arXiv:2303.08513v1 [math.NA] 15 Mar 2023

分区式流固耦合模拟中每个耦合步的子问题迭代次数

Thomas Spenke[a, 1, *], Nicolas Delaiss´e[b, *], Joris Degroote[b, c], Norbert Hosters[a]

[a]技术系统计算分析讲席（*CATS*）、仿真与数据科学中心（*JARA-CSD*）、亚琛工业大学，*Schinkelstraße 2*，德国亚琛[b]机电、系统与金属工程系，根特大学，*Sint-Pietersnieuwstraat 41*，比利时根特[c]*Flanders Make @ UGent – MIRO* 核心实验室

摘要

在文献中，分区式流固耦合方案的成本通常以每个时间步所需的耦合迭代次数来评估，而忽略了非线性子问题内部的迭代。本文表明，这些内部迭代会显著影响耦合模拟的计算成本。我们特别关注如何通过限制每次求解器调用中的迭代次数来缩短总运行时间，因为这样可以避免利用尚未收敛的耦合数据来精细调整子问题解。基于系统的参数研究，我们探讨每个耦合步的最优子问题迭代次数。

最后，本文针对耦合系统提出了一种新的收敛准则，该准则基于子问题的残差，因此无需为耦合循环设置额外的收敛容差。

关键词：流固耦合、分区算法、求解器迭代、耦合问题

1. 引言

在求解流固耦合（FSI）问题时，分区方法得到广泛应用，因为它们能够重用成熟可靠、针对两个子问题分别设计的求解器 [1]。流动求解器和结构求解器被视为黑箱，所有数据交换都局限于共享界面。分区技术的主要缺点是需要进行迭代耦合循环，在每个时间步内反复求解子问题，以确保界面上的平衡条件得到满足。此外，由于附加质量效应 [2–4]，这些耦合迭代容易出现稳定性问题。

在过去二十年间，人们对这些失稳进行了广泛研究 [2, 5–8] ，并提出了多种用于稳定和加速耦合的技术。最简单的方法是对界面数据进行松弛，这虽然能提高稳定性，却要以收敛缓慢为代价。像艾特肯松弛 [9–11] 这样动态更新松弛因子的方法缓解了这一缺点，但仍将所有误差分量一视同仁；然而，研究[12, 13] 表明，附加质量失稳仅由正确解与迭代耦合所得解之间误差的最低波数分量引起。

这一认识为拟牛顿技术开辟了道路。这类技术基于先前求解器调用的输入输出数据，采用雅可比矩阵的低秩近似，通过牛顿–拉夫森方法更新子问题之间交换的界面数据。多年来，人们提出了许多不同的变体。

*贡献相同，应视为共同第一作者电子邮箱： spenke@cats.rwth-aachen.de (Thomas Spenke), nicolas.delaisse@ugent.be (Nicolas Delaiss´e),joris.degroote@ugent.be (Joris Degroote), hosters@cats.rwth-aachen.de(Norbert Hosters)[1]通讯作者

这些方法的区别在于界面数据的更新时间，即在其中一个求解器运行后更新，还是在两个求解器都运行后更新；也在于采用何种技术近似雅可比矩阵，例如最小二乘法或多向量方法 [14]。[15]中给出了全面综述。

在文献中，耦合算法的效率通常根据每个时间步所需的耦合迭代次数来评估，其依据是子问题的求解远远是耦合方案中成本最高的部分。然而，尽管从未明确指出，这种成本度量也意味着每次耦合迭代（因此也包括每次求解器调用）的计算成本都是恒定的。

本研究表明，实际情况并非如此。由于子问题通常具有非线性，因此需要通过定点迭代或牛顿迭代建立各自的内部迭代循环来处理这种非线性。这些内部子问题迭代会显著影响求解器调用的计算成本。举例来说，设想两次固体求解器调用：第一次执行十次牛顿迭代，第二次只执行一次；显然，预计第二次调用耗时要少得多，尽管很可能不会少到只有十分之一。在此背景下，本研究提出一种新的成本度量，不仅考虑耦合迭代次数，还考虑两个求解器各自的子问题迭代次数，并将它们合并为加权和。数值结果证实，与仅考察耦合迭代次数相比，这种新度量能更准确地反映实际运行时间。

迄今为止，子问题迭代对计算成本的影响一直被忽视，因此文献中虽有大量旨在减少耦合步数以实现收敛的技术，但据作者所知，尚无研究探讨如何尽量减少子问题迭代次数。为开始弥补这一空白，本文考察了限制每次求解器调用所执行的迭代次数所产生的影响，并证明每次求解器调用时不必收敛至最终子问题容差，即可显著加快计算速度。这主要是因为，只要分区方案尚未收敛，就可以避免花费时间精化一个将在下一次耦合迭代中被覆盖的初步子问题解。另一方面，返回不精确的结果可能会损害耦合循环的稳定性，以及拟牛顿方法所用输入输出数据的质量。

无论如何，在每个时间步结束时，要准确求解耦合问题，就要求两个子问题都收敛至各自的求解器容差。每次求解器调用都迭代至完全收敛时，这一条件自然得以满足；但若限制每次调用的子问题迭代次数，则必须对其进行显式监测。因此，本文提出了一种新的收敛准则，仅依据求解器残差来评估耦合方案的收敛情况。此外，这种新方法无需像文献中通常那样选择一个不太直观的耦合容差值，并在耦合精度与各子问题精度之间建立了自然联系。

由于在现代工程科学中，采用不同的离散化方法求解子问题十分常见，因此本文讨论了通过不同的分区式流固耦合框架获得的数值结果，以拓展研究范围。第一个框架对两个子问题都采用有限元，而第二个框架则对流动问题采用有限体积法，对结构问题采用有限元。

本文其余部分安排如下。第 2 节介绍流固耦合问题；第 3 节讨论采用有限元或有限体积法求解子问题，并重点探讨这两种技术如何处理非线性。这对于第 4 节至关重要，该节提出一种新的代价函数，同时考虑耦合迭代次数和子问题迭代次数。此外，该节还讨论每次求解器调用中的子问题迭代次数对这一成本度量的影响，并提出新的收敛准则。针对两个测试案例——顶盖驱动方腔和柔性管案例，采用两个框架进行计算，并在第 5 节讨论结果，最后在第 6 节给出结论。

2. 分区式流固耦合

本文研究的大部分问题预计会影响以分区方式求解的任何多场问题。不过，我们将研究范围限定为 $\Omega^f \in \mathbb{R}^{n_{sd}}$ 域内不可压缩流体与 $\Omega^s \in \mathbb{R}^{n_{sd}}$ 内弹性动力学固体之间的相互作用，其中n_{sd}为空间维数。

2.1. 不可压缩流

流动　速度$\mathbf{v}^f$（$\mathbf{x}$，t）和流体压力p_f（$\mathbf{x}$，t）由非定常纳维-斯托克斯方程控制

$$\frac{\partial \mathbf{v}^f}{\partial t} + \mathbf{v}^f \cdot \boldsymbol{\nabla}\mathbf{v}^f - \frac{1}{\rho^f}\boldsymbol{\nabla}\cdot\mathbf{T}^f = \mathbf{b}^f \qquad \text{in } \Omega^f \ \ \forall t \geq 0\,, \tag{1a}$$

$$\boldsymbol{\nabla}\cdot\mathbf{v}^f = 0 \qquad \text{in } \Omega^f \ \ \forall t \geq 0\,, \tag{1b}$$

其中流体密度 ρ_f为常数，外部体力为 $\mathbf{b}^f$。对于不可压缩牛顿流体，柯西应力张量表示为 $\mathbf{T}_f(\mathbf{v}_f, p^f) = -p^f\mathbf{I} + \mu_f\left(\nabla\mathbf{v}_f + (\nabla\mathbf{v}_f)^T\right)$，其中 μ_f为动力黏度。通过在 $\partial\Omega^f$ 上施加一组合适的边界条件以及指定一个无散初始速度场，即可封闭该问题。

2.2. 弹性固体

固体位移场 $\mathbf{d}^s(\mathbf{x},t)$ 由下式确定 f　　rom

$$\frac{d^2\mathbf{d}^s}{dt^2} - \frac{1}{\rho_0^s}\boldsymbol{\nabla}_0\cdot\left(\mathbf{S}^s\mathbf{F}^T\right) = \mathbf{b}^s \qquad \text{in } \Omega_0^s \ \ \forall t \geq 0\,, \tag{2}$$

其中固体密度为 ρ_0^s，外部体力为 $\mathbf{b}^s$。按照全拉格朗日观点，该运动方程相对于未变形参考构形 Ω_0^s 建立；对于所有受影响的量和算子，均以下标 0 标示。因此，内应力不再用柯西应力张量$\mathbf{T}^s$表示，而是用第二类皮奥拉-基尔霍夫应力张量 $\mathbf{S}^s = \det(\mathbf{F})\mathbf{F}^{-1}\mathbf{T}^s\mathbf{F}^{-T}$ 表示，其中变形梯度为 $\mathbf{F}$。本构方程采用胡克模型或圣维南-基尔霍夫材料模型，由此形成几何非线性固体问题 [16]。指定初始（零）位移场，并在 $\partial\Omega^s$ 上施加适当的边界条件，即可封闭该问题。

2.3. 耦合条件

为确保共享界面 $\Gamma^{fs} = \partial\Omega_f \cap \partial\Omega^s$上的质量、动量和机械能守恒，两个子问题的解场必须满足运动学和动力学连续性：

$$\mathbf{d}^f = \mathbf{d}^s \qquad \text{on } \Gamma^{fs} \ \ \forall t \geq 0\,, \tag{3a}$$

$$\mathbf{T}^f\cdot\mathbf{n}^f = -\mathbf{T}^s\cdot\mathbf{n}^s \qquad \text{on } \Gamma^{fs} \ \ \forall t \geq 0\,, \tag{3b}$$

其中，$\mathbf{d}^f$表示流体的位移，而$\mathbf{n}_f$和$\mathbf{n}^s$是分别从相应区域向外指的界面单位法向量。请注意，式（3a）也意味着速度和加速度相等。

2.4. 狄利克雷-诺伊曼分区

在分区式流固耦合模拟中，两个子问题由两个不同的求解器处理，并以黑箱方式耦合，即仅通过交换界面数据进行耦合。这种策略在求解器方面非常灵活且模块化，但它们之间的通信也带来了一些额外挑战。一方面，两个子问题的界面离散通常并不匹配，因此传递数据需要进行空间投影。另一方面，每个时间步内都需要一个迭代过程，以求得两个相互依赖的子问题的一致解。

最常见的分区方法是将狄利克雷-诺伊曼 分区与高斯-赛德尔型迭代方案[2]结合起来。对于每次耦合迭代，先根据当前界面变形求解流动问题，再将界面牵引力作为诺伊曼边界条件传递给固体。随后，固体求解器计算新的变形状态，并将界面位移返回给流体求解器，由此得到的界面速度构成狄利克雷条件。该过程收敛后，便开始下一个时间步。

狄利克雷-诺伊曼分区的主要缺点是对附加质量效应敏感。对于本研究，只需指出这种失稳是分区求解格式固有的，其程度主要随密度比 ρ^f/ρ^s 增大；更详细的研究可参见文献 [2–4]。

[2]尽管本文所有数值实验均采用狄利克雷-诺伊曼分区，但预计所研究的子问题迭代对计算成本的影响对于其他分区方法（例如罗宾-诺伊曼或罗宾–罗宾格式）基本相同 [7, 17, 18]。

2.5. 界面拟牛顿方法

应对附加质量效应的一种有效方法是在将界面变形传递给流体求解器之前，使用界面拟牛顿 （IQN）方法对其进行修正[3]。将耦合问题的解视为耦合迭代循环的一个不动点，其基本思路是采用类似牛顿法的更新步骤

$$\mathbf{d}^{k+1} = \tilde{\mathbf{d}}^{k} + \Delta\tilde{\mathbf{d}}^{k}_{IQN} = \tilde{\mathbf{d}}^{k} - \left(\frac{\partial \mathbf{r}^{k}}{\partial \tilde{\mathbf{d}}^{k}}\right)^{-1} \mathbf{r}^{k}\,, \tag{4}$$

其中，$\tilde{\mathbf{d}}^{k}$ 是固体求解器计算得到的界面变形，而$\mathbf{d}^{k+1}$ 是发送给流体求解器的界面变形，随后开始下一次耦合迭代$k+1$。$\mathbf{r}^{k} \equiv \tilde{\mathbf{d}}^{k} - \mathbf{d}^{k}$ 表示界面变形的不动点残差。

然而，由于黑箱求解器无法提供精确的雅可比矩阵 $\frac{\partial \tilde{\mathbf{d}}^{k}}{\partial \mathbf{d}^{k}}$，因此改用低秩近似。为避免额外的求解器调用，该逆雅可比近似$\widetilde{\mathbf{J}}^{-1} \approx \left(\frac{\partial \mathbf{r}^{k}}{\partial \tilde{\mathbf{d}}^{k}}\right)^{-1}$ 由此前耦合迭代中计算得到的界面变形状态构造。

有关 界面拟牛顿方法的概念、不同变体和实现方面的详细信息，作者建作者建议。以及文献 [15, 18–22]。

3. 非线性子问题的解

分区式流固耦合将两个黑箱求解器耦合起来：一个用于流动问题，另一个用于固体问题。为拓展本文的研究范围，本文考察了两组不同的子问题求解技术；这两组技术都广泛应用于流固耦合。第一组对流动问题和固体问题均采用有限元法，而第二种模拟框架对固体问题采用有限元法，对流动问题则采用有限体积法。下面对这两种方案进行概述：

