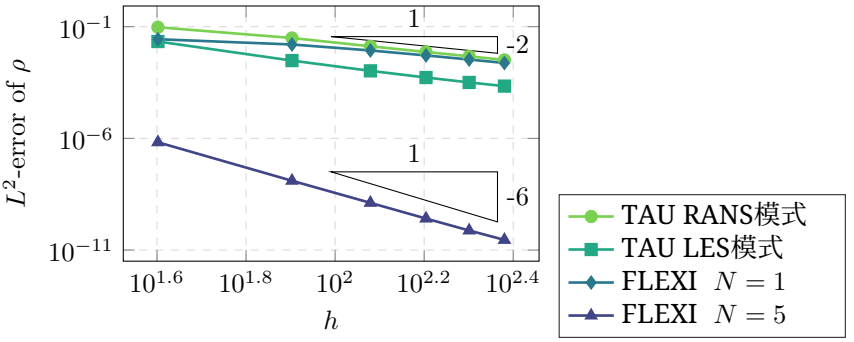


图 2：TAU 和 FLEXI 在不同设置和网格下的收敛行为对比。

体积法通常仅限于 $n_{\mathrm{PPW}} \approx 20$。这意味着，要解析具有给定波长的结构，高阶方法所需的分辨率点数量可减少至五分之一。不过，这一特性高度依赖于所用的多项式次数 N，如图 2所示。因此，在相同网格上，使用 FLEXI 进行的仿真 $N = 5$ 不仅误差下降得更快，而且在给定自由度数量下误差也更小。显而易见，FLEXI $N = 1$ 或 TAU 在 $h = 10^{2.4}$ 网格上的自由度数量与 FLEXI $N = 5$ 在 $h = 10^{1.6}$上的自由度数量相对应。本文采用 Shu 涡测试案例 [13] 开展研究。所有仿真均独立运行，因此未进行映射。

标记为“TAU RANS模式”的结果采用了TAU中混合RANS/LES模拟的RANS区域常用的数值设置。黎曼求解器采用二阶中心通量近似，并通过人工耗散进行稳定化；该近似源自James、Schmidt和Turkel [14]提出的方案。与TAU默认的采用标量耗散的通量平均方案相比，采用矩阵耗散的斜对称格式[15]已能降低耗散水平。标记为“TAU LES模式”的仿真还采用线性梯度外推，在单元面处重构对流通量，以降低斜对称格式的数值色散 [16]。此外，人工耗散系数降低了16倍。这些设置适用于混合仿真中的大涡模拟区域。FLEXI计算采用Roe黎曼求解器 [9]。对于 $N = 1$，FLEXI仿真的结果与收敛阶数相同的TAU计算结果相似。然而， $N = 1$ 通常不用于实际应用，因为在如此低的多项式次数下，高阶格式的优势并不明显。采用 $N = 5$的计算更贴近实际情况，并展现了高阶多项式的优势。

2.2 工作流程

在介绍映射程序之前，我们首先讨论如何使用时间分辨输入数据运行仿真的一般工作流程。该工作流程如图3所示。

首先，必须提供源数据。通常，这些数据可以是逐点散布数据。由于本文重点讨论将 TAU 数据映射到 FLEXI 的过程，因此我们假设从 TAU 获取体积快照或表面数据。

其次，我们通过选择适当的空间映射机制来处理数据。此外，我们还必须决定是仅映射表面数据以获得瞬时边界条件，还是同时获取体积信息，例如为 FLEXI 生成重启文件。该工具会针对每个输入文件创建一个插值文件。结果以 HDF5® 格式保存，该格式采用了与 FLEXI 密切相关的多项式结构。为确保与 FLEXI 兼容，输出多项式次数与随后要运行的仿真的次数相同。

高阶表示通常容易出现混叠和振荡，结果的质量很大程度上取决于所用的点集，以及进行插值时采用的多项式次数。由于输出次数和点集由我们最终要执行的仿真决定，因此无法利用这些参数来减轻误差。因此，我们对目标点表示进行超采样，以避免源点非多项式特性所导致的振荡误差。随后，我们将源数据映射到超采样后的目标数据点。我们发现，使用 $M \approx [1.5, 2] \cdot (N+1)$ 个目标点进行超采样，能够获得良好的一致性，并显著减轻振荡。这与湍流数据过积分的文献取值一致；在 DG 社区中，过积分通常用于去混叠 [17]。将源数据插值到超采样后的目标点后，我们将解投影到原始基$N < M$，从而去除高阶模态信息；这些信息尤其容易受到混叠的影响。

第三，我们对第二步得到的结果进行时间插值。第二步创建的已映射体积或表面文件会转换为一个数据集，其中包含每个解点的时间插值器。插值器由多项式系数组成，这些系数取决于求值时间。此外，我们将数据划分为用户指定数量的子集，以限制每个插值器的数据量，并避免 FLEXI 运行时发生内存溢出。

第四，我们将生成的文件提供给 FLEXI。随后，FLEXI 会在每个时间步对插值函数进行求值，并将相应的边界条件设为插值后的数值。

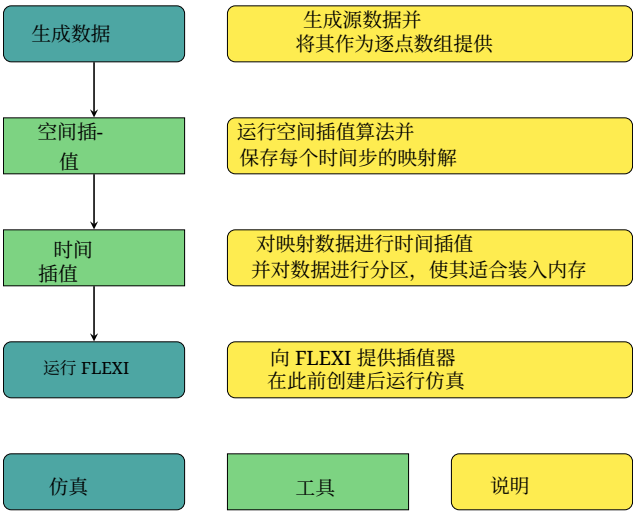


图 3：施加时间分辨边界条件的工作流程。

2.2.1 关于表面数据的一些说明

我们提出的映射算法既可应用于体数据，也可应用于表面数据。根据所提供的数据，空间映射算法将使用体解提取目标边界，或直接使用所提供的表面平面。

TAU 能够将用户定义平面上的二维数据写出，流动变量则通过嵌套网格技术的算法在内部插值到该平面上[18]。当然，这种插值也会引入误差，导致 TAU 的体解与平面上的二维数据不一致。生成的嵌套网格平面可以单独读取。因此，可以大幅简化区域处理，从而降低映射算法的总体成本。

2.3 空间插值

实现 TAU 与 FLEXI 耦合的一个重要步骤是空间映射。由于最终我们要创建瞬时边界条件，因此只需映射表面数据。为使插值方法更加通用，我们实现了一种三维方法，以支持体插值以及源域中任意取向的表面平面。

在 TAU 与 FLEXI 之间建立映射的一大挑战在于二者空间离散方式的差异。工业有限体积代码通常采用四面体网格。因此，直接插值较为困难，因为仅支持六面体的代码所使用的网格数据结构与之不同。与 FLEXI 不同，TAU 将解存储为低阶逐点数据。这会生成包含坐标（x，y，z）和解 $\vec{U}$ [19] 的数组。然而，FLEXI 将每个单元的解存储为多项式数据 [7, 20]。

现在的难点是将 TAU 的逐点数据一致地映射为 FLEXI 中解多项式所需的多项式系数。由于 FLEXI 仅使用非结构六面体单元，而 TAU 还采用四面体，因此我们无法直接使用 TAU 网格信息。图中展示了这种不兼容性的示例，并示意了插值解。4。

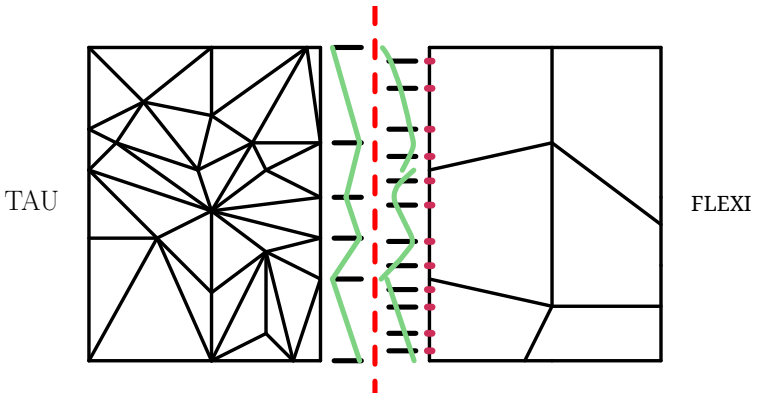


图 4：TAU-FLEXI 接口及不同点集的可视化。浅绿色曲线表示将 TAU 解插值到 FLEXI 点集后得到的近似解。

因此，插值算法必须能够处理散乱数据。这意味着我们面对的是一个点云，其下方嵌有目标网格。因此，必须使用非结构化插值算法，将 TAU 源数据插值到 FLEXI 目标数据。有若干适用于此类任务的算法。

散乱数据插值通常能够达到较高的精度和性能，但高度依赖于源点的分布 [21]。另一方面，实现散乱数据插值例程有助于构建更通用的接口，因为其他代码和解格式也能够轻松实现。

由于源求解器的解数据是预先提供的，因此我们无需在运行时映射数据，从而节省大量计算时间。此外，两套代码使用的网格都是先验已知的（并且在计算过程中保持不变）。不过，良好的性能仍至关重要。因此，我们使用 MPI 并行化实现了界面框架。

通过读入网格，并使用 MPI 在各处理器之间均匀分配单元来完成此操作。单元沿希尔伯特曲线排序 [20]，以最大限度地减少各个处理器之间的通信接口（参见图中红色曲线）。5a不过，这种方法只能用于 FLEXI 目标网格，因为我们需要网格信息才能直接分配数据。对于源数据，我们只读入数据点。

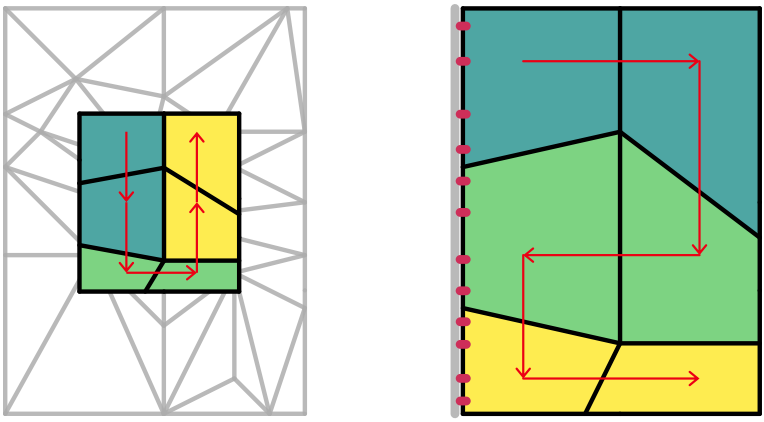
因此，第一步无法简化并在处理器之间分配负载。于是，我们将源数据读入共享内存[11, 22]。随后，各处理器访问共享内存中的源数据，并通过仅选取在各自区域内进行空间插值所需的数据来简化数据。采用这种方法，我们可以节省大量计算时间并减少内存占用，同时不牺牲精度。

如果将表面数据用作空间插值算法的输入，就必须更详细地考虑负载均衡。前文所述的体单元分布并未考虑表面。因此，对于表面映射，我们无法确保分解后的区域各部分都包含需要映射的表面单元（参见图 5a）。因此，尤其是对于处理器较多的小区域，可能并非所有处理器都参与任务，从而导致映射效率低下。因此，我们必须根据表面单元的数量在处理器之间重新分配负载（参见图 5b）。为此，可以在区域分解时为每个表面单元赋予较高权重。该权重根据每个单元中需要映射的边界边数量确定，通常会降低包含此类耦合界面的 MPI 秩上的计算负载。为此，我们首先按常规方式读取网格文件，然后应用表面加权，最后重新初始化网格读取过程 [23]。采用这种方法，可以提升性能，但由于需要重新初始化网格，初始化过程会多花费几秒钟。表面插值的总体成本将低于使用体数据的成本。图 5展示了体数据与表面数据分布之间的差异。

此外，我们还必须确保在每个独立的 MPI 区域周围提供一个缓冲区域，以便为每个点建立插值模板。估算缓冲区域的大小时，需要考虑源点所在完整区域中最大单元的尺寸。由于无法直接得知最大单元的尺寸，因此我们考虑离散点之间的距离，并取其中的最大值。

2.3.1 最近邻插值

最近邻插值会检查源数据坐标，并通过逐点比较，找到距离所需 FLEXI 目标点最近的点。随后，源数据的值 U 会直接存储为 FLEXI 中的节点系数。这种插值会得到分片常数解。另一个要求是源网格均匀分布。如果无法满足这些要求，就有可能产生不良结果。这并不一定意味着结果不符合物理规律，而是指 DG 单元内生成的插值多项式条件不佳，并且可能因跳跃过大而导致映射解出现不自然的振荡。另一个可能观察到的现象是，如果未包含边界节点，目标网格的单元边界上可能会出现跳跃。另一方面，



体映射 表面映射

(a) 目标域完全位于源数据（灰色）之内。区域分解无需考虑表面单元。(b) 边界与表面源数据（灰色）对齐。必须调整 MPI 分布，使三个处理器都包含已映射的表面单元。

图 5：三个 MPI 处理器上，体映射与表面映射的 MPI 区域分解差异。相同颜色表示相同的 MPI 区域。空间填充曲线以红色显示。

如果源数据和目标数据对齐良好，或者网格在界面处重合，这种插值技术则能快速得到良好结果。

2.3.2 反距离加权

一种更通用的方法是使用反距离加权 [24]。目标解通过对源值进行加权平均计算得到

$$u(\vec{x})_{\text{target}} = \frac{\sum_{i=i}^{N_{\text{source}}} \omega_i(\vec{x}) u_{i,\text{source}}}{\sum_{i=i}^{N_{\text{source}}} \omega_i(\vec{x})} \tag{1}$$

其中 N_{source} 表示整个区域内源点的数量。与最近邻方法不同，我们不只考虑一个点，而是考虑源区域内的所有点。权重 $\omega(\vec{x})$ 取决于源点与目标解点之间的距离

$$\omega_i = \frac{1}{\left(\|\vec{x} - \vec{x}_i\|_{L_2}\right)^p} \tag{2}$$

以及加权指数 p。对于 $p \Rightarrow \infty$ 反距离加权方法与最近邻方法类似。Shepard 对一般反距离加权方法进行了改进，提出只考虑目标点周围预定义半径 R范围内的源点 [25]。其表达式如下

$$\omega_i = \left(\frac{\max(0, R - \|\vec{x} - \vec{x}_i\|_{L_2})}{R \|\vec{x} - \vec{x}_i\|_2} \right)^2 \tag{3}$$

其中 R 表示预定义的搜索半径。

2.3.3 径向基函数

非结构化插值的第三种选择是径向基函数 φ [26, 27]。这些方法能够对非结构化数据 s进行高阶精度插值。插值函数由径向基函数的加权和构成。与前面介绍的其他方法不同，我们必须求解线性方程组，以求逆范德蒙德矩阵并确定权重 ω，使其满足

$$s(\vec{x}) = \sum_{i=1}^{N_{\text{source}}} \omega_i \varphi(\|\vec{x} - \vec{x}_i\|_{L_2}) \tag{4}$$

因此

$$u_{j,\text{source}} = \sum_{i=1}^{N_{\text{source}}} \omega_i \varphi(r_{ji}) \tag{5}$$

其中 $r_{ki} = \|\vec{x}_k - \vec{x}_i\|_{L_2}$。我们用矩阵表示法重写插值条件

$$\begin{bmatrix} \varphi(r_{11}) & \varphi(r_{12}) & \cdots & \varphi(r_{1N}) \\ \varphi(r_{21}) & \varphi(r_{22}) & \cdots & \varphi(r_{2N}) \\ \vdots & \vdots & \ddots & \vdots \\ \varphi(r_{N1}) & \varphi(r_{N2}) & \cdots & \varphi(r_{NN}) \end{bmatrix} \begin{bmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \end{bmatrix} = \begin{bmatrix} u_{1,\text{source}} \\ u_{2,\text{source}} \\ \vdots \\ u_{N,\text{source}} \end{bmatrix}. \tag{6}$$

这可以用矩阵形式重写为 $\mathbf{\Phi}_{ij}\vec{\omega}_i = \vec{u}_{j,\ \text{source}}$ 采用指标表示法。由于插值时必须对矩阵 $\mathbf{\Phi}$ 求逆，径向基方法是所介绍的方法中计算成本最高的。