- 在第一种模拟框架中，自研求解器 XNS 采用稳定化的拉格朗日 P1-P1 有限元在空间离散流动问题 [23, 24] ，并在时间上采用后向欧拉法 [25]。ALE 网格通过线弹性网格更新法适应变形区域 [26, 27], ，其速度则由一阶有限差分格式确定 [28]。固体子问题由自研代码 FEAFA 通过等几何分析求解 [29, 30], ：空间上采用基于样条的有限元变体，时间上采用广义-α格式 [31, 32]。非匹配界面离散通过经样条增强的有限插值单元处理 [33, 34]。下文将此方案称为**有限元-有限元**。

- 第二种框架使用有限体积法求解流动问题，并在商用求解器 ANSYS Fluent [35]中采用后向欧拉时间离散化。网格为结构化网格，动量方程中对流项的离散格式为二阶迎风格式 [36]。对于压力方程，顶盖驱动方腔和柔性管案例分别采用二阶格式和标准格式 [37] ，详见第 5 节。变形流体域通过任意拉格朗日-欧拉（ALE）参考系处理，网格变形则采用基于弹簧的平滑方法 [38]。固体问题在空间上采用分片线性有限元离散，在时间上采用广义-α 格式，并使用 Kratos 多物理场代码中的结构力学应用程序 [39]。两者之间的耦合由内部代码 CoCoNuT [20]实现。最新代码可在 GitHub 仓库 pyfsi/coconut 中找到。非匹配界面上的数据交换通过径向基映射 [40]实现。以下将这一组求解技术称为**有限体积-有限元**。

有限元法和有限体积法都是现代计算工程科学中常见的离散化格式。由于相关文献已有详尽介绍，例如有限体积法见 [41–43] ，有限元法见 [16, 44–46]，为简洁起见，此处不再深入讨论。

[3] 虽然这种做法不太常见，但也可以先更新界面牵引力，再将其传递给固体求解器 [18]。

对于本研究而言，最重要的一点是，这两种方法都会将连续的偏微分方程（PDE）及其边界条件转化为一组离散的代数方程。通常，这个方程组是非线性的，也就是说，系统矩阵 $\mathbf{A}$ 取决于解 $\mathbf{u}$，从而得到矩阵形式

$$\mathbf{A}(\mathbf{u})\,\mathbf{u} = \mathbf{b}, \tag{5}$$

其中，$\mathbf{A} \in \mathbb{R}^{n\times n}$ 是稀疏系数矩阵，$\mathbf{u} \in \mathbb{R}^{n}$ 包含所关注的因变量，而n 表示自由度（DOF）的数量。请注意，在每个子问题[4]中，右端项（RHS）向量$\mathbf{b}$包含源项和边界条件，并被视为独立于$\mathbf{u}$ 。然而，第 4.3 节将说明，在流固耦合（FSI）等情况下耦合两个子问题时，右端项实际上也成为$\mathbf{u}$ 的函数，即$\mathbf{b} = \mathbf{b}(\mathbf{u})$。对于非定常偏微分方程（PDE），例如式（1）或式（2），还需应用时间步进格式，从而在每个时间步得到式（5）所示的代数方程组。随后可采用适当的数值方法求解。

以下小节重点讨论如何处理离散方程组（式（5））中的非线性。虽然有限元法和有限体积方法都可以通过牛顿迭代或不动点迭代求解该方程组，但本文采用最常见的方法：有限元法使用牛顿迭代，有限体积法使用不动点迭代。最重要的是，有限元法通过少量牛顿迭代（通常为 2 至 5 次）求解式（5）；而有限体积法则通过成本低得多的不动点迭代求得解，但所需迭代次数较多（通常为 50 至 200 次）。

3.1. 有限元法

尽管有限元方法最初是为固体力学而发展起来的，但如今已广泛应用于计算工程的许多领域，包括流体动力学 [24, 47]。将偏微分方程转换为其变分形式后，计算域被划分为由元素 构成的网格，并通过元素基函数的线性组合来近似解。随后，根据得到的代数方程组（形式如式 (5)）确定未知系数。

在有限元方法中，通常使用牛顿法处理$\mathbf{A}$的非线性。从初始猜测$\mathbf{u}^0$,开始，在每次迭代$i = 1, \cdots$中求解线性化方程组

$$\underbrace{\left(\mathbf{A}^{i-1} + \left.\frac{\partial \mathbf{A}}{\partial \mathbf{u}}\right|_{i-1} \mathbf{u}^{i-1}\right)}_{=:\mathbf{K}^{i-1}} \Delta\mathbf{u}^{i} = \underbrace{\mathbf{b} - \mathbf{A}^{i-1}\,\mathbf{u}^{i-1}}_{=:\mathbf{r}_p^{i-1}} \quad \Longleftrightarrow \quad \mathbf{K}^{i-1}\,\Delta\mathbf{u}^{i} = \mathbf{r}_p^{i-1} \tag{6}$$

以求得解增量 $\Delta\mathbf{u}^i$，其中$\mathbf{K}^{i-1}$ 是切线刚度矩阵（或系统矩阵），而$\mathbf{A}^{i-1}$ 是$\mathbf{A}(\mathbf{u}^{i-1})$的简写。此外，$\mathbf{r}_p^{i-1}$是所考虑子问题的残差向量，即流体（$p = f$）或固体（$p = s$）问题的残差向量。随后，通过$\mathbf{u}^i = \mathbf{u}^{i-1} + \Delta\mathbf{u}^i$更新解场。当残差范数低于某个容差 ε 时，认为牛顿迭代已收敛，即若 $\frac{\|\mathbf{r}_p^i\|_2}{\sqrt{n}} < \varepsilon$。

需要注意的是，该过程的计算成本通常主要来自两方面：一是组装式（6）中的线性系统，二是对其进行数值求解（本文采用预条件广义最小残差法 [48]）。这两项操作在每次牛顿迭代中都会重复进行。

3.2. 有限体积法

有限体积方法常用于求解流动问题。有限体积法的原理是将空间域离散为有限体积或单元，并在每个单元上应用控制方程的积分形式。在这些积分形式中，利用高斯定理将散度的体积分转换为其边界上的曲面积分，因此该方法在设计上具有守恒性。最终，这会得到一个大型代数方程组，其形式如式（5）所示。在基于压力的有限体积求解器中，压力和速度可以通过所谓的耦合方法（本文采用的方法）一起求解，也可以通过分离方法依次求解。在后一种情况下，式（5）仅是一种符号表示。

[4]在许多算法中，对$\mathbf{u}$ 的部分依赖关系采用显式处理，因此会对右端项$\mathbf{b}$产生滞后贡献。由于这并非本文关注的重点，且为简化起见，假设对$\mathbf{u}$ 的所有依赖关系均采用隐式处理，即在系统矩阵$\mathbf{A}(\mathbf{u})$中处理。

方程组也可以通过牛顿迭代求解，但更常见的做法是使用 Picard 迭代或不动点迭代对非线性系数矩阵**A** 进行线性化

$$\mathbf{M}^{i-1}\,(\mathbf{u}^{i}-\mathbf{u}^{i-1}) = \mathbf{b} - \mathbf{A}^{i-1}\,\mathbf{u}^{i-1}, \tag{7}$$

其中$\mathbf{M}^{i-1} = \mathbf{M}(\mathbf{u}^{i-1})$是$\mathbf{A}^{i-1}$的某种近似，例如 $\mathrm{diag}(\mathbf{A}^{i-1})$。需要指出的是，这种记法仅为符号表示，因为实际的求解技术更为复杂，因此不在本文的讨论范围内；相关内容可参见多重网格方法，例如采用平滑器 [49] 的代数多重网格法（ AMG ）。当残差范数低于某个容差 ε 时，即认为迭代已收敛。在本文中，残差按总不平衡量计算，并乘以一个系数进行缩放，该系数代表**u** 相应分量流经计算域的流量。

4. 分区算法的计算成本与收敛

尽管上一节阐述的技术细节有所不同，有限元求解器和有限体积求解器都通过内部迭代循环处理非线性问题；该循环在每次迭代中重复组装并（部分）求解线性化方程组。为简化术语，以下将使用“子问题迭代”一词，指牛顿迭代或不动点迭代。

本节讨论这些子问题迭代对流固耦合分区方案计算成本的影响。此外，本节提出了一种新准则，仅依据子问题残差来评估耦合循环的收敛性。

注 *1*：为使符号清晰，本文使用大写字母N 表示整个模拟中的所有迭代次数，例如耦合迭代总次数N^c。相较之下，小写字母n 表示流体或固体求解器每次调用中的子问题迭代次数，n^f 和n^s。每次求解器调用时，子问题迭代索引i 都会从 1 重新开始；每个时间步的耦合迭代索引k 亦然。此外，为简化推导，第 4.1 节使用索引$\bar{k}$ 遍历整个模拟中的所有N^c 个耦合步。

4.1. 成本度量

在分区式流固耦合模拟中，通常将两个求解器视为黑箱，只了解其输入和输出，而不了解其内部特性和功能。此时，特定方案的计算成本以达到收敛所需的耦合迭代次数来量化。尽管这一度量基于这样的假设：求解子问题的成本远高于数据交换或耦合过程中的其他步骤，但它实际上也假定单次求解器调用的成本是恒定的。

这可能与经验观察不符，但若尽可能严格地遵循黑箱视角，即完全不了解求解器的内部工作原理，那么这确实是最佳推测。然而在实践中，几乎总是能够判断子问题是否为非线性问题，以及求解器是否采用有限体积或有限元等标准离散化技术。

如第 3 节所述，有限体积和有限元求解器都会在每次子问题迭代中重复执行开销最大的步骤，即组装并（部分）求解线性化系统。因此，求解器调用成本中的相当一部分预计会随着其执行的子问题迭代次数而变化。因此，本文提出如下方式$\bar{}$估算在耦合迭代$p\ k$ 中调用子问题的计算成本：

$$C^{p}(\bar{k}) \approx C^{p}_{\mathrm{fix}} + n^{p}(\bar{k}) \cdot C^{p}_{\mathrm{iter}}\,, \tag{8}$$

其中，$n^{p}(\bar{k})$ 表示此次求解器调用所执行的子问题迭代次数。固定成本C^{p}_{fix} 涵盖每次求解器调用仅执行一次的所有操作，例如更新网格或计算流体载荷。相比之下，C^{p}_{iter} 表示每次子问题迭代产生的所有成本，尤其包括组装和求解线性系统。请注意，成本系数C^{p}_{fix} 和C^{p}_{iter} 假定与迭代次数或时间步无关。第 5 节的结果将验证这一假设是否成立。

对于模拟的总运行时间$t_{\text{simulation}}$，成本度量$C_{\text{simulation}}$ 通过将求解器调用和数据传输的成本在耦合迭代总次数N^c范围内求和得到，即

$$t_{\text{simulation}} \approx C_{\text{simulation}} = \sum_{\bar{k}=1}^{N^c} \left[C^c + \sum_{p=f,s} C^p(\bar{k}) \right] = N^c \cdot C^c + \sum_{\bar{k}=1}^{N^c} \sum_{p=f,s} C^p(\bar{k}) \,, \tag{9}$$

其中，每次耦合迭代中数据传输和更新技术（如松弛或 IQN）的成本C^c 被认为是常数。请注意，使用遍历所有耦合迭代的索引k ，是为了避免引入额外的时间步求和。

本文采用的成本度量，是将式 (8) 代入式 (9)，并根据各项是否随耦合迭代次数或子问题迭代次数变化，对各项重新分组后得到的：

$$\begin{aligned} C_{\text{simulation}} &= N^c \cdot C^c \;+ \sum_{\bar{k}=1}^{N^c} \sum_{p=f,s} [C^p_{\text{fix}} \;+\; n^p(\bar{k}) \cdot C^p_{\text{iter}}] \\ &= N^c \cdot \underbrace{(C^c \;+ \sum_{p=f,s} C^p_{\text{fix}})}_{=:\,\overline{C}^c} \quad + \sum_{p=f,s} C^p_{\text{iter}} \underbrace{\sum_{\bar{k}=1}^{N^c} n^p(\bar{k})}_{=:\,N^p} \\ &= N^c \cdot \overline{C}^c \;+ \sum_{p=f,s} N^p \cdot C^p_{\text{iter}} \end{aligned} \tag{10}$$

新引入的变量N_f、N^s以及$C^c = C^c + C^f{}_{\text{fix}} + C^s{}_{\text{fix}}$ 分别表示问题f或s的子问题迭代总次数，以及每次耦合迭代中发生的所有成本。

成本因子$C^p{}_{\text{iter}}$ 和C^c 由模拟框架、计算机体系结构、问题规模等因素决定，因此视为已知常数；由此，成本度量可归结为迭代次数N^c、N_f和N^s的加权和。因此，高效的分区方案不仅要减少耦合迭代总次数，尤其还要减少子问题迭代总次数。

注 *2*： 在分区式流固耦合中，流体问题的网格更新有时会被视为流体和结构之外的第三个问题。虽然只需将网格更新加入子问题求和项，即可轻松将这种观点纳入式 (9)，但为使符号更简洁，本文将其计入流动求解器成本。只要网格更新次数随 N^c （或 N_f）成比例变化，这种处理就是等效的。

4.2. 每次求解器调用的迭代次数

在整个模拟过程中，影响子问题迭代总次数 N^f 和 N^s 的一种直接而简单的方法，是限制每次求解器调用所执行的子问题迭代次数。更具体地说，每次求解器调用最多只能执行 n^f_{max} 或 n^s_{max} 次子问题迭代，而不是始终迭代直至完全收敛。

这自然引出了本文探讨的核心研究问题：每次求解器调用的子问题迭代次数上限 n^f_{max} 和 n^s_{max}，会如何影响分区算法的总计算成本？乍一看，这个问题似乎很简单，但每次求解器调用的子问题迭代次数与计算成本之间的关系并非显而易见：

- 如果每次求解器调用执行的子问题迭代次数较少，求解器就会更频繁地交换数据，从而使流固耦合界面上的边界条件以及右端项**b** 保持最新，提升每次子问题迭代的质量。然而，除了通信成本较高之外，这也会带来将不准确的数据反馈到耦合循环以及（如适用）拟牛顿雅可比近似中的风险，从而导致收敛变慢，即N_c增大，甚至发散。

- 另一方面，每次求解器调用中执行更多子问题迭代，可以提高交换的数据场和 IQN 方法输入输出对的准确性，通常会减少耦合步数

N^c。缺点是，计算时间被浪费在优化一个边界条件仍不正确的解上，而该解无论如何都会在下一耦合步中被覆盖。

每次流动和固体求解器调用所采用的子问题迭代次数 n^f_{max} 和 n^s_{max}应合理选择，以平衡这些相反趋势，从而降低总体计算成本。

4.3. 收敛准则

在分区式流固耦合模拟中，通常通过将不动点残差$\mathbf{r}^k$ 的某种范数与容差 ε^c进行比较，来判断耦合循环是否收敛，参见例如 [11, 50–52]。为了获得准确结果，两个子问题的迭代循环也必须收敛；然而，这些条件很少被明确考虑，因为只要所有求解器调用都迭代至完全收敛，这些条件便会自然满足。但是，若将每次求解器调用的子问题迭代次数限制为n^f_{max} 或n^s_{max}，则需要一种方法来确保两个子问题在进入下一个时间步之前都满足其残差容差 ε^f 和 ε^s 。遗憾的是，文献中很少讨论不动点容差 ε^c 的影响，更不用说它与子问题残差容差 ε^f 和 ε^s之间的相互作用了。

为解决这一问题，本文为分区算法提出了一种新颖的收敛准则，无需引入任何耦合容差 ε^c。该准则仅依赖于子问题残差。此外，新准则还使收敛速率得以公平比较，而不受每次求解器调用中子问题迭代次数的影响。本节余下部分将推导所提出的收敛准则。

如第 3 节所述，有限元和有限体积求解器都会迭代求解以下形式的非线性问题：$\mathbf{A(u)\,u} = \mathbf{b}$。换言之，尽管它们考虑了系统矩阵**A**的非线性，但假设右端向量**b** 在整个子问题迭代过程中保持不变。然而，在耦合流固问题中，两个子问题的右端项实际上也取决于**u**，即使这种依赖仅以隐式方式存在。例如，流体作用于固体上的牵引力和压力会随变形状态而改变。类似地，更新流场会改变固体的变形，进而改变流体边界的位置和速度。在实践中，采用分区算法耦合两个黑箱求解器时，无法显式考虑右端项的这种隐式非线性。

一个有趣的推论是，求解器调用中只有第一次子问题迭代使用了正确的右端向量 **b**，因为在第一次迭代中，系统矩阵**A** 和右端项 **b** 都对应当前解$\mathbf{u}^0$, , 即子问题求解的初始值。另一方面，后续子问题迭代不可避免地缺少其他问题对**b(u)**的贡献；随着当前求解器调用过程中**u** 发生变化，这部分贡献本应随之产生。但此时只有系统矩阵更新为新的解$\mathbf{u}^i$，而$\mathbf{b} = \mathbf{b}(\mathbf{u}^0)$保持不变[5]。其影响示意于表1中。

迭代	不动点迭代	牛顿迭代
$i = 1$	$\mathbf{M}(\mathbf{u}^0)\,(\mathbf{u} - \mathbf{u}^0) = \mathbf{b}(\mathbf{u}^0) - \mathbf{A}(\mathbf{u}^0)\,\mathbf{u}^0$	$\mathbf{K}(\mathbf{u}^0)\,\Delta\mathbf{u} = \mathbf{b}(\mathbf{u}^0) - \mathbf{A}(\mathbf{u}^0)\,\mathbf{u}^0$
$i > 1$	$\mathbf{M}(\mathbf{u}^{i-1})\,(\mathbf{u} - \mathbf{u}^{i-1}) = \mathbf{b}(\mathbf{u}^0) - \mathbf{A}(\mathbf{u}^{i-1})\mathbf{u}^{i-1}$	$\mathbf{K}(\mathbf{u}^{i-1})\,\Delta\mathbf{u} = \mathbf{b}(\mathbf{u}^0) - \mathbf{A}(\mathbf{u}^{i-1})\mathbf{u}^{i-1}$

表1：由于右端向量 **b** 对解 **u** 的依赖关系源自另一个子问题，因此在分区式流固耦合模拟的子问题迭代过程中缺失，所以每次求解器调用中只有第一次子问题迭代使用了正确的右端项。对于 $i > 1$，右端向量有误，即 $\mathbf{b}(\mathbf{u}^0) \neq \mathbf{b}(\mathbf{u}^{i-1})$。

因此，只有求解器调用中首次子问题迭代的残差，$\mathbf{r}^0_p = \mathbf{b}(\mathbf{u}^0) - \mathbf{A}(\mathbf{u}^0)\mathbf{u}^0$, 才能用于判断耦合循环是否收敛，因为它量化了当前子问题在多大程度上已与当前耦合数据达到平衡；也就是说，这些耦合数据是由当前子问题解产生的。例如，固体求解器的首次残差是柯西应力与流体求解器以当前变形状态作为边界条件所确定的外载荷之间的差值。子问题

[5]如脚注 4（第 5 页）所述，假定子问题固有的所有关于 **u** 的依赖关系都采用隐式处理，即在系统矩阵 **A(u)** 中处理。

迭代$i = 2, 3, \cdots$则只是试图针对有缺陷的右端向量$\mathbf{b} = \mathbf{b}(\mathbf{u}^0) \neq \mathbf{b}(\mathbf{u}^{i-1})$求得收敛解。

这启发我们通过第一个子问题迭代的子问题残差来触发耦合问题的收敛：如果所有子问题的第一个子问题迭代残差都满足各自的容差 ε^f 或 ε^s，则认为耦合迭代已收敛。不同的迭代循环和新的收敛准则如图1所示。

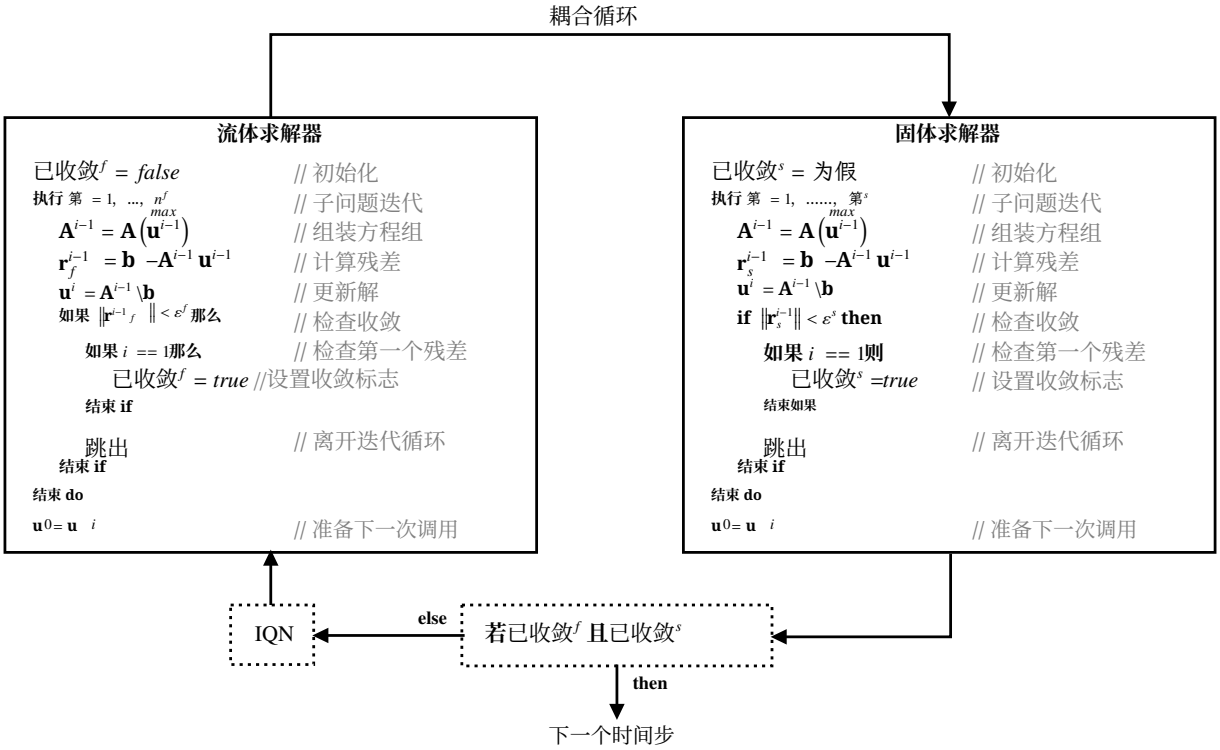


图1：本文所考虑的不同迭代循环示意图。虽然只有一个耦合循环负责管理求解器调用，但每个子问题都有自己的内部迭代。请注意，伪代码 $\mathbf{u}^i = \mathbf{A}^{i-1} \backslash \mathbf{b}$ 表示求解牛顿迭代对应的式（6）或不动点迭代对应的式（7）。同样，$\left\|\mathbf{r}_f^{i-1}\right\| < \varepsilon^f$ 仅仅是对第 3 节所讨论的收敛检查的符号表示。最后，所示的拟牛顿更新并非必需。

要将新的判据集成到现有的黑箱流固耦合框架中，通常最简单的做法是在每次求解器调用后检查：(1) 子问题是否已收敛；(2) 是否仅执行了一次子问题迭代。只有当当前耦合迭代中，流动求解器和固体求解器都满足这两个条件时，该时间步才算收敛。

注释 *3*：请注意，在图1中，子问题的解是在计算残差之后、检查收敛之前更新的。因此，即使子问题迭代已经收敛，仍会再进行一次更新。也可以在收敛时退出子问题迭代。然而，对于黑箱求解器，这通常是不可能的；此外，耦合解的精度将由最先满足的子问题收敛容差决定，而不是由最后满足的容差决定。

注释 *4*：将一次求解器调用中的第一个子问题残差解释为耦合循环收敛程度的度量，只有在求解器之间交换的数据场未被修改时才完全准确。尽管这种修改在实践中很常见，例如松弛或 IQN 方法（第 2.5 节）便包含此类修改，但在这些方法中，更新增量与不动点残差 $\mathbf{r}^k$ 成正比，因此会在收敛时趋于零，这一点将在第 5 节得到验证。因此，在所提出的收敛准则中忽略更新步骤的影响是稳妥的。

5. 结果

本节讨论将所开发的概念应用于数值测试案例。为考察其对计算成本的影响，我们针对每个测试案例进行多次运行，并改变每次求解器调用中子问题迭代的最大次数。

5.1. 算例设置

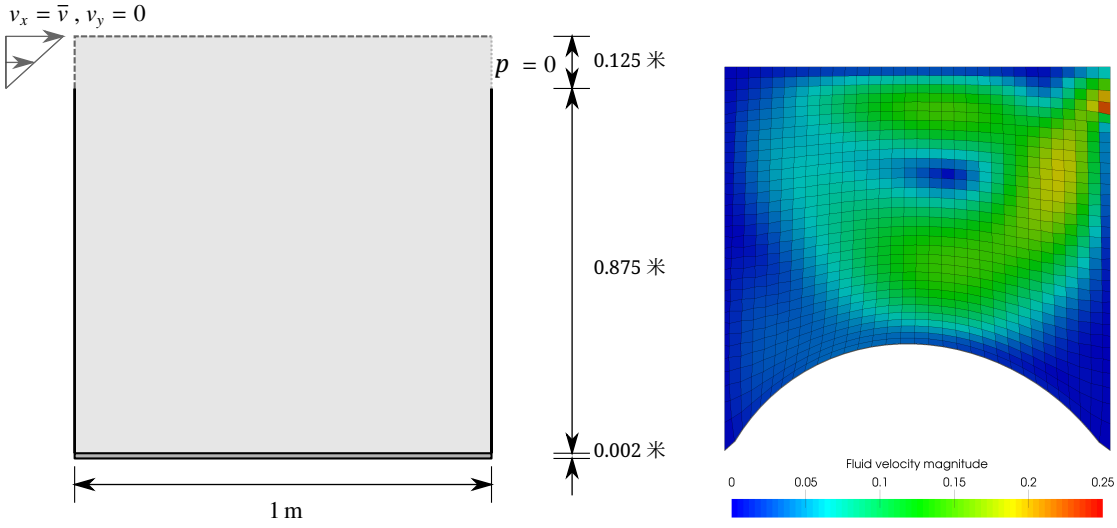
研究了两个不同的算例，它们都是流固耦合领域中广为人知的基准算例。分别采用有限元-有限元和有限体积-有限元框架对其进行模拟。