我们计算插值函数

$$u(\vec{x}) \approx \sum_{i=1}^{N} \omega_i \varphi(\|\vec{x} - \vec{x}_i\|_{L_2}) \tag{7}$$

并获取计算域内任意点的值。

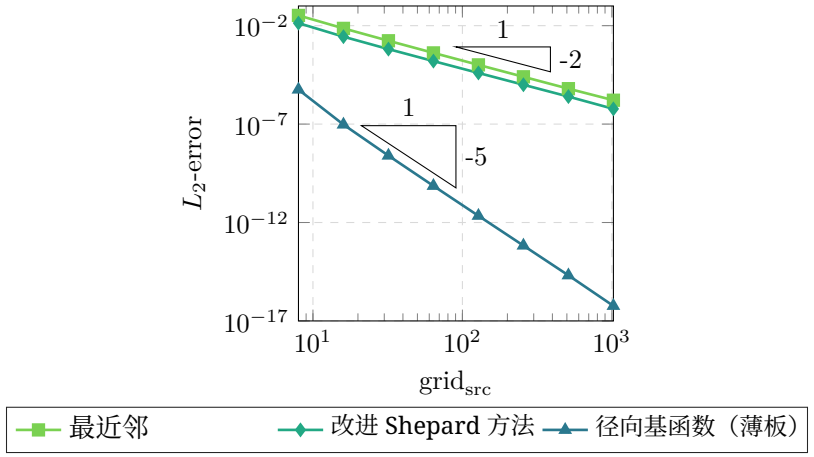


图 6：采用不同插值方法对一维正弦函数进行插值时， L_2误差的收敛情况。

用于插值的典型径向基函数包括多二次函数 $\varphi(r) = \sqrt{1 + }$反多二次函数 $\varphi(r) = \frac{1}{\sqrt{1+(\varepsilon r)^2}}$、高斯函数 $\varphi(r) = e^{-(\varepsilon r)^2}$以及薄板样条函数 $\varphi(r) = r^2 \ln(r)$，其中 $r = \|\vec{x}_j - \vec{x}_i\|_{L_2}$。参数 ε 定义函数的形状，并用于缩放。对于这一应用场景，多二次函数和薄板样条函数被证明是最可靠的径向基函数。由于薄板样条函数不需要任何额外的用户参数 ε ，因此后续研究均采用该函数。

在实现上述算法的过程中，我们得到了一些观察结果。首先，没有一种散乱插值方法的设计具有守恒性。因此，我们对原始变量进行插值，并出于一致性考虑，在映射之后将其转换为守恒变量。

2.3.4 空间插值方法的比较

在评估映射程序中的空间插值程序性能之前，我们先在一个独立测试案例中考察其收敛行为。因此，我们计算了对正弦函数 $f(x) = \sin(2\pi x)$ 进行简单一维插值时的 L_2-误差。

我们在图 6 中绘制了误差随源数据采样分辨率变化的情况。可以看出，与最近邻和反距离加权插值相比，径向基函数插值的结果显然最佳，误差更低，收敛率也更高 $EOC = 5$ ，且反距离加权插值具有 $EOC = 2$。我们将在第 3 节中更详细地评估精度，并比较各种空间插值方案。

2.4 时间插值

除了空间插值之外，我们还必须进行时间插值，以适应源代码与目标代码中不同的时间推进格式。例如，FLEXI 使用显式低存储 Runge-Kutta 方法推进方程组。另一方面，TAU 使用隐式时间离散来实现这一点。然而，除了时间推进格式不同之外，不同仿真中的时间步和模拟数据输出频率也可能有所变化。若要将数据用作瞬时边界条件，我们必须确保能够在任意时刻为目标求解器提供正确的入口数据。因此，必须对空间插值的结果进行时间插值，以获得连续的时间插值器。

与体积与表面映射不同，时间插值完全由一维问题构成。对于一维数据，有大量不同的插值技术。本研究将多项式插值与拉格朗日基相结合，并采用样条插值。

我们采用拉格朗日插值基，因为系数值与解值一致 [28]。此外，借助 FLEXI 中已有的工具，也可以轻松完成求值，因为每个单元中的解由三个一维节点拉格朗日函数的张量积构成。

此外，我们还实现了两种不同的样条插值变体：常见的开放样条和 Akima样条 [29]。与典型样条不同，Akima样条不考虑二阶导数，因此得到的解分布更均匀，与开放样条相比，过冲也更少。对于 $p \geq 2$次多项式插值，同样会出现过冲问题。当隐式源方法与显式目标求解器搭配使用时，这一点尤为重要。在这种情况下，由于显式格式的时间步通常远小于隐式时间步，时间插值需要覆盖大量时间步。因此，过冲可能会显著影响整体映射质量。如果源方法与目标方法的时间步相近，时间插值的影响就会减小。不过需要注意的是，即便如此，仍可能产生过冲。基于上述原因，通常建议采用线性插值或 Akima 插值，以获得最可靠的结果。我们将在第3节展示这一点。

因此，最终的插值质量取决于多个因素。首先，取决于所选的插值方法；其次，取决于所提供的状态或边界源文件的采样率。因此，很难对时间插值产生的误差进行一般性预测。

插值在一个独立工具中完成，不仅限于表面数据，也可以使用任何 FLEXI 仿真的重启文件进行。处理后的结果会保存为 HDF5® 文件，其中包含每个时间采样点上每个多项式的系数。

时间插值程序生成的文件既可直接在 FLEXI 中用于计算插值函数，也可用于生成重启文件，以便在任意时刻继续仿真。

生成的时间插值器包含每个自由度对应的多项式/样条。因此，插值器数组的整体维度比解数组更多（包括多项式系数和时间）。随着维数增加，数组的内存需求也会随之增加。根据可用源数据的数量，可能需要对生成的时间插值函数进行分区，以避免在仿真目标域时发生内存溢出。因此，用户必须为插值器数组指定最大尺寸。随后，插值算法会将数据划分为大小相等的数据集，每个数据集包含一段时间，其长度由用户参数决定。FLEXI 随后只读取包含当前 FLEXI时间步时间信息的数据集。因此，在 FLEXI 运行期间，只需对已保存的插值函数求值，即可获得任意时刻的插值解。

3 界面的验证

本节开始验证映射算法。我们采用循序渐进的方法，先展示概念验证，再介绍时间算法，最后评估界面空间插值算法的收敛行为。

我们先将算法应用于非常简单的测试案例，以此开始评估算法。因此，我们选择线性标量对流方程

$$u_t + \nabla \cdot (au) = 0 \quad \text{with} \quad a \in \mathbb{R} \tag{8}$$

因为它形式简单，且给定初始条件后，其精确解可预先确定。输运速度设为 $a = 1$。我们在各个测试场景中改变初始条件，并在相应章节中进行更详细的说明。源域 $\Omega \in [-1,1]^3$ 和目标域 $\Omega \in [1,3] \times [-1,1]^2$ 的设计使二者在 $x = 1$处共享一个界面。这些测试案例的源数据和目标数据均由 FLEXI 完全生成。源网格采用三重周期性边界条件，目标网格则在 y 和 z方向采用周期性边界条件。在 x方向上，我们在 $x = 1$ 处施加瞬时界面条件，并在 $x = 3$ 处设置流出边界。

3.1 概念验证

我们首先将界面的验证应用于一个非常基础的正弦测试案例，

$$u(x,t) = sin(\pi(x - at)). \tag{9}$$

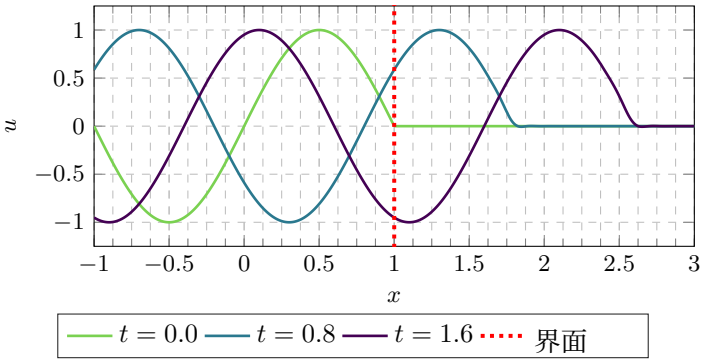


图 7：行进正弦波空间映射过程概览。

初始条件设为 $u_0(x, t = 0)$。精确解 $u(x, t)$ 完全依赖于 x，因此界面平面 $u(x = 1, t)$ 上的数值不会随 y 和 z 变化。目标域初始化为常数解u_0 =const. $= 0$。