5.1.1. 顶盖驱动方腔算例

第一个算例研究一个二维（2D）方腔，其底部为柔性底板，受振荡来流和顶盖运动 [53, 54]激励。方腔和柔性底板的几何形状详见图 2a，表 2 列出了顶盖的结构密度 ρ^s、弹性模量 E^s和泊松比 ν^s，以及液体的密度 ρ^f和动力黏度 μ^f。柔性底板的两端均固定。此外，在方腔的顶部边界施加水平速度 $\bar{v}(t)$，该速度（单位为 m/s）表示为时间t 的函数，形式如下：

$$\bar{v} = 1 - \cos\left(\frac{2\pi t}{5}\right). \tag{11}$$

在左侧边界的上部施加相同的速度，但该速度从零开始沿竖直方向线性增加，至顶部达到 $\bar{v}(t)$，如图 2a 所示。在右侧边界的对应区域施加 0 Pa 的压力。模拟运行 700 个时间步，每步 0.1 s，总时间为 70 s。图 2b 展示了柔性底板的变形。



(a) 几何形状和边界条件示意图。

(b) 柔性底板的变形及相应的速度场（单位：米/秒），在 t = 40秒，在有限体积-有限元框架中进行模拟。

图 2：顶盖驱动方腔算例的可视化结果。

图 3 对比了采用两种框架得到的结果，以及这些结果与文献数据的对比，表明振荡周期吻合良好。尽管振幅存在一定差异，但两种框架之间的差异与文献中观察到的差异一致。

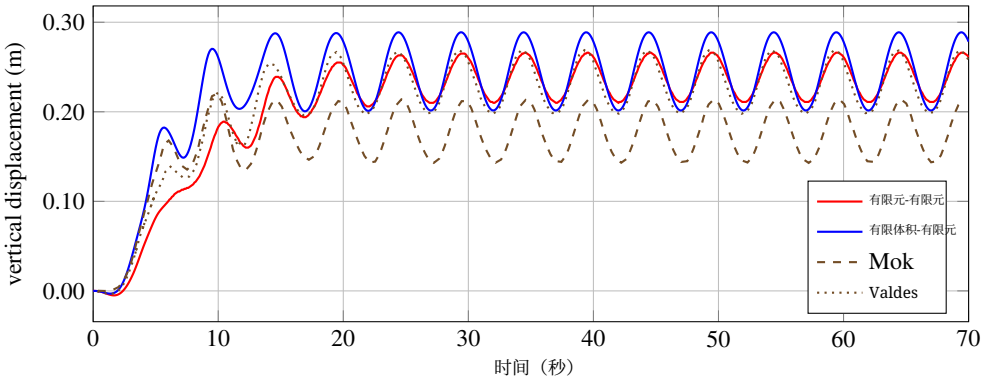


图 3：柔性底壁顶部中心点的垂直位移随时间的变化。将两种框架的结果与 Mok [53] 和 Valdes [54]所得结果进行比较。

	流动参数	数值	固体参数	数值
顶盖驱动方腔	ρ^f	1 千克/米^3	ρ_s	500 千克/米^3
	μ^f	0.01 Pa 秒	E^s	250 N/m^2
			ν^s	0.0
柔性管	ρ^f	1000 kg/m^3	ρ^s	1200 kg/m^3
	μ^f	0.003 Pa·秒	E^s	300 000 N/m^2
			ν^s	0.3

表 2：顶盖驱动方腔和柔性管案例的参数值。

5.1.2. 柔性管案例

第二个案例是压力脉冲在柔性管中传播的模拟 [22]。与顶盖驱动方腔算例不同，该模拟在三维（3D）中进行。管长为 0.05 m，半径为 0.005m，壁厚为 0.001m。材料参数见表 2。

管壁两端均被固定。在入口处，前 0.003 秒施加 1333.2 Pa 的压力，之后压力为 0 Pa。出口处的压力固定为 0 Pa。模拟总时间为 0.01 秒，分为 100 个时间步，每个时间步为 0.0001 秒。图 4 展示了压力脉冲在管内传播的过程。

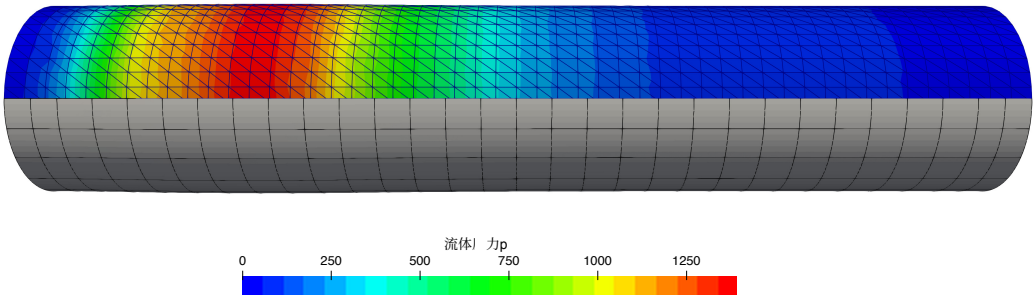


图 4：在有限元-有限元框架中模拟的 $t = 0.004$ 秒时的柔性管。上半部分显示以 Pa 为单位的流体压力场，下半部分展示等几何分析壳体结构。

本研究旨在考察两种软件框架中类似的影响，而非对二者进行定量比较，因此并不严格要求结果完全一致。尽管如此，图5比较了两种框架所得的数据。虽然存在细微的定量差异，但两张图非常相似。

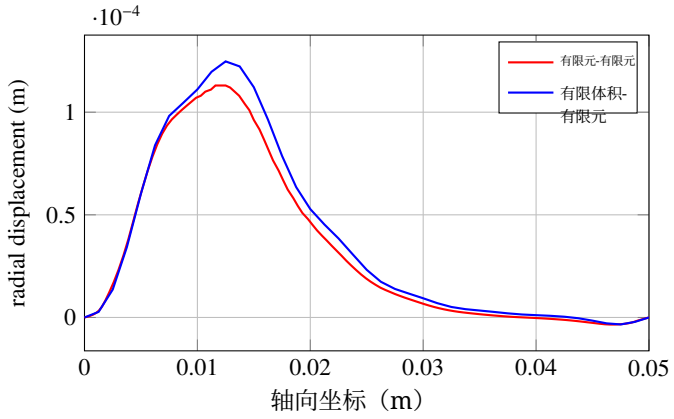


图5：在t = 0.004 s时，两个求解器框架的径向位移随轴向坐标的变化。

	q	ε^{fil}	ε^{f}	ε^{s}
顶盖驱动方腔				
有限元-有限元	3	5E−9	1E−9	1E−9
有限体积-有限元	3	1E−9	1E−6	1E−9
柔性管				
有限元-有限元	5	1E−12	1E−9	1E−9
有限体积-有限元	5	1E−12	1E−7	1E−9

表 3：耦合和求解器设置摘要。

5.1.3. 耦合和求解器设置

上述流固耦合问题采用拟牛顿算法 IQN-ILS [22]求解。各问题的复用参数 q 详见表 3。在构造近似雅可比矩阵之前，通过滤波 [50]去除存储的残差数据中的线性相关性。表 3 还列出了所用的滤波容差 ε_{fil}，以及流动和固体求解器的子问题容差，分别为 ε^{f} 和 ε^{s}；相关讨论见第 3 节。表 4 显示，两种框架均力求采用相似的边长。

5.2. 成本因子和回归

如第 4 节所述，分区模拟的成本并非仅由耦合迭代次数决定，还取决于子问题迭代次数。因此，公式（10）将成本度量定义为迭代次数N^{c}、N_{f}和N^{s}的加权和。权重由一组成本系数组成，用于计入数据传输成本，以及每次求解器调用的固定成本和随迭代次数变化的成本。虽然通常无法确定这五个成本因子的具体数值，但可以针对特定案例和框架，通过线性回归对其进行近似估算。

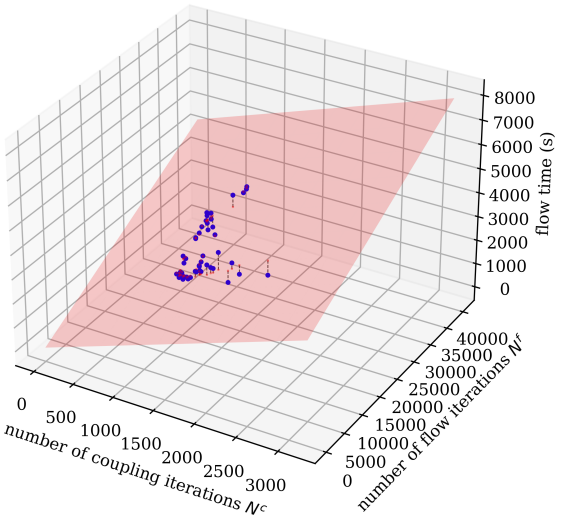
	流体求解器			固体求解器		
	单元/元素	节点	DOFs	元素	控制点/节点	DOFs
顶盖驱动方腔						
有限元-有限元	2 048	1 089	3 267	31	99	198
有限体积-有限元	1 024	1 089	3 072	64	99	198
柔性管						
有限元-有限元	35 986	7 195	28 780	600	800	4 800
有限体积-有限元	7 200	7 913	28 800	780	800	4 800

表 4：离散化汇总。

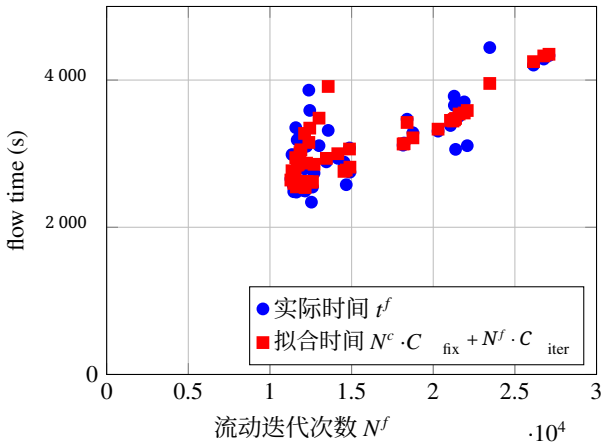
为此，将每次模拟的总运行时间分为三部分：流体求解器所用时间，即流动时间t^f；固体求解器所用时间，即固体时间t^s；以及剩余时间，称为耦合时间t^c。对于每项参数研究，即特定案例与框架的组合，回归将所有已完成运行的耗时和迭代次数作为数据集。

下面以 FV-FE 柔性管案例的计算为例进行回归分析。结果本身将在第5.3节中进一步详细描述。

针对流动时间和固体时间，采用多元线性回归，并使用两个自变量：耦合迭代次数 N^c 以及子问题迭代次数 N^f 或 N^s。忽略每次模拟中仅发生一次的成本，因此采用零截距，使回归结果实际上是一个过原点的平面。例如，图6a展示了使用 FV-FE 框架求解管道算例时流动时间对应的平面，而图6b则将相同数据投影到以流动求解器迭代次数为横轴的图中。



(a) 通过多元回归得到的平面。



(b) 在流动时间和N_f 平面上的投影。

图 6：采用有限体积-有限元框架，对管道算例中的t^f关于N^c 和N_f进行多元线性回归。

通过相对均方根误差（RRMSE）评估拟合精度，流动时间的相对均方根误差计算如下：

$$\text{RRMSE} = \sqrt{\frac{\sum^m \left| t^f - (N^c \cdot C^f_{\text{fix}} + N^f \cdot C^f_{\text{iter}}) \right|^2}{\sum^m \left| t^f \right|^2}}, \tag{12}$$

其中，m 表示所考虑的计算次数。对于采用有限体积-有限元的柔性管，流动时间回归的相对均方根误差为 7.43 %。这一数值表明回归结果具有较高的准确性。剩余差异是由于处理器的负载和时钟频率变化导致运行时间自然波动，以及模型未考虑的一些次要影响所致，例如求解组装后的矩阵向量系统的成本变化。此外，求解器在后台执行的优化操作（如负载均衡或优化某些求解器参数，例如松弛因子）也会导致运行时间发生波动。对于有限体积-有限元框架中的商用有限体积求解器，这种情况尤为明显[6]。作为补充，固体时间回归的相对均方根误差为 3.44 %。

对于耦合时间，仅以耦合步总数N^c作为自变量进行线性回归分析。同样，假设截距为零，因此得到的直线经过原点。图 7 展示了使用有限体积-有限元框架求解管道算例所得的直线。相对均方根误差为 2.65 %，同样较低，表明耦合迭代次数与耦合成本之间假设的正比关系是准确的。

[6]因此，仅使用科学计算代码的有限元-有限元框架在两个测试案例中的相对均方根误差都要低得多，见表 A.7。

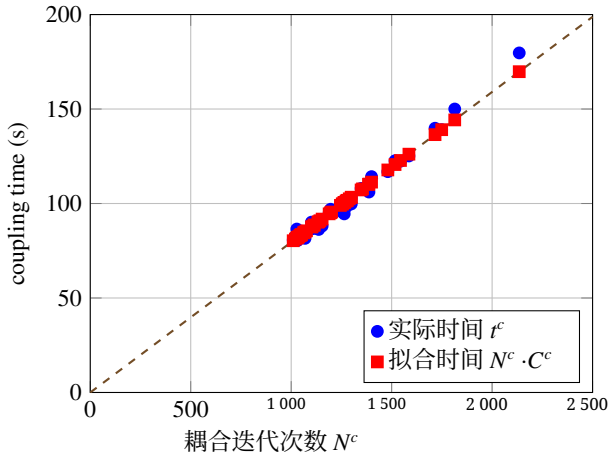


图 7：采用有限体积-有限元框架的管道算例中t^c 关于 N^c 的线性回归。

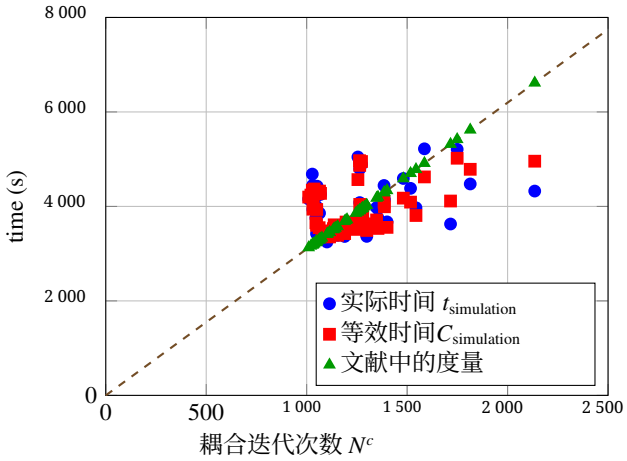


图 8：在有限体积-有限元框架下的柔性管案例中，实际时间、本文采用的等效时间，以及假设每次耦合迭代成本恒定时的成本度量之间的差异。请注意，假设每次耦合迭代成本恒定时，等效时间呈一条经过原点的直线。

如第 4 节所述，所有随耦合迭代次数缩放的成本系数都可以合并为C^c。将这些系数代入公式（10），即可得到运行时间的成本度量，称为等效时间。对于采用有限体积-有限元方法的柔性管案例，实际运行时间与等效时间之间相对差异的绝对值平均为 3.80 %，且最高不超过 14.62 %。前一个指标称为平均绝对百分比误差（MAPE），类似地，后一个指标称为最大绝对百分比误差（maxAPE）。其定义如下：

$$\text{MAPE} = \frac{1}{m}\sum^{m}\left|\frac{t_{\text{simulation}} - C_{\text{simulation}}}{t_{\text{simulation}}}\right| \quad \text{and} \quad \text{maxAPE} = \max_{m}\left(\left|\frac{t_{\text{simulation}} - C_{\text{simulation}}}{t_{\text{simulation}}}\right|\right). \tag{13}$$

作为对比，若假设总成本仅随耦合迭代次数变化（这是文献中的常见做法），即设置$C^{f,s}_{\text{迭代}} = 0$，则偏差增至平均值 13.58 %、最大值 52.96 %。图 8 展示了这些度量与模拟实际成本之间的差异及其误差。因此，本文提出的度量并不完美，但确实优于文献中占主导地位的度量。

这种等效时间的另一项优点是，它不受运行时间波动的影响，因为

	C^f_{fix}	C^f_{iter}	C^s_{fix}	C^s_{iter}	C^c	$\overline{C}^c$	MAPE	最大绝对百分比误差
顶盖驱动方腔								
有限元-有限元	0.0208	0.0298	0.0002	0.0023	0.0004	0.0214	0.33 %	0.89 %
有限体积-有限元	0.8985	0.0727	0.0243	0.0041	0.0218	0.9446	3.86 %	15.76 %
柔性管								
有限元-有限元	0.6459	1.4756	0.0128	0.2076	0.1873	0.8460	0.22 %	0.58 %
有限体积-有限元	1.1542	0.1068	0.1587	0.2510	0.0795	1.3924	3.80 %	14.62 %

表5：每次相应迭代的成本因子（以秒为单位），以及实际运行时间与等效运行时间之间的差异；差异以平均最大绝对百分比误差和最大绝对百分比误差表示。

实际运行时间中的随机波动会被滤除。因此，从此处开始将使用等效时间作为成本度量，而非实际时间。表5总结了顶盖驱动方腔和柔性管案例在两种框架下的成本因子，以及实际运行时间与等效运行时间之间的差异 [7]

本节阐述的方法通过采用有限体积-有限元框架的柔性管模拟进行了说明。附录 A 给出了其他问题与框架组合的回归质量度量。

5.3. 参数研究结果

本研究的主要问题是：限制每次求解器调用中的子问题迭代次数，会如何影响分区式流固耦合算法的整体性能。为此，我们针对n^f_{max}和n^s_{max}的不同取值，多次运行了模拟。得益于新的收敛准则，每次运行都将耦合问题求解至相同的容差。两个流固耦合软件框架分别研究了两个测试案例，共进行了四项参数研究。以下小节将介绍研究结果，重点探讨n^f_{max}和n^s_{max}的变化如何影响不同的量和指标。

5.3.1. 迭代次数

一个核心方面是每次求解器调用中的子问题迭代次数对三个全局迭代次数的影响，即耦合迭代总数N^c，以及流动求解器和固体求解器的子问题迭代总数N_f和N^s。因此，图9和图10分别通过三张等高线图展示它们与n^f_{max}和n^s_{max}的关系。为了使示例更具代表性，图9基于使用有限体积-有限元框架模拟的顶盖驱动方腔算例，而图10则展示了有限元-有限元框架下的管道算例。

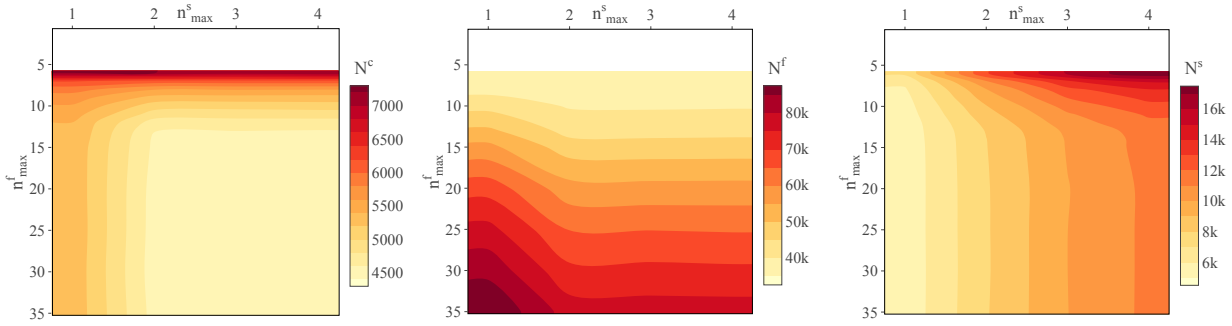
这两个示例呈现的总体趋势高度一致。最大的差异在于，有限体积流动求解器所得的曲线更为“细密”，因为它采用定点迭代而非牛顿步，因此需要更多次迭代，不过每次迭代的成本通常较低。三个子图的主要发现如下：

(a) 每次求解器调用中执行更多子问题迭代时，耦合迭代次数明显减少，因此每次调用都迭代至完全收敛时，N^c 最小。根据第 4.2 节的分析，这一现象符合预期，因为在给定输入数据的情况下，可确保子问题反馈至耦合循环的界面数据尽可能准确。反过来，减小 n^f_{max} 和/或 n^s_{max} 会降低交换的数据场的质量，导致耦合迭代次数增加。这种影响甚至可能严重到导致耦合发散；如图9所示，当 $n^f_{max} < 6$ 时，有限体积-有限元框架就出现了这种情况。
(b) 对流动迭代 影响最大的因素是总数N_f随n^f_{max}增大。这表明，随着n_{fmax}增大，每个耦合步内执行额外子问题迭代所导致的N_f增量，超过了(a)中讨论的因耦合迭代次数减少而节省的迭代次数。固定n^f_{max}后可看出一个次要趋势：每次求解器调用执行更多固体迭代，会减少流动迭代总数。这同样可以用固体求解器返回的界面数据质量有所提高来解释；这些数据用作流体的边界条件。

[7]乍看之下，表5中的顶盖驱动方腔算例里，有限体积流动求解器的一次不动点迭代比其有限元-有限元对应求解器中的流体求解器执行一次牛顿步开销更大，这似乎有些奇怪。然而，这两个框架不仅使用了不同的软件，还运行在不同的硬件基础设施上，因此无法直接比较。

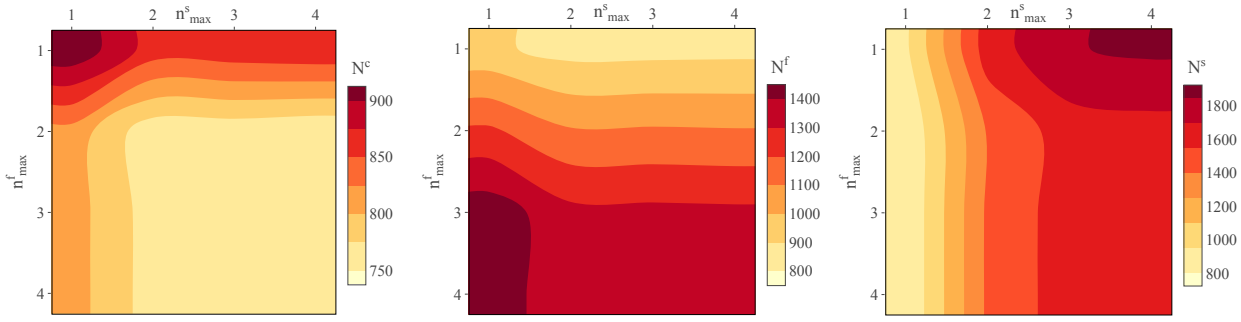
(c) 关于固体迭代 的等高线图及其特征，基本上是(b)中讨论的流动迭代相关内容的转置：N^s 主要取决于n^s_{max}，但也受n^f_{max} 影响。

为简洁起见，省略了两种缺失组合的等高线图，即采用有限元-有限元设置的顶盖驱动方腔，以及采用有限体积-有限元求解器框架模拟的管道算例。它们呈现出相同的趋势，表明上述讨论在一定程度上具有普遍性，因为其潜在影响并不十分依赖于具体问题，或者至少在典型的流固耦合模拟中可以观察到。



(a) 耦合迭代 N^c。 (b) 流动子问题迭代 N^f。 (c) 固体子问题迭代 N^s。

Figu图 9：采用有限体积-有限元框架模拟的顶盖驱动方腔算例中，不同迭代次数随 n^f_{max} 和 n^s_{max} 的变化 .



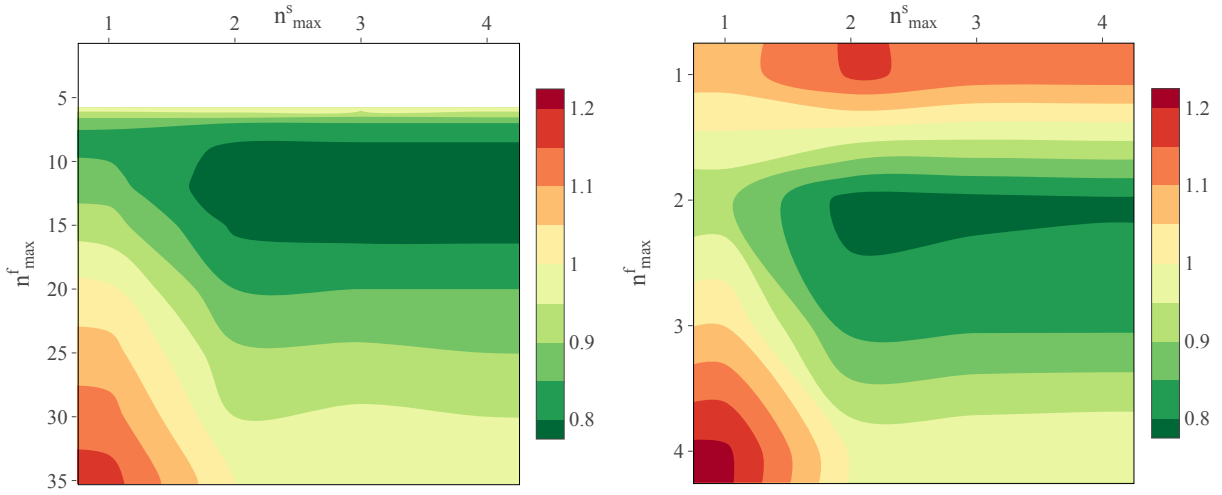
(a) 耦合迭代 N^c。 (b) 流动子问题迭代 N^f。 (c) 固体子问题迭代 N^s。

图10：采用有限元-有限元框架模拟的管道算例中，不同迭代次数随 n^f_{max} 和 n^s_{max} 的变化情况。

5.3.2. 计算成本

研究迭代次数 N^c、N_f、N^s 当然具有科学意义，但在实践中，最重要的考量是模拟总运行时间；本文采用公式（10）中定义的等效时间度量来表示这一时间。由于该度量是 N^c、N_f和 N^s的加权和，因此上一节讨论的所有影响因素也会影响等效时间度量。不过，这些因素的重要性取决于权重因子，即由回归模型近似得到的每次耦合、流动或固体迭代的成本。与图9和图10所示的更一般的依赖关系不同，等效时间度量还强烈依赖于问题本身、求解器框架、高性能计算架构等其他方面。