在这项初步测试中，我们匹配界面处的表面单元，因此可以使用最近邻插值而不损失精度（源点与目标点重合）。在这种情况下，最近邻算法只需将数据从源域复制到目标域。源网格和目标网格仅在 x方向上偏移了一个区域长度。图 7 中以浅绿色显示了初始条件。还可以看到式（8）在 $t = 0.8$ 和 $t = 1.6$后的解。竖直灰色虚线表示模拟网格的网格线，红色点线表示源域与目标域之间的界面。图形取自等间距笛卡尔立方体的x方向中心线；每个立方体的分辨率为 $16 \times 16 \times 16$，每个单元中使用 $N = 4$ 个多项式。源模拟和目标模拟均采用勒让德–高斯–洛巴托点。此外，我们通过在每个物理时间步 dt对界面数据进行采样，避免了时间插值。

在图 7 中，我们可以看到，所呈现的总体工作流程按预期运行，信息以 $a = 1$ 的方式通过界面传递。由于在此测试案例中我们没有以任何方式对数据进行插值，因此预期源正弦波与目标正弦波之间的总体误差相近。在源域中，L_2 误差为 9。6506E−7，此时 $t = 2$。目标域中的 L_2 误差在 $t = 2$ 后以相同方式计算，结果为9。6859E−7。这有力地证明，该工作流程能够映射数据而不会造成任何信息损失。需要强调的是，整个框架的运行方式就如同我们在两个异构求解器之间进行耦合，唯一的区别是源点与目标点重合。

请注意，我们在源区域和目标域之间使用了连续的初始条件。我们发现应避免出现跳跃或较大的差异

由于目标域入口处产生的非物理扰动进一步传播，源区域与目标域之间的初始条件会出现偏差。不过，这并非界面映射算法所致，而是高阶格式本身的特性。在实际应用中，尤其是瞬态模拟中，这不会造成问题，因为所有从自由来流开始形成的结构都会被映射到目标域中。

3.2 评估时间插值与采样

接下来，我们评估时间插值/采样对界面映射过程的影响。因此，我们研究不同时间插值格式和采样率对入流解的影响，并通过瞬时边界条件将其映射。对于此测试案例，我们为线性平流方程选择了不同的精确解和初始条件 8。为评估采样的影响，我们选择了一个包含间断的初始条件，以便直观呈现信息损失。因此，我们采用

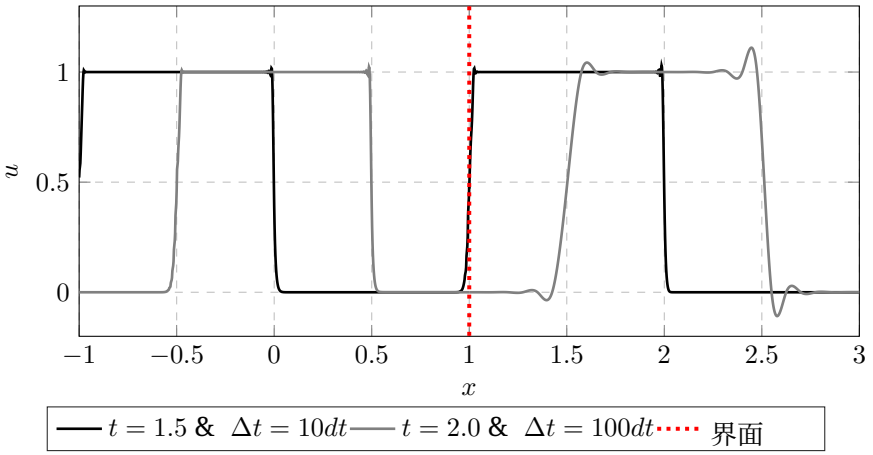
$$u(x,t) = \begin{cases} 1. & \text{if} \quad -0.5 < x - at < 0.5 \\ 0. & \text{else} \end{cases} \quad . \tag{10}$$

我们使用勒让德-高斯-洛巴托节点，节点数为 $N = 4$，用于 $256 \times 1 \times 1$源域和目标域。界面处的表面单元也再次进行了匹配。因此，我们可以使用最近邻插值对表面数据进行空间插值，而不会损失精度（将源数据的值复制到目标位置）。图 8a展示了在两个离散时刻、采用不同Δt插值函数进行求值的仿真。界面位于 $x = 1$ 处，间断以 $a = 1$的速度穿过红色虚线界面，向右侧的目标域传播。

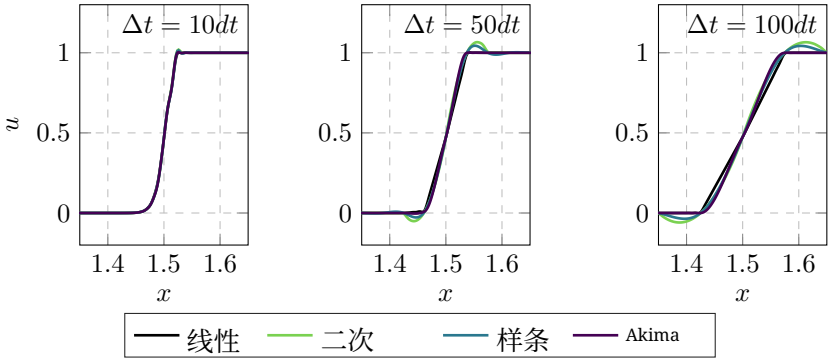
从图 8a的概览图中，我们已经可以看出两种插值得到的跳跃存在显著差异。本测试共选用了三种采样率：一种较高的采样率，为 $\Delta t \approx 10dt$，约为 FLEXI显式时间步的十倍 dt；另外两种较低的采样率分别为$\Delta t \approx 50dt$ 和 $\Delta t \approx 100dt$。图 8a 仅展示了最高和最低采样率。图 8b 更详细地展示了同一求值时间 t 下三种采样率对应的跳跃。由于 x轴采用相同的比例尺，可以看出时间映射对目标FLEXI模拟的影响非常大。图中可以看到两个主要效应：

1. 两个采样点之间的时间间隔会影响跳跃的斜率；2. 时间插值方法会影响跳跃表示的质量。

第一个观察结果只能归因于时间插值，因为激波在源域中的斜率更陡。此外，我们看到，降低 Δt 会再次增大斜率。因此，Δt 的选择必须确保能够充分表示数据中的最大梯度。不过，这在很大程度上取决于具体问题，需要了解数据。



（a）使用样条插值的跳跃测试算例模拟域概览。图中更详细地展示了跳跃，见8b。



(b) 插值技术可视化的跳跃的详细视图，见 8a。

图8：不同 Δt 和时间插值算法下跳跃测试算例的概览图及详细图。

在第 4 节中，我们评估了一种在湍流涡旋背景下确定这一点的方法。

对于第二点，可以给出更一般的结论，因为这一观察几乎独立于 Δt，且仅在Δt 足够大时才会更加明显。高阶多项式近似往往会产生振荡，尤其是在时间采样所采用的等间距点分布情况下。因此，图 8 中仅展示了二次及以下的多项式插值。此外，众所周知，样条插值在 Δt较大时也容易产生振荡。因此，线性插值或 Akima 插值最为理想，能够最好地表现垂直跃变并还原最大梯度。在这种情况下，高阶多项式插值和样条插值主要因为

我们采样的物理时间步 dt 大致等间隔。因此，使用时间上等间隔的点集进行插值时，会出现所谓的龙格现象。这一点至关重要，因为 TAU 的输出频率仅由其隐式时间步决定。因此，目标求解器 FLEXI 必须在每个显式时间步恢复数据。然而，在源采样率固定的情况下，减小目标时间步并不会进一步增大误差，因为插值函数仅由源数据的采样率决定，且只在 FLEXI 运行时进行求值。