对于顶盖驱动方腔算例，两种求解器框架的参数研究得到了图 11 所示的等高线图。为便于比较，将数值除以每次求解器调用迭代至完全收敛所需的等效时间进行归一化。最值得注意的是，尽管这些图分别来自运行于不同计算机系统上的不同求解器框架，但其总体特征非常相似。唯一显著的差异出现在图的顶部，即n_{fmax}取值较小时。如上一节所述，有限体积-有限元框架在$n^f_{max} < 6$时发散，而有限元-有限元设置似乎不存在下限，因为每次流动求解器调用仅执行一次牛顿迭代就足以确保收敛。尽管如此，



(a) 有限体积-有限元框架。

(b) 有限元-有限元框架。

图 11：顶盖驱动方腔算例的等效时间度量等高线图，并以完全收敛时得到的值进行归一化。

$n^f_{max} = 1$ 对稳定性的负面影响会导致耦合迭代次数增加，从而提高计算成本。

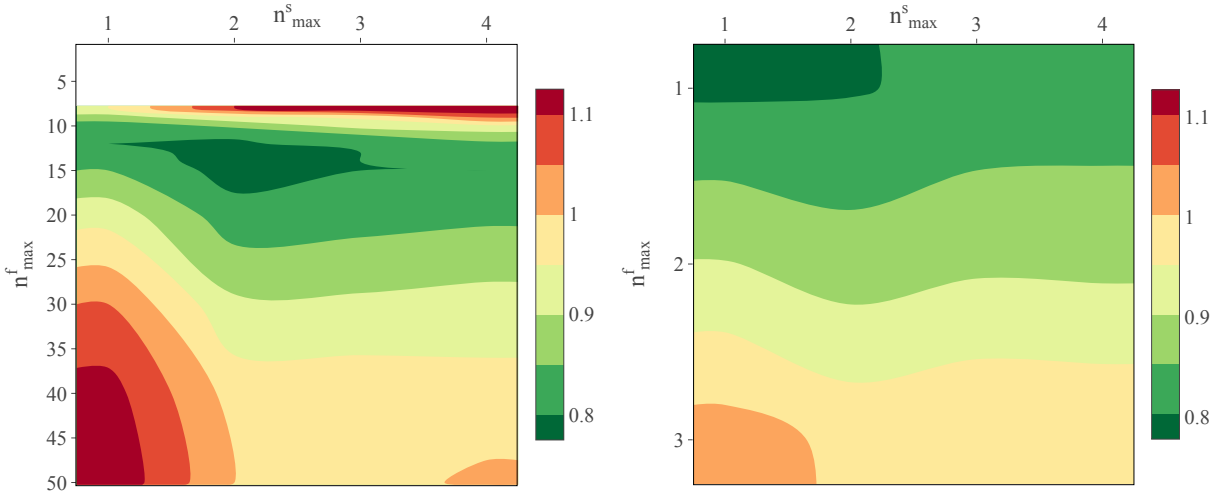
除此之外，两个图中呈现的趋势也非常一致。图中清楚地表明，并非每次调用求解器都迭代至完全收敛，才能使总运行时间最短。因此，运行时间和耦合迭代次数N^c 的最优值并不一致。此外，在两种情况下，效率最低的选择都是设置$n^s_{max} = 1$ 和$n^f_{max} = \infty$。另一方面，计算成本最低的区域对应于两个参数的一些中间值，例如有限元-有限元框架中的$n_{fmax} = n^s_{max} = 2$，或有限体积-有限元框架中的$n^f_{max} \approx 12$和$n^s_{max} = 2$。虽然进一步增大n^f_{max} 会显著降低效率，但在这种情况下增加固体迭代次数并无太大影响，因为对这两种设置而言，求解固体问题的成本都低得多。

尽管相较于每次求解器调用都完全收敛的情形，所能实现的最大加速幅度（即 $n^f_{max} = n^s_{max} = \infty$）仅约为 22%，但顶盖驱动方腔算例的参数研究清楚地表明，每次求解器调用的子问题迭代次数会显著影响计算成本。因此，流固耦合求解器框架若要实现高效运行，就应将这一影响纳入考量。

与顶盖驱动方腔的讨论类似，图 12 展示了柔性管案例中计算成本随 n^f_{max} 和 n^s_{max} 变化的情况。毫无疑问，与上一幅图相比，两个求解器框架之间的差异更加显著。不过，最大的差异仍出现在 n^f_{max}取值非常小时。与前一个案例类似，有限体积流动求解器需要达到一定的子问题迭代次数下限（此处为 $n_{fmax} \gtrsim 8$），才能避免发散。另一方面，对于有限元-有限元框架，每次调用进行一次牛顿迭代不仅足以使其收敛，在本案例中甚至能带来最低的计算成本。一种可能的解释是，一次牛顿迭代已经足够精确，能够获得与有限体积-有限元框架中 $n^f_{max} \approx 12$附近绿色区域类似的结果，因此有限元-有限元设置不会进入其上方收敛稍慢的区域。对此推断的支持是，在参数研究的所有有限元流体求解器调用中，最多进行三次牛顿迭代就足以收敛，这表明对于该测试案例，牛顿步是处理流动问题非线性的非常有效的方法。

在这两幅图中，每次求解器调用的流动迭代次数增加，会使计算成本平稳但显著地上升，而固体迭代的影响则小得多。此外，对于这两种框架而言，设置 $n^s_{max} = 1$ 和 $n^f_{max} = \infty$ 的成本最高。就这一点而言，这两幅图相似。然而，与有限元-有限元情况相比，有限体积-有限元情况中的这一最大值明显得多。根据图10中的三幅等高线图，显然，左下角的峰值源于流体求解器的成本。表5所列的成本因子并未表明，与其他成本相比，有限体积-有限元框架中的流体求解器成本高于有限元-有限元框架中的相应成本，因此，这种差异

这可能是由于流动迭代次数N_f在局部增加得更多所致。附录 B 中的表格支持这一假设。



(a) 有限体积-有限元框架。

(b) 有限元-有限元框架。

图 12：柔性管案例的等效时间度量等高线图，并以完全收敛时得到的数值进行归一化。

5.3.3. 进一步说明

迄今讨论的等高线图展示了参数研究中最重要的趋势和发现。此外，附录 B 包含四张表格，分别对应一项特定的参数研究，即一个测试案例和一种框架。表格列出了该项参数研究中所有运行所得到的耦合、流动和固体迭代次数，以及相对等效时间。以下补充观察和说明值得注意：

- 对于两个测试案例，有限元-有限元框架所需的耦合迭代次数略少。这一现象是由子问题迭代类型不同造成的。由于牛顿步的残差降低远比不动点迭代显著，因此有限元求解器中第一个满足收敛容差的子问题残差通常会显著低于其 FV 对应求解器中的残差。因此，即使残差容差相同，反馈至耦合循环的有限元结果平均而言也更准确。根据第 4.2 节和第 5.3.1 节中的分析，这会使N_c降低。由于每次求解器调用至少包含一次子问题迭代（有限元求解器执行牛顿步，FV 求解器执行不动点迭代），即使子问题已经收敛，因而这一影响还会进一步放大。耦合迭代次数较多本身并无大碍，但这会导致拟牛顿耦合技术中的模式数量增加，从而有可能出现（几乎）线性相关的情况。这就需要采用高效的筛选技术，尤其是在复用参数较高时。

  然而，数值实验表明，对于 FV 求解器，可以通过批次执行不动点子问题迭代（例如每批十次），并且仅在每批迭代结束后检查流体求解器是否收敛，来抵消观察到的耦合迭代次数增加。

- 在文献中，通常通过监测界面位移的不动点残差$\mathbf{r}_k$来判断耦合是否收敛，并设定绝对或相对容差。为说明参数研究中收敛准则的严格程度，表6列出了四项研究中每个时间步收敛时确定的 $\|\mathbf{r}_k\|_2$ 和 $\frac{\|\mathbf{r}_k\|_2}{\|\mathbf{r}_k\|_2}$ 的平均值和最大值。

- 如注4所述，只有当 IQN 更新增量为零时，收敛准则才完全精确。因此，表6列出了所有研究中每个时间步结束时 IQN 增量范数的平均值和最大范数，这表明在实践中可以放心地忽略这一细微因素。

	$\|\mathbf{r}^k\|_2$		$\frac{\|\mathbf{r}^k\|_2}{\|\mathbf{d}^k\|_2}$		$\|\Delta\tilde{\mathbf{d}}^k_{IQN}\|_2$		$\frac{\|\mathbf{d}k - \mathbf{d}kref\|_2}{\sqrt{n_\Gamma}}$	
	平均值	Max	平均值	Max	平均值	Max	平均值	Max
顶盖驱动方腔								
有限元-有限元	1.98E−9	6.58E−7	7.92E−9	7.84E−6	8.53E−10	5.80E−7	5.12E−10	7.36E−9
有限体积-有限元	2.54E−11	9.04E−10	1.18E−9	1.49E−6	4.95E−11	9.14E−10	3.99E−11	1.93E−10
柔性管								
有限元-有限元	1.42E−11	1.90E−10	2.18E−8	1.10E−6	9.85E−12	1.56E−10	7.48E−12	2.63E−11
有限体积-有限元	9.99E−12	2.33E−10	9.04E−9	5.44E−7	1.40E−11	2.56E−10	2.99E−12	8.34E−12

表6：参数研究中所有模拟的每个时间步结束后评估的四种不同量。报告了平均值和最大值。首先给出界面变形不动点残差的范数$\mathbf{r}^k$，随后给出其相对于界面变形$\mathbf{d}^k$的相对大小。第三种量是 IQN 增量的范数 $\Delta\tilde{\mathbf{d}}^k_{IQN}$，最后一种量评估其与$n^f_{max} = n^s_{max} = \infty$所得结果之间的偏差。

- 凭借新的收敛准则，参数研究中的所有运行都以相同的容差收敛。这样，它们的结果几乎相同。为量化偏差，在参数研究中每次运行的每个时间步之后，将计算得到的流固耦合界面位移场$\mathbf{d}_k$与每次调用$\mathbf{d}_{k\,ref}$时迭代至完全收敛所得的位移场进行比较，计算

$$\frac{\left\|\mathbf{d}^k - \mathbf{d}^k_{ref}\right\|_2}{\sqrt{n_\Gamma}}, \tag{14}$$

其中n_Γ为流固耦合界面处的位移自由度数。表6的最后两列列出了各项研究中该偏差的平均值和最大值。

附录 B 中的表格表明，计算效率的提升幅度和最优值的位置取决于具体问题及所选求解器。然而，耦合技术和设置也有显著影响。为说明这一点，附录 C 针对柔性管案例给出了另外两项参数研究，两者设置相同，但未在 IQN 雅可比近似中重用以往时间步的数据（q=0）。对于这两张表，在每次调用中都使子问题完全收敛会导致性能最差。最优方案是在有限元中执行 1 次牛顿迭代，并在有限体积中执行约 8 次不动点迭代。请注意，采用这些方案时，计算时间约为参考值的一半。

5.3.4. 成本因子的影响

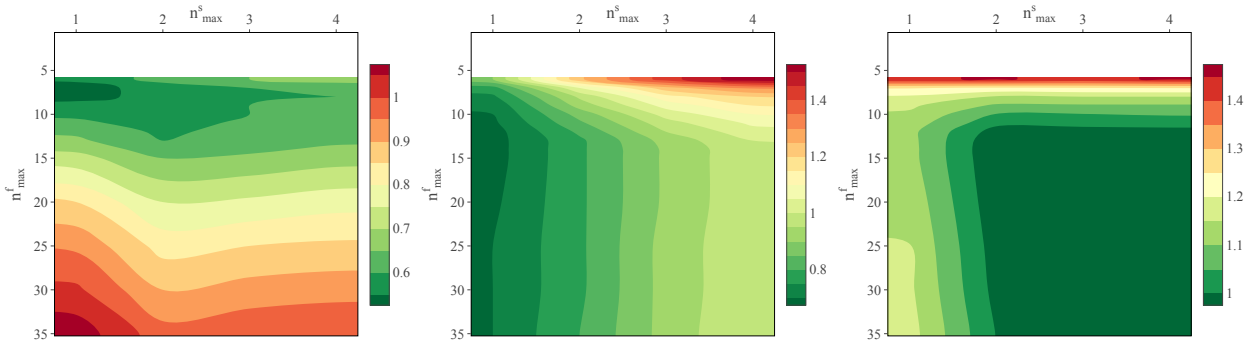
参数研究表明，限制每次求解器调用的子问题迭代次数，会显著影响分区模拟的计算成本；而基于迭代次数 N^c、N_f 和 N^s 的加权和计算得出的新等效时间度量，能够准确反映这一成本。然而遗憾的是，权重因子和迭代次数都因案例而异：权重因子表示单次迭代的成本，因此尤其取决于软件框架和计算机体系结构；迭代次数则会受到所选求解器容差和收敛准则等因素的影响。

尽管如此，第 5.3.1 节表明，这些迭代次数遵循相当普遍的趋势。因此，利用参数研究中获得的迭代次数，并人为设定成本因子 C^c、C^f iter 和 C^s iter，即可近似估算不同成本情景下的计算成本。

作为一个示例，图 13 使用有限体积—有限元框架中腔体案例的迭代次数，绘制了采用不同人为设定的成本因子时产生的计算成本，分别模拟了计算成本较低的流体求解器（图 13b）或成本极高的数据交换（图 13c）的效果。