由于 Akima 插值在梯度较陡时能得到略为平滑的结果，因此在以下所有测试案例中，我们都采用 Akima 插值。

3.3 空间映射的收敛性

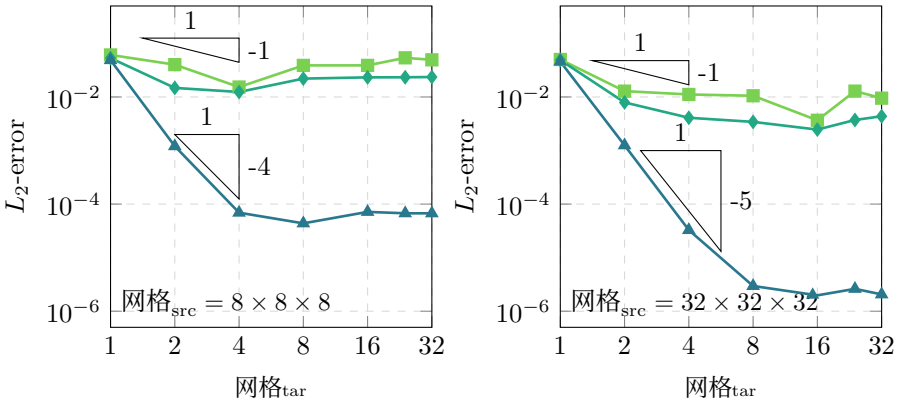
另一个需要考虑的重要方面是映射过程的收敛行为。为衡量插值例程的影响，我们决定计算整个映射和仿真过程的误差范数。因此，该误差包括空间插值误差和时间插值误差，以及在 FLEXI 中施加瞬时边界条件所带来的误差（例如离散误差）。

为了进行收敛性测试，我们修改了图中一维正弦波的初始条件。7。我们在精确解中加入 y 和 z 依赖关系u ，以便界面平面上的 u 取值发生变化。因此，我们得到

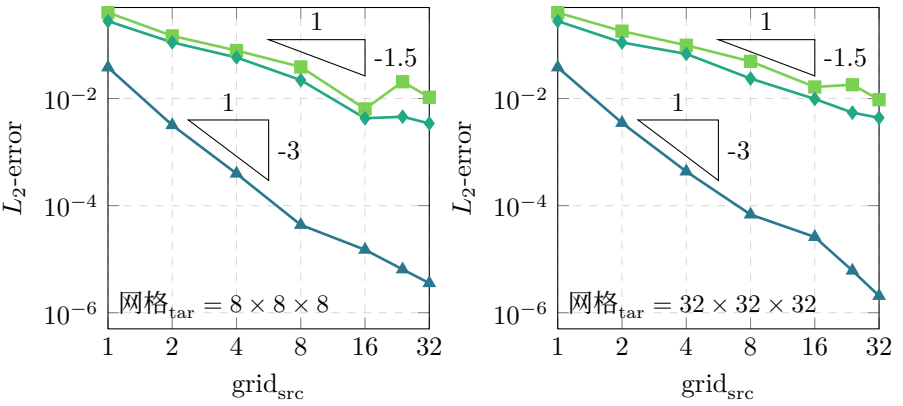
$$u(x, y, z, t) = sin(\pi(x - at)) + sin(\pi y) + sin(\pi z). \tag{11}$$

对于正弦波和线性输运，我们先前已经看到，如果不涉及空间和时间插值（只需将源中的值复制到目标域），就能在目标域中恢复精确解，并且信息会通过瞬时边界条件正确传播。因此，我们希望研究不同非匹配界面（点集和分辨率）对仿真误差的影响；为此，我们首次结合空间和时间插值技术。我们再次在源域和目标域中使用勒让德-高斯-洛巴托点， $N = 5$ 。此外，我们使用超采样， $N = 8$。对于线性输运，这并非必要，因为与纳维-斯托克斯方程不同，这里不会出现混叠。不过，我们希望尽可能贴近后续应用来评估收敛性，并且避免在任何情况下都匹配所有自由度（$N_{\mathrm{src}} = 5 \neq 8 = N_{\mathrm{tar,\ super}}$）。

在图9 中，我们可以看到两种不同的测试情形。图9a 展示了 L_2误差随目标分辨率提高而变化的情况，此时源网格固定。图9b则展示了误差随源分辨率提高而变化的情况，此时目标网格固定。这里的“网格”指笛卡尔立方体每个空间方向上的单元数量。采样时间步长定义为源数据的物理时间步 dt 。因此，例如对于源网格“1”，我们每个物理时间步都提取界面数据，并使用 Akima



(a) 两种不同源网格下线性标量对流方程组的收敛行为。



(b) 两种不同目标网格下线性标量对流方程组的收敛行为。

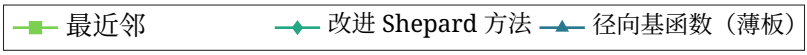


图 9：整个映射过程（包括空间映射、时间映射和瞬时边界条件）的收敛行为比较。

插值将其插值到“32”目标网格的物理时间步，该时间步比前者小 32 倍。

在图 9a 中，我们评估了目标网格分辨率变化对总体误差的影响。源网格的分辨率固定为 $8 \times 8 \times 8$ 和$32 \times 32 \times 32$。我们预计总体误差会收敛，因为如果误差主要由源数据决定，就无法再进一步减小。因此，我们可以看到源数据对目标域的影响。对于 $8 \times 8 \times 8$ 源网格，我们可以很好地观察到这一现象。从

在网格$_{\mathrm{tar}} \approx 4$ 处，我们看到，对于所有插值算法，误差都不再进一步改善。对于更精细的源网格，我们可以观察到类似的行为，但总体误差更低，且误差较晚才达到收敛。由于空间插值引入了误差，在源网格分辨率达到之前，我们无法观察到误差下降。

在图 9b 中，我们研究了在固定目标网格上改变源网格的影响。这可以理解为，对于给定目标域，映射输入质量的提高。在该测试案例中，我们再次评估了两个固定目标分辨率下的影响，分别为 $8\times8\times8$ 和 $32\times32\times32$。随着源网格分辨率的提高，最近邻、Shepard 以及径向基函数插值均表现出不断下降的误差。这一次，直到达到目标网格分辨率之前，都可以观察到近似线性下降的误差。然而，尤其对于网格$_{\mathrm{tar}} = 8\times8\times8$ 案例，我们可以看到，径向基函数插值能够从分辨率高于目标网格的源网格中恢复信息。Shepard 和最近邻在此处的性能明显较弱，并且在这一源网格区间内，误差曲线的斜率不断变化。

总体而言，我们可以对测试的三种空间插值技术的性能进行排序。正如预期的那样，最近邻插值的表现最弱，其实验收敛阶为 $EOC \approx 1.2$（见图 9b）。总体而言，Shepard 插值的误差较低，收敛阶大致相同 $EOC \approx 1.5$。然而，如图 9b 所示，即使源分辨率高于目标分辨率，Shepard 插值也能保持误差稳定，而最近邻插值则呈现出不一致的结果。最后，径向基函数插值显然能得到最佳结果，其收敛阶为 $EOC \approx 2.8$。因此，建议使用径向基函数插值。总体而言，图 9b 印证了图 6中的观察结果。不过，图 9b中所有插值方法的收敛率都较低。但定性观察结果相同，且EOC 中的损失对于所有插值技术而言均相当。需要注意的是，图9b 中的测试案例更为复杂，因为它是二维的，而且我们评估的是整个映射过程中的误差；相比之下，图6中评估的是孤立的一维插值。

对于大型源数据集，需要注意的是，求解方程组以获得径向基函数的插值系数，计算成本会非常高；即使在这个简单的测试案例中，径向基函数插值的速度也明显较慢（最多约慢一个数量级），逊于最近邻插值和 S
hepard 插值。

4 结果：圆柱绕流

本节研究雷诺数为 $Re_c = 3900$ [30, 31]时圆柱周围的流动。圆柱直径定义为 c，并用作本研究中的特征长度。区域的展向长度为 c。我们首次将实际的 TAU 表面数据映射到 FLEXI 区域中。

在图 10 中展示了仿真设置。请注意，图 10 中 x_I 的界面平面尺寸与实际仿真中的尺寸并不相符。在此设置中，界面平面的设计确保所有湍流尾迹结构都能被平面捕捉，且尾流的所有涡结构都完全位于界面平面内。

评估界面性能最重要的方面是确定界面位置并定义记录点（也称为探针点）。我们决定在尾流中的位置 x_I 设置两个界面平面，并设置两个记录点 x_P。

我们在 TAU 源网格和附加的 FLEXI 目标网格中采用相同数量的自由度。因此，目标网格（包括界面）的分辨率必须缩小八倍，以适应 FLEXI 较高的多项式次数 $N = 7$。采用这种方法可以最大限度地减少由此产生的误差（参见图 9a）。我们将在后文展示，这一分辨率足以映射此测试案例中出现的所有物理结构。本例中的目标网格是一个简单的盒体，其尺寸与界面相同，具有相同的 y 和 z 尺寸，并带有足够长的 x延伸段，以便湍流尾迹得以发展并向前传播。

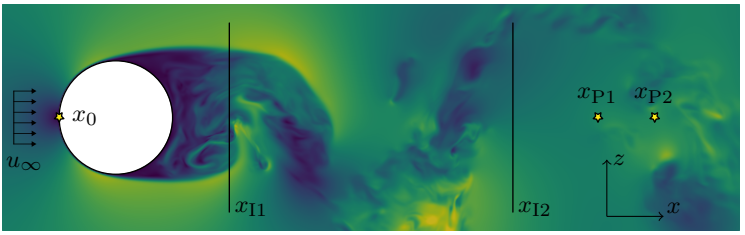


图 10: $Re_c = 3900$ 测试案例定义，包含界面平面和评估探针点处的圆柱。

4.1 模拟设置

接下来，我们将讨论圆柱测试案例的模拟设置，并介绍 FLEXI 和 TAU 各自的设置。

选择圆柱绕流作为主要测试案例的主要原因是，我们有能力使用 FLEXI 和 TAU 对整个区域进行仿真。因此，我们不仅可以将已映射结果与 TAU 的解进行比较，还可以将其与使用 FLEXI 得到的参考直接数值模拟结果进行比较。此外，圆柱是流体力学领域广为人知的几何形状，此前已得到详细研究。

TAU 和 FLEXI 的直接数值模拟参考模拟采用相同的基础网格设置。唯一的区别仍然是，考虑到 FLEXI 在每个单元中使用的高阶多项式，其网格粗化了八倍。因此，如果 FLEXI 以 $N = 7$ 运行，我们就拥有相同数量的

自由度。我们还将在后文考察低阶多项式的解质量。FLEXI 仿真中使用 $N \in [3, 5, 7]$ 多项式。

4.2 分辨率敏感性

首先，我们评估测试案例对分辨率的敏感性。由于我们关注所选分辨率对于直接数值模拟是否足够，因此在 FLEXI 中开展了这项研究。之所以专门选择此测试案例，是因为它可以在 FLEXI 和 TAU 中进行完全解析的仿真。对于入流条件的典型应用，这一点将无法实现。

在图 11 中展示了圆柱尾流中三个离散点处的湍动能谱。每幅图都包含三种仿真的谱，每种仿真采用不同的多项式次数。

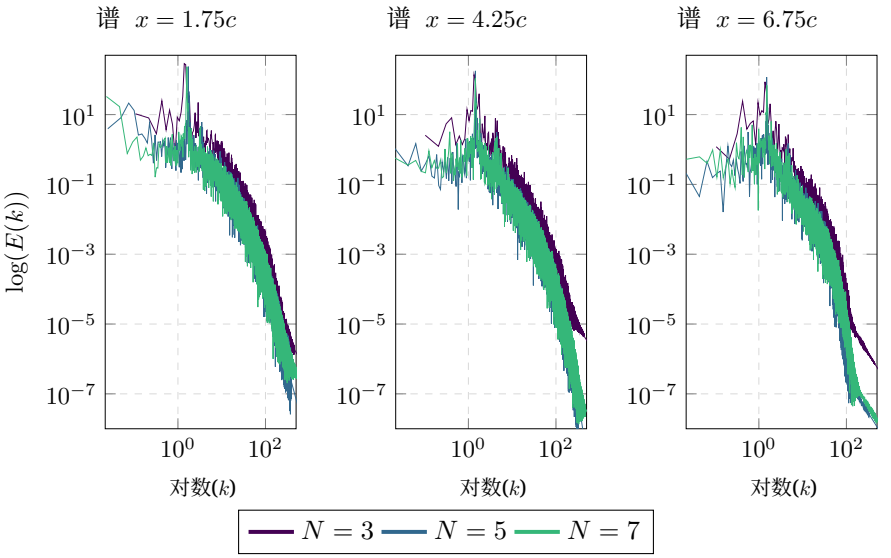


图 11：不同多项式次数下湍动能的比较 N。

图 $N = 3$ 11 中的谱 与 $N = 5$ 和 N =7 条曲线相比，在所有评估位置均显示出偏差。因此，我们可以假设$N = 3$ 的分辨率不足以进行直接数值模拟，且无法产生足够的耗散。由于 $N = 5$ 和 $N = 7$ 的湍动能谱一致，我们可以假设在此分辨率下已达到收敛，因此 $N = 5$ 足以用于直接数值模拟。圆柱的预期斯特劳哈尔频率在谱 [31] 中清晰可见，表现为一个明显的峰值，且适用于所有多项式次数。

采用 $N = 7$ （与 TAU 网格具有相同自由度数量）的仿真对于典型的大涡模拟/直接数值模拟而言过于精细，因为该网格最初是为

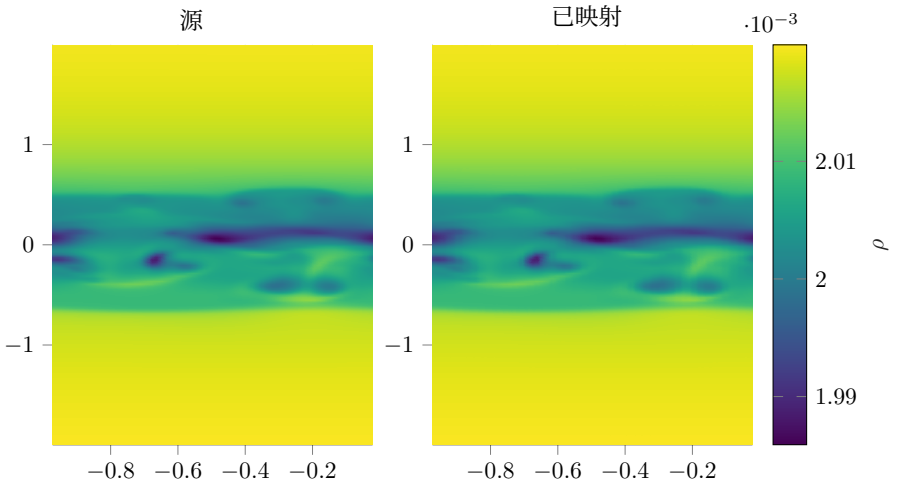


Fig. 12：圆柱尾流状态瞬时流场的比较 .

适用于混合雷诺平均纳维–斯托克斯/直接数值模拟和有限体积代码而创建的。在 FLEXI 中评估黏性壁面间距，得到 $y^+ \approx 0.01$ ，这甚至足以满足直接数值模拟的要求。不过，这也凸显了高阶格式在解析湍流涡旋方面的优势。

如图 2所示，在 FLEXI 和 TAU 中使用相同数量的自由度，并在 FLEXI 中采用较高的多项式次数，展现了高阶格式的优势，因为这一分辨率足以使 FLEXI 进行直接数值模拟。

4.3 插值误差

接下来，我们评估将图中定义的尾流平面进行插值所产生的插值误差。 10 至 FLEXI 边界条件（网格 $_{\text{tar}}$ $=32\times 8$ ）上的误差，所用方法为改进的谢泼德方法。我们研究平面x_{I1}上的误差，该平面距离圆柱最近。我们以 TAU 源数据作为参考数据来评估误差。为使 TAU 和 FLEXI 使用相同的点集，我们在 TAU 解点上计算已映射 FLEXI 解的多项式。结果如图12所示。

从视觉上看，图12 中的结果非常令人信服。源数据中的所有结构都在已映射解中得到了呈现。仔细观察可以发现，已映射解在单元边界处存在轻微过冲。这种效应已通过采用超采样作为去混叠技术得到缓解。为量化误差，我们考察图中所示的源数据与映射数据之间的差异。13。

因此，我们评估了三种分辨率网格下的密度误差$_{\text{tar}}$ $=32\times 8$、网格 $_{\text{tar}}=16\times 4$ 和网格$_{\text{tar}}=8\times 2$。图中显示了基于源数据插值得到的目标数据的相对偏差。密度图证实了我们在图中观察到的情况12 ，并显示