6. 结论

本研究探讨非线性子问题迭代对分区方案计算成本的影响。尽管本研究重点关注用于流固耦合问题的分区算法，但其



(a) $\bar{C}^{*} = C^{f}_{\text{iter}} = C^{s}_{\text{iter}} = 1$：每次调用仅运行少量子问题迭代，可减少迭代总数迭代 $N^{c}+N_{f}+N^{s}$，这与该场景的总体成本成正比。

(b) $\bar{C}^{*} = C^{s}_{\text{iter}} = 1, C^{f}$ 迭代 $= 0.01$：在这种情况下，流动迭代的成本很低，因此较高的 n^{f}_{max} 以及较小的 n^{s}_{max} 是最高效的。

(c) $\bar{C}^{*} = 120, C^{f}_{\text{iter}} = C^{s}_{\text{iter}} = 1$：如果每个耦合步的成本成为主要因素，那么最好运行每次调用更多子问题迭代，因为这样可以减少 N^{c}。因此，潜在的加速效果会降低。

图 13：腔体案例中不同成本分布的等高线图。底层近似数据通过采用有限体积-有限元框架获得的迭代次数，并人为设定每次耦合迭代和子问题迭代的成本生成。

研究结果也广泛适用于任何通过耦合黑箱求解器求解的多场问题。虽然流固耦合文献通常只考虑耦合迭代总次数，但研究表明，子问题迭代对总体计算成本同样至关重要。因此，尽可能减少耦合迭代次数并不一定能缩短分区模拟的运行时间。为提高效率，耦合算法还应减少流动求解器和固体求解器中的子问题迭代次数。因此，本研究提出的成本度量根据每次耦合迭代、流动迭代或固体迭代的成本，对这些相互冲突的目标进行加权。除此之外，成本因子还取决于所考虑的问题、所采用的求解器以及计算机硬件，因此因具体情况而异。

尽管如此，本研究通过开展多项参数研究，基于典型基准问题和常见求解器类型，揭示了子问题迭代与计算成本之间的关系。这些研究系统地考察了限制每次求解器调用的最大子问题迭代次数，如何影响耦合、流动和固体迭代的总次数以及所需的运行时间。为确保所有参数集都能得到相同的解场，本文提出了一种新的收敛准则，该准则仅依赖于子问题残差的收敛容差，而不像文献中通常的做法那样，另行引入不动点残差的容差。显然，这种评估收敛性的新方法也可应用于其他流固耦合模拟。

参数研究的关键发现是，限制每次求解器调用的子问题迭代次数，不仅能够减少子问题迭代的总次数，还能显著降低模拟的总体计算成本。这引出了一个问题：每次求解器调用进行多少次子问题迭代，才能使分区算法的效率最高？

遗憾的是，这个问题既重要又不简单。目前尚无明确答案，而且预计答案会因具体情况而异。本文讨论的结果表明，在每次求解器调用中都迭代至完全收敛，通常会造成额外的计算开销。相比之下，在每个耦合步中只进行少量子问题迭代，效率可能高得多。在四项参数研究中，计算成本最多降低了 20% 以上。根据耦合技术的不同，节省幅度甚至可能更大；例如，在 IQN 中不重用过去的时间步数据时，研究结果显示计算成本几乎减半。理想的选择可能很难甚至无法事先确定，但结果表明，最优值附近的性能变化相当平缓。例如，对于所研究的所有测试案例和框架，将每次求解器调用的牛顿迭代次数限制为 2 次、不动点迭代次数限制为 12 次，可带来 12% 至 24% 的加速。虽然这些性能提升或许称不上飞跃，但值得注意的是，这些提升无需付出额外努力或成本，也不需要修改代码。

此外，为了让参数研究更简洁、更易于解读，本文将每次求解器调用的子问题迭代次数上限保持不变。以“智能”的方式动态调整 n^{f}_{max} 和 n^{s}_{max}，例如依据某种耦合数据质量度量进行调整，有望进一步缩短运行时间。

总而言之，本研究的主要目标是提高人们对当前流固耦合及其他耦合问题相关文献中一个研究空白的认识：每次求解器调用所执行的子问题迭代次数，以及该次数对分区算法计算成本的影响。此外，本研究也为未来可能开展的研究奠定了基础，以填补这一空白。

致谢

T. Spenke 衷心感谢 JARA Vergabegremium 授予的计算时间，该计算资源由 JARA Partiti on 提供，使用的是亚琛工业大学 CLAIX 超级计算机的一部分。

N. Del aiss 谨此感谢弗兰德斯研究基金会（FW O).

附录 A. 其他回归度量

在第 5.2 节中，以采用 FV-FE 方法对柔性管进行的模拟为例，说明了获取成本因子的回归方法。表 A.7 总结了其他参数研究中各项回归的准确度。

除 RRMSE 外，还给出了绝对误差指标，即均方根误差（RMSE）

$$\mathrm{RMSE} = \sqrt{\frac{\sum^{m} \left| t^{f} - (N^{c} \cdot C_{\mathrm{fix}} + N^{f} \cdot C_{\mathrm{iter}}) \right|^{2}}{m}}. \tag{A.1}$$

	耦合		Flow		固体		新的等效时间		文献中的度量	
	均方根误差	相对均方根误差	均方根误差	相对均方根误差	均方根误差	相对均方根误差	平均绝对百分比误差	最大绝对百分比误差	MAPE	最大绝对百分比误差
顶盖驱动方腔										
有限元-有限元	0.2 秒	7.88 %	1.7 秒	0.44 %	0.3 秒	1.26 %	0.33 %	0.89 %	22.77 %	34.32 %
有限体积-有限元	13.8 秒	11.95 %	468.7 秒	5.60 %	6.3 秒	3.74 %	3.86 %	15.76 %	12.29 %	36.20 %
柔性管										
有限元-有限元	2.8 秒	1.86 %	7.3 秒	0.32 %	0.8 秒	0.26 %	0.22 %	0.58 %	12.70 %	33.32 %
有限体积-有限元	2.7 秒	2.65 %	233.2 秒	7.43 %	27.7 秒	3.44 %	3.80 %	14.62 %	13.58 %	52.69 %

表 A.7：所有参数研究中耦合、流动和固体回归的质量度量，以及使用新度量$C_{\mathrm{simulation}}$和文献中采用的度量（即仅考虑耦合迭代次数）确定的实际运行时间与等效运行时间之间的偏差。

附录 B. 附加结果

为提高可读性，第 5 节的讨论、表格和图示每次只聚焦于一种趋势。为求完整，本附录针对每项参数研究，将所有发现汇总在一张表中，列出该研究每次运行的等效时间度量、耦合迭代次数和子问题迭代次数。表 B.8 和表 B.9 分别列出了采用有限元-有限元框架和有限体积-有限元框架的顶盖驱动方腔算例结果。同样，采用有限元-有限元设置的柔性管案例结果列于表 B.10，而对应的有限体积-有限元算例结果列于表 B.11。

附录 C. 无重用时的其他结果

表 C.12 和表 C.13 分别展示了有限元-有限元和有限体积-有限元框架的另外两项参数研究。研究采用相同设置模拟柔性管案例，但在 IQN 雅可比近似中不重用先前时间步的数据（q=0）。

每次耦合迭代的牛顿迭代次数——结构求解器

Newton iterations per coupling iteration - Flow solver	1		2		3		∞	
1	**1.07**	*8554*	**1.15**	*8876*	**1.12**	*8542*	**1.12**	*8499*
	8554	8554	8876	17043	8542	18894	8499	19662
2	**0.91**	*5066*	**0.78**	*4304*	**0.78**	*4293*	**0.79**	*4299*
	9014	5066	7470	7908	7452	9294	7473	10183
3	**1.05**	*4985*	**0.83**	*3989*	**0.84**	*4014*	**0.84**	*4002*
	11098	4985	8489	7278	8515	8729	8458	9579
∞	**1.21**	*5004*	**0.99**	*4007*	**0.99**	*3982*	**1.00**	*3974*
	13487	5004	10853	7314	10761	8667	10786	9523

表 B.8：采用有限元-有限元框架的顶盖驱动方腔算例。行标题和列标题分别包含流体求解器和固体求解器的子问题迭代最大次数，n^f_{max} 和 n^s_{max}。对于每次运行，给出了归一化**等效时间**，以及耦合迭代、流动求解器迭代和固体求解器迭代的次数。

每次耦合迭代的牛顿迭代次数——结构求解器

Fixed-point iterations per coupling iteration - Flow solver	1		2		3		∞	
5	**-**	-	**-**	-	**-**	-	**-**	-
	-	-	-	-	-	-	-	-
6	**0.96**	*7209*	**0.97**	*7204*	**0.95**	*7111*	**0.96**	*7176*
	37477	7209	37581	12900	36929	16148	37399	17802
7	**0.86**	*6218*	**0.85**	*6126*	**0.85**	*6132*	**0.85**	*6138*
	36473	6218	35965	10865	36001	13791	36068	15110
8	**0.84**	*5882*	**0.81**	*5613*	**0.81**	*5618*	**0.81**	*5608*
	38449	5882	36629	9926	36706	12648	36590	13896
9	**0.84**	*5692*	**0.79**	*5328*	**0.79**	*5318*	**0.79**	*5310*
	40841	5692	38382	9443	38163	11991	38127	13234
10	**0.85**	*5581*	**0.78**	*5091*	**0.78**	*5075*	**0.78**	*5103*
	43394	5581	39559	8943	39430	11354	39631	12683
11	**0.86**	*5485*	**0.77**	*4906*	**0.77**	*4885*	**0.77**	*4888*
	45919	5485	40814	8613	40662	10865	40715	12124
12	**0.87**	*5398*	**0.76**	*4706*	**0.76**	*4718*	**0.76**	*4720*
	48329	5398	41953	8279	42020	10519	41978	11790
13	**0.89**	*5387*	**0.76**	*4607*	**0.76**	*4602*	**0.76**	*4588*
	51042	5387	43364	8056	43320	10201	43319	11447
14	**0.91**	*5373*	**0.77**	*4566*	**0.77**	*4545*	**0.77**	*4546*
	53842	5373	45288	7982	45168	10091	45386	11354
15	**0.92**	*5352*	**0.79**	*4558*	**0.78**	*4534*	**0.78**	*4534*
	56336	5352	47478	7966	47241	10068	47344	11351
20	**1.01**	*5349*	**0.85**	*4519*	**0.85**	*4487*	**0.85**	*4500*
	67992	5349	56902	7903	56645	9958	56739	11219
25	**1.07**	*5357*	**0.91**	*4511*	**0.91**	*4498*	**0.90**	*4483*
	76627	5357	64779	7896	64833	9994	64530	11213
30	**1.12**	*5345*	**0.95**	*4499*	**0.96**	*4503*	**0.95**	*4491*
	83527	5345	71215	7886	71336	9997	71008	11211
∞	**1.17**	*5348*	**1.00**	*4523*	**1.00**	*4492*	**1.00**	*4509*
	89966	5348	77618	7922	77344	9983	77272	11221

表 B.9：采用有限体积-有限元框架的顶盖驱动方腔算例。行标题和列标题分别包含流体求解器和固体求解器的子问题迭代最大次数，n^f_{max} 和 n^s_{max}。对于每次运行，给出了归一化**等效时间**，以及耦合迭代、流动求解器迭代和固体求解器迭代的次数。缺失值表示耦合未能收敛。

每次耦合迭代的牛顿迭代次数——结构求解器

Newton iterations per coupling iteration - Flow solver	1		2		3		∞	
1	**0.79**	*920*	**0.80**	*863*	**0.82**	*866*	**0.82**	*870*
	920	920	863	1626	866	1840	870	1879
2	**0.91**	*815*	**0.88**	*757*	**0.89**	*758*	**0.89**	*752*
	1218	815	1109	1414	1112	1623	1107	1642
3	**1.03**	*811*	**0.99**	*763*	**1.00**	*761*	**1.00**	*755*
	1461	811	1328	1426	1326	1628	1321	1647
∞	**1.03**	*811*	**0.99**	*763*	**1.00**	*761*	**1.00**	*755*
	1461	811	1328	1426	1326	1628	1321	1647

表 B.10：采用有限元-有限元框架的柔性管案例。行标题和列标题分别给出了流体求解器和固体求解器的子问题迭代最大次数，n^f_{max} 和 n^s_{max}。对于每次运行，均给出了归一化的**等效时间**，以及耦合迭代、流动求解器迭代和固体求解器迭代的次数。