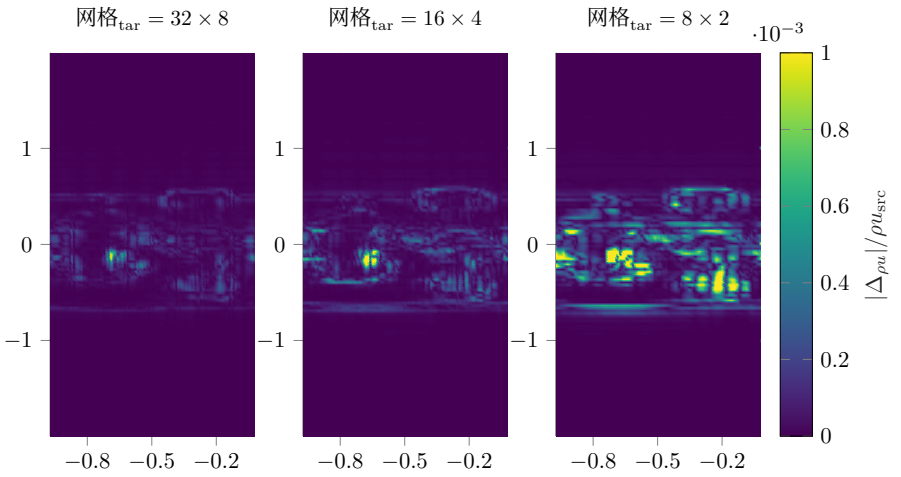


图13：圆柱尾流状态插值误差的比较。

表1：原始变量的最小值、最大值和积分平均值。

		ρ	u	v	w	p
Mean	源	2.014E-03	2.806E+01	2.001E-01	-6.915E-01	1.578E+02
	已映射	2.014E-03	2.803E+01	2.013E-01	-6.888E-01	1.578E+02
Min.	源	1.986E-03	-2.975E+01	-3.402E+01	-2.510E+01	1.553E+02
	已映射	1.986E-03	-2.978E+01	-3.333E+01	-2.508E+01	1.553E+02
Max.	源	2.020E-03	4.800E+01	3.178E+01	2.967E+01	1.583E+02
	已映射	2.020E-03	4.787E+01	3.153E+01	2.932E+01	1.583E+02

整个区域内的误差。分辨率较粗时，误差如预期般增大。计算所有三种网格的 L_2误差后得到 L_2误差$(32 \times 8) = 5.89\text{E}{-7}$,$L_2$误差$(16 \times 4) = 3.03\text{E}{-7}$ 以及 L_2误差$(8 \times 2) = 9.65\text{E}{-8}$，得到的收敛率为 $EOC = 1.31$，这与我们从图9b中得出的关于Shepard方法的发现一致。特别是在界面平面中部包含涡旋的区域，我们在涡旋边界处看到了较大的误差。

在表 1 中，我们可以看到源数据和映射数据中原始变量的最小值和最大值。此外，还列出了积分平均值。可以注意到，映射对于密度和压力给出了非常好的结果。相比之下，尤其是速度方面，与源数据相比存在微小偏差。这种误差源于插值并非守恒。由于速度分量的符号会发生变化，这种影响尤为明显。在插值后将原始变量转换为守恒变量，可确保尽管插值并非守恒，我们仍能得到一致的守恒变量。

因此，鉴于该方案具有灵活性，且对映射结果的影响通常很小，我们可以忽略非守恒性的影响（参见表 1），并直接将映射平面用作入流条件。

4.4 采样率的影响

现在，我们评估源数据的采样率对目标域中解的质量的影响。这是创建耦合模拟时必须考虑的一个非常重要的用户参数。我们通过考察来流湍流对湍动能贡献的影响来进行评估。

图 14 展示了两个不同探针点处的湍动能谱。不同颜色表示不同的时间采样。我们用N_{Skip} 表示时间上跳过的 TAU 快照数量。 $N_{\mathrm{Skip}} = 1$表示使用每一个时间快照。TAU 物理采样率为 ~ 150 个快照/特征时间。特征时间定义为流体流过圆柱直径所需的时间。对于 $N_{\mathrm{Skip}} = 2$ 我们只使用每隔一个快照。颜色越浅，用于恢复 FLEXI 中 TAU 解的快照就越少。

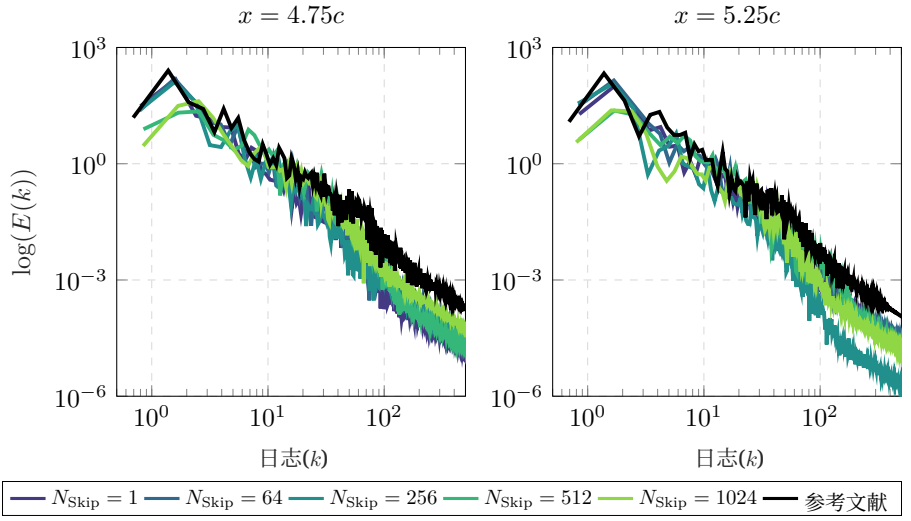


图 14：圆柱尾流中两个不同探针点处随采样率变化的湍动能。

图 14 表明，结果在很大程度上取决于采样率。这似乎是合理的，因为采样率决定了哪些结构会通过瞬时边界条件映射。根据奈奎斯特准则，存在一个 N_{Skip} 值，使得解不再能够被表示。在本例中，当 $N_{\mathrm{Skip}} \geq 512$ 时，我们不再能看到与.....的一致性

参考解。 N_{Skip} 越小，与黑色参考解的一致性越好。因此，可以得出两个主要观察结果。首先，当N_{Skip} 较高时，无法重现主要流动结构，甚至无法正确表示斯特劳哈尔频率。此外，在 $x = 5.25c$处的目标域中经过一段发展后，我们可以看到，即使在 k较低时，结果也存在很大差异。其次，我们可以观察到，能量无法适应；当 N_{Skip}较大时，高阶模态中的能量会损失。

根据这些观察结果，我们可以得出结论：采样频率取决于必须映射到新区域中的结构。因此，我们定义了一种度量，用于量化“涡尺度—采样率”关系；该关系与底层空间离散格式密切相关。根据文献（例如 [1, 10]），我们知道空间离散也有类似的判据，它使用每波长点数参数 n_{PPW} 来量化空间离散格式解析多尺度结构的特性。对于 DGSEM，已知 $n_{\mathrm{PPW,\ DGSEM}} \approx 4$。

在这种情况下，我们以两种尺度作为参考。首先，根据 [32]，大尺度结构的尺寸与圆柱相当，对应于 $L = 1c$。根据模拟设置和 DG 格式的性质，我们估算最小结构的尺度如下

$$l = \frac{L_{\mathrm{domain}}}{\#\mathrm{DOF} \cdot n_{\mathrm{PPW,DGSEM}}} \approx 0.06c. \tag{12}$$

其中 L_{domain} 表示区域的尺寸，而 #DOF 表示用于离散区域的自由度数量。考虑到 u_∞，我们可以估算涡旋扫过界面平面所需的时间，并假设泰勒假设 [33]成立。考虑采样频率后，我们可以估算，对于最小结构，需要 $N_{\mathrm{Skip}} \approx 4$，而对于大尺度结构， $N_{\mathrm{Skip}} \approx 64$ 就足够了。图 $L = c$ 也印证了 14中的这一趋势。每隔 64^{th} 个样本 $N_{\mathrm{Skip}} = 64$取一个样本，仍能保留主要结构并得到正确的幅值；而 $N_{\mathrm{Skip}} > 64$ 则显示出欠分辨率的迹象。根据这些信息，我们可以估算将结构/涡旋输运过界面所需的采样点数。结果表明，对于大、小涡旋，每个涡旋都需要大约 2。3 个样本。正如预期，我们可以得出结论：对于界面，空间和时间离散的要求相似。

我们对两个界面平面 x_{I1} 和 x_{I2}都重复了这项评估。两者的结果在定性上完全一致。

5 总结

本研究提出了一种基于前置模拟生成瞬时边界条件的方法。我们介绍了通过插值处理空间和时间离散差异所需的数值方法，并验证了该方案在简单测试案例和更复杂的圆柱尾流中的有效性。

我们展示了如何利用本文所述的方法生成数值稳定的来流和初始条件。这些方法具有普适性，同样适用于 TAU 以外的求解器，甚至适用于实验数据。

采样率的要求与空间离散的要求相似，因此，根据所采用的时间插值格式，每个波长大约需要四个采样点。

我们实现了多种映射技术，展示了它们在插值质量方面的差异，并进一步说明了它们重构离散源数据的能力。此外，我们对插值进行超采样，以提高整体精度，并减轻混叠和数值不兼容性造成的误差。

对于界面两侧的空间分辨率差异，我们发现，提高源数据的分辨率从未造成问题。然而，过度降低数据分辨率可能会产生较大的混叠误差，从而影响高阶格式。因此，我们建议界面两侧采用相近的分辨率。

现在需要将引入的界面应用于更复杂的场景。本文介绍的工具链已设计为能够处理这类具有挑战性的仿真。

致谢。 作者衷心感谢德国研究联合会（DFG）（德国研究基金会）资助本研究，并将其纳入研究组 FOR2895 的研究框架。我们还感谢高斯超级计算中心 www.gauss-centre.eu为本项目（GCS-lesdg）提供资金支持，并提供在斯图加特高性能计算中心的 GCS 超级计算机 HAWK 上的计算时间（www.hlrs.de）。

参考文献

[1]Flad, D., Beck, A., Guthke, P.：一种采用非线性优化松弛滤波器的 DGSEM 大涡模拟方法。《计算物理学杂志》**408** (2020)。https://doi.org/10.1016/j.jcp.2020.109303

[2] Jarrin, N., Benhamadouche, S., Laurence, D., Prosser, R.：一种用于大涡模拟生成来流条件的合成涡方法。《国际传热与流体流动杂志》**27**(4), 585–593 (2006)。https://doi.org/10.1016/j.ijheatfluidflow.2006.02.006

[3]Jarrin, N., Prosser, R., Uribe, J.C., Benhamadouche, S., Laurence, D.：采用合成涡方法重构混合RANS/LES模拟中的湍流脉动。《国际传热与流体流动期刊》**30**(3), 435–442（2009年）。https://doi.org/10.1016/j.ijheatfluidflow.2009.02.016

[4] Lund, T.S., Wu, X.H., Squires, K.D.：湍流来流数据的生成

用于空间发展边界层仿真。《计算物理学杂志》**140(2)**, 233–258（1998）。https://doi.org/10.1006/jcph.1998.5882

[5] Kuhn, T., Kempf, D., Beck, A., Munz, C.-D.：采用间断伽辽金求解器的分区大涡模拟新型湍流入口方法。已提交至《计算机与流体》（2020）。已提交至《计算机与流体》

[6] Kempf, D., Munz, C.-D.：采用高阶间断伽辽金方法对后缘噪声进行分区混合计算气动声学仿真（2022）。https://doi.org/10.2514/6.2022-3021

[7] Krais, N., Beck, A., Bolemann, T., Frank, H., Flad, D., Gassner, G., Hindenlang, F., Hoffmann, M., Kuhn, T., Sonntag, M., Munz, C.-D.：FLEXI：用于双曲-抛物守恒律的高阶间断伽辽金框架。《计算机与应用数学》**81**, 186–219（2021年）。https://doi.org/10.1016/j.camwa.2020.05.004

[8] Schwamborn, D.、Gardner, A.D.、von Geyr, H.、Krumbein, A.、Lüdecke, H.、Stürmer, A.：《面向航空航天应用的TAU代码开发》。载于：第50届NAL国际航空航天科学与技术会议，印度班加罗尔（2008年）

[9]Roe, P.L.：《近似黎曼求解器、参数向量和差分格式》。《计算物理学杂志》**43**（2），357–372（1981年）。https://doi.org/10.1016/0021-9991(81)90128-5

[10] Gassner, G.J.、Winters, A.R.、Hindenlang, F.J.、Kopriva, D.A.：《BR1格式对可压缩纳维–斯托克斯方程是稳定的》。《科学计算杂志》**77**（1），154–200（2018年）。https://doi.org/10.1007/s10915-018-0702-1

[11]Blind, M., Kopper, P., Kempf, D., Kurz, M., Schwarz, A., Beck, A., Munz, C.-D.：使用间断伽辽金框架 FLEXI 提升大规模模拟的性能。已提交至《科学与工程中的高性能计算'22》。Springer，Cham（2022）

[12] Gassner, G., Kopriva, D.A.：高斯和高斯-洛巴托间断伽辽金谱元方法的色散与耗散误差比较。SIAM科学计算杂志 **33**(5), 2560–2579（2011年）。https://doi.org/10.1137/100807211

[13] Spiegel, S.C., Huynh, H.T., DeBonis, J.R.：《等熵欧拉涡旋问题的高阶方法综述》（2015年）。https://doi.org/10.2514/6.2015-2444

30 分区大涡模拟的时间精确入口耦合[14] Jameson, A.：Jameson–Schmidt– Turkel 方案的起源及进一步发展。《AIAA 期刊》**55**(5), 1487–1510 (2017年)。https://doi.org/10.2514/1.J055493[15] Swanson, R.C., Turkel, E.：关于中心差分格式和迎风格式。《计算物理学杂志》**101**(2), 292–306 (1992)。https://doi.org/10.1016/0021-9991(92)90007-L[16]Löwe, J., Probst, A., Knopp, T., Kessler, R.：适用于非结构有限体积流动求解器的低耗散、低色散二阶格式。《AIAA 期刊》**54**(10), 2961–2971 (2016)。https://doi.org/10.2514/1.J054956[17] Flad, D., Beck, A., Munz, C.-D.：采用高阶间断伽辽金谱元法，通过自适应滤波仿真欠分辨湍流。《计算物理学杂志》**313**, 1–12 (2016)。https://doi.org/10.1016/j.jcp.2015.11.064

[18]Burggraf, U., Kuntz, M., Schöning, B.: 第13章：在非结构化德国航空航天中心有限体积代码 Tau 中实现嵌套网格法。《数值流体力学札记》（NNFM），第93–100页。Vieweg+Teubner 出版社，威斯巴登（1999）。https://doi.org/10.1007/978-3-663-10901-3_13_

[19] NetCDF：NetCDF 经典格式规范（2022）。https://docs.unidata.ucar.edu/netcdf-c/current/file_格式_规范.html[20] Hindenlang, F.：非结构网格上高阶并行仿真的网格弯曲技术。学位论文（2014）[21]Laughton, E., Tabor, G., Moxey, D.：非共形高阶间断伽辽金方法的插值技术比较（2020）。https://arxiv.org/pdf/2007.15534[22]Kopper, P., Copplestone, S., Pfeiffer, M., Koch, C., Fasoulas, S., Beck, A.：基于 MPI-3 共享内存的欧拉–拉格朗日模拟混合并行化（2022）。https://ui.adsabs.harvard.edu/abs/2022arXiv220313840K[23] Appel, D., Jöns, S., Keim, J., Müller, C., Zeifang, J., Munz, C.-D.：水平集虚拟流体法的基于窄带的动态负载均衡方案。Springer，Cham（2022）。即将出版[24] Mueller, T.G., Pusuluri, N.B., Mathias, K.K., Cornelius, P.L., Barnhisel, R.I., Shearer, S.A.：普通克里金法和反距离加权插值的地图质量。《美国土壤科学学会期刊》**68**（6），2042–2047（2004）。https://doi.org/10.2136/sssaj2004.2042

[25]Shepard, D.：用于不规则间隔数据的二维插值函数（1968年）。https://doi.org/10.1145/800186.810616[26] Hardy, R.L.：地形及其他不规则表面的多重二次方程。《地球物理研究杂志》**76**（8），1905（1971年）。https://doi.org/10.1029/JB076i008p01905[27] Hardy, R.L.：多重二次双调和方法的理论与应用——1968至1988年的二十年探索。《计算机与数学应用》**19**（8-9），163–208（1990年）。https://doi.org/10.1016/0898-1221(90)90272-L[28] Kopriva, D.A.：偏微分方程谱方法的实现：面向科学家与工程师的算法。科学计算。Springer，多德雷赫特（2009年）[29] Akima, H.：基于局部过程的插值与光滑曲线拟合新方法。《美国计算机学会会刊》**17**（4），589（1970年）。https://doi.org/10.1145/321607.321609[30] Kravchenko, A.G., Moin, P.：对雷诺数为下圆柱体绕流的数值研究。《流体物理学》**12**（2），403–417（2000年）。https://doi.org/10.1063/1.870318[31] Parnaudeau, P., Carlier, J., Heitz, D., Lamballais, E.：雷诺数为3900时圆柱体绕流的实验与数值研究。《流体物理学》**20**（8）（2008年）。https://doi.org/10.1063/1.2957018[32] Pope, S.B.：《湍流》，第12次印刷版。剑桥大学出版社，剑桥（2015年）[33] Moin, P.：重新审视泰勒假设。《流体力学杂志》**640**，1–4（2009年）。https://doi.org/10.1017/S0022112009992126