每次耦合迭代的牛顿迭代次数——结构求解器

Fixed-point iterations per coupling iteration - Flow solver	1		2		3		∞	
7	**1.14**	*2135*	**-**	-	**-**	-	**-**	-
	13567	2135	-	-	-	-	-	-
8	**0.95**	*1716*	**1.10**	*1814*	**-**	-	**1.16**	*1749*
	12120	1716	13006	3471	-	-	12438	5006
9	**0.88**	*1544*	**0.94**	*1517*	**0.96**	*1481*	**1.06**	*1586*
	11896	1544	11838	2847	11581	3492	12381	4358
10	**0.82**	*1400*	**0.86**	*1348*	**0.92**	*1387*	**0.94**	*1385*
	11758	1400	11351	2488	11722	3225	11667	3646
11	**0.81**	*1355*	**0.81**	*1245*	**0.85**	*1254*	**0.88**	*1279*
	12242	1355	11281	2291	11430	2865	11542	3275
12	**0.80**	*1299*	**0.79**	*1189*	**0.83**	*1201*	**0.84**	*1196*
	12701	1299	11457	2175	11614	2695	11584	3055
13	**0.82**	*1299*	**0.78**	*1154*	**0.80**	*1143*	**0.81**	*1133*
	13455	1299	11767	2093	11717	2552	11621	2835
14	**0.84**	*1289*	**0.77**	*1114*	**0.80**	*1114*	**0.83**	*1137*
	14155	1289	12101	2036	12085	2462	12278	2852
15	**0.85**	*1278*	**0.78**	*1104*	**0.80**	*1101*	**0.80**	*1076*
	14872	1278	12604	2013	12540	2430	12156	2658
20	**0.93**	*1264*	**0.82**	*1062*	**0.83**	*1049*	**0.84**	*1046*
	18399	1264	14898	1914	14672	2280	14534	2570
30	**1.05**	*1255*	**0.91**	*1050*	**0.91**	*1031*	**0.92**	*1033*
	23462	1255	18756	1873	18217	2233	18154	2518
40	**1.12**	*1264*	**0.98**	*1069*	**0.98**	*1043*	**0.97**	*1010*
	26140	1264	21584	1922	21056	2265	20296	2486
50	**1.14**	*1273*	**1.00**	*1064*	**0.98**	*1021*	**1.01**	*1037*
	26768	1273	22083	1932	21210	2206	21368	2564
∞	**1.14**	*1263*	**0.99**	*1052*	**0.98**	*1030*	**1.00**	*1028*
	27089	1263	21898	1892	21292	2206	21282	2543

表 B.11：采用有限体积-有限元框架的柔性管案例。行标题和列标题分别给出了流体求解器和固体求解器的子问题迭代最大次数，n^f_{max} 和 n^s_{max}。对于每次运行，均给出了归一化的**等效时间**，以及耦合迭代、流动求解器迭代和固体求解器迭代的次数。缺失值表示耦合未能收敛。

每次耦合迭代的牛顿迭代次数 - 结构求解器

Newton iterations per coupling iteration - Flow solver	1		2		3		∞	
1	**0.53**	*1465*	**0.69**	*1417*	**0.76**	*1417*	**0.77**	*1417*
	1465	1465	1417	2734	1417	3330	1417	3362
2	**0.67**	*1463*	**0.82**	*1416*	**0.90**	*1416*	**0.91**	*1416*
	2573	1463	2479	2732	2479	3328	2479	3360
3	**0.77**	*1463*	**0.92**	*1416*	**1.00**	*1416*	**1.00**	*1416*
	3318	1463	3206	2732	3206	3328	3206	3360
∞	**0.77**	*1463*	**0.92**	*1416*	**1.00**	*1416*	**1.00**	*1416*
	3318	1463	3206	2732	3206	3328	3206	3360

表 C.12：使用有限元-有限元框架模拟的柔性管案例，但未在 IQN 雅可比近似中重用过去的数据。格式与前面的表格相同。通过该参数研究的回归模型确定的成本因子（单位为秒）为：C^f iter = 0.2177、C^s_{iter} = 0.2253 和 C^c = 0.1620。

每次耦合迭代的牛顿迭代次数 - 结构求解器

Fixed-point iterations per coupling iteration - Flow solver	1		2		3		∞	
5	**0.54**	*2077*	**0.61**	*2120*	**0.61**	*2044*	**0.64**	*2060*
	9641	2077	9815	3970	9444	4721	9511	5286
6	**0.50**	*1834*	**0.55**	*1827*	**0.57**	*1827*	**0.59**	*1828*
	10097	1834	10052	3427	10050	4299	10045	4892
7	**0.50**	*1746*	**0.54**	*1757*	**0.57**	*1763*	**0.59**	*1756*
	10960	1746	10973	3261	10986	4101	10965	4715
8	**0.50**	*1691*	**0.54**	*1681*	**0.57**	*1685*	**0.59**	*1685*
	11881	1691	11844	3101	11880	3928	11899	4582
9	**0.50**	*1638*	**0.54**	*1610*	**0.57**	*1620*	**0.59**	*1633*
	12889	1638	12767	2994	12811	3807	12841	4490
10	**0.51**	*1595*	**0.55**	*1578*	**0.57**	*1583*	**0.59**	*1593*
	13941	1595	13729	2923	13758	3727	13772	4418
11	**0.52**	*1579*	**0.55**	*1555*	**0.58**	*1559*	**0.60**	*1559*
	15046	1579	14713	2865	14741	3672	14741	4360
12	**0.54**	*1572*	**0.56**	*1531*	**0.59**	*1523*	**0.61**	*1520*
	16166	1572	15756	2820	15747	3608	15745	4302
13	**0.55**	*1564*	**0.58**	*1520*	**0.60**	*1515*	**0.62**	*1515*
	17309	1564	16836	2787	16814	3577	16815	4280
14	**0.57**	*1562*	**0.59**	*1511*	**0.61**	*1509*	**0.63**	*1508*
	18463	1562	17905	2767	17872	3565	17863	4267
15	**0.59**	*1563*	**0.60**	*1502*	**0.63**	*1500*	**0.65**	*1500*
	19615	1563	18969	2746	18915	3541	18914	4246
20	**0.67**	*1564*	**0.67**	*1483*	**0.70**	*1474*	**0.72**	*1470*
	25033	1564	24210	2684	24124	3471	24107	4178
30	**0.80**	*1564*	**0.81**	*1479*	**0.83**	*1472*	**0.85**	*1472*
	34388	1564	33540	2676	33450	3461	33440	4170
40	**0.90**	*1564*	**0.91**	*1479*	**0.93**	*1472*	**0.95**	*1472*
	41504	1564	40657	2675	40565	3461	40556	4170
50	**0.93**	*1564*	**0.94**	*1479*	**0.96**	*1472*	**0.99**	*1472*
	43672	1564	42838	2675	42749	3461	42741	4170
∞	**0.93**	*1564*	**0.96**	*1479*	**0.98**	*1472*	**1.00**	*1472*
	43677	1564	43793	2675	43704	3461	43697	4170

表 C.13：使用有限体积-有限元框架模拟的柔性管案例，但未在 IQN 雅可比近似中重用过去的数据。格式与前面的表格相同。通过该参数研究的回归模型确定的成本因子（单位为秒）为：C^f iter = 0.1139、C^s_{iter} = 0.2410 和 C^c = 1.2825。

参考文献

[1] C. A. Felippa、K.-C. Park、C. Farhat, 《耦合机械系统的分区分析》, 《计算力学与工程中的计算方法》190 (24-25) (2001) 3247–3270。 doi:10.1016/s0045-7825(00)00391-1。[2] P. Causin、J.-F. Gerbeau、F. Nobile, 《流固问题分区算法设计中的附加质量效应》, 《计算力学与工程中的计算方法》194 (42–44) (2005) 4506–4527。 doi:10.1016/j.cma.2004.12.005。[3] C. F¨orster、W. Wall、E. Ramm, 《非线性结构与不可压缩黏性流顺序交错耦合中的人工附加质量不稳定性》, 《计算力学与工程中的计算方法》196 (7) (2007) 1278–1293。 doi:10.1016/j.cma.2006.09.002。[4] H. van Brummelen, 《流固耦合中可压缩流与不可压缩流的附加质量效应》, 《应用力学杂志》76 (2) (2009) 021206–1–7。 doi:10.1115/1.3059565。[5] L. Formaggia、J.-F. Gerbeau、F. Nobile、A. Quarteroni, 《顺应性血管流动问题中三维与一维纳维-斯托克斯方程的耦合》, 《计算力学与工程中的计算方法》191 (6–7) (2001) 561–582。 doi:10.1016/s0045-7825(01)00302-4。[6] J. Vierendeels、K. Dumont、E. Dick、P. Verdonck, 《刚体运动流固耦合算法的分析与稳定化》, 《美国航空航天学会期刊》43 (12) (2005) 2549–2557。 doi:10.2514/1.3660。[7] S. Badia、F. Nobile、C. Vergara, 《基于罗宾传递条件的流固分区过程》, 《计算物理杂志》227 (14) (2008) 7027–7051。 doi:10.1016/j.jcp.2008.04.006。[8] M. M. Joosten、W. G. Dettmer、D. Peri´c, 《强耦合模型问题中块高斯–赛德尔求解过程的分析及其在流固耦合中的应用》, 《国际工程数值方法杂志》

78 7 2009 757 778 doi:()() 10.1002/nme.2503 – 。 。[9] D. P. Mok 、 W. A. Wall 、 E. Ramm, 《非定常流固耦合的加速迭代子结构方法》, 载于: K.-J. Bathe （编）, 《计算流体与固体力学》, 爱思唯尔, 2001年, 第1325–1328页。 doi:10.1016/b978-008043944-0/50907-0。[10] D. P. Mok、W. A. Wall, 《不可压缩流与非线性柔性结构瞬态相互作用的分区分析方案》, 载于: K. Schweizerhof、W. A. Wall、K.-U. Bletzinger （编）, 《计算结构力学新趋势》, CIMNE, 西班牙巴塞罗那, 2001年, 第688–698页。[11] U. K¨uttler、W. Wall, 《采用动态松弛的定点流固耦合求解器》, 《计算力学》43 (1) (2008) 61–72。doi:10.1007/s00466-008-0255-5。[12] J. Degroote、P. Bruggeman、R. Haelterman、J. Vierendeels, 《流固耦合应用中分区求解器耦合技术的稳定性》, 《计算机与结构》86 (23–24) (2008) 2224–2234。 doi:10.1016/j.compstruc.2008.05.005。[13] J. Degroote、S. Annerel、J. Vierendeels, 《流固耦合分区模拟中高斯–赛德尔迭代的稳定性分析》, 《计算机与结构》88 (5–6) (2010) 263–271。 doi:10.1016/j.compstruc.2009.09.003。[14] N. Delaiss´e、T. Demeester、D. Fauconnier、J. Degroote, 《黑箱求解器耦合中不同拟牛顿技术的比较》, 载于: F. Chinesta、R. Abgrall、O. Allix、M. Kaliske（编）, 14th 2020年世界计算力学大会暨欧洲计算力学协会大会, 论文集, CIMNE, 法国巴黎, 2021年, 第1–12页。 doi:10.23967/wccm-eccomas.2020.088。[15] N. Delaiss´e、T. Demeester、R. Haelterman、J. Degroote, 《广义布罗伊登框架下流固耦合分区模拟拟牛顿方法的综述》, 《工程计算方法档案》, 即将出版（2023）。[16] K.-J. Bathe, 《有限元过程》, 普伦蒂斯·霍尔出版社, 新泽西州萨德尔河上城, 1996年。[17] F. Nobile、C. Vergara, 《利用广义罗宾条件实现血管动力学的有效流固耦合公式》, 《SIAM科学计算杂志》30 (2) (2008) 731–763。 doi:10.1137/060678439。[18] T. Spenke、M. Make、N. Hosters, 《采用拟牛顿加速的流固耦合罗宾–诺伊曼格式》, 《国际工程数值方法杂志》124 (4) (2023) 979–997。 doi:10.1002/nme.7151。[19] T. Spenke、N. Hosters、M. Behr, 《一种具有线性复杂度的流固耦合分区多向量界面拟牛顿方法》, 《计算力学与工程中的计算方法》361 (2020) 112810。 doi:10.1016/j.cma.2019.112810。[20] N. Delaiss´e、T. Demeester、D. Fauconnier、J. Degroote, 《流固耦合模拟中拟牛顿技术的基于代理模型的加速》, 《计算机与结构》260 (2022) 106720。 doi:10.1016/j.compstruc.2021.106720。[21] F. Lindner、M. Mehl、K. Scheufele、B. Uekermann, 《各种流固耦合分区拟牛顿格式的比较》, 载于: B. Schrefler、E. O˜nate、M. Papadrakakis（编）, 6th 科学与工程耦合问题计算方法国际会议, CIMNE, 2015年, 第477–488页。[22] J. Degroote、K.-J. Bathe、J. Vierendeels, 《流固耦合中一种新型分区过程与整体式过程的性能比较》, 《计算机与结构》87 (11-12) (2009) 793–801。 doi:10.1016/j.compstruc.2008.11.013。[23] L. Pauli、M. Behr, 《各向异性网格上的稳定化时空有限元法: 不可压缩纳维-斯托克斯方程及其在医疗器械血流中的应用》, 《国际流体数值方法杂志》85 (3) (2017) 189–209。 doi:10.1002/fld.4378。[24] J. Donea、A. Huerta, 《流动问题有限元方法》, 约翰·威立父子公司, 2003年。 doi:10.1002/0470013826。[25] D. Forti、L. Ded`e, 《高性能计算框架下采用VMS-LES建模的纳维-斯托克斯方程半隐式BDF时间离散》, 《计算与流体》117 (2015) 168–182。 doi:10.1016/j.compfluid.2015.05.011。[26] A. A. Johnson、T. E. Tezduyar, 《具有移动边界与界面的流动问题并行有限元计算中的网格更新策略》, 《计算力学与工程中的计算方法》119 (1-2) (1994) 73–94。 doi:10.1016/0045-7825(94)00077-8。[27] M. Behr、F. Abraham, 《倾斜壁面附近的自由表面流动模拟》, 《计算力学与工程中的计算方法》191 (47-48) (2002) 5467–5483。 doi:10.1016/S0045-7825(02)00444-9。[28] C. F¨orster、W. A. Wall、E. Ramm, 《变形区域瞬态流动计算中的几何守恒律》, 《国际流体数值方法杂志》50 (12) (2006) 1369–1379。 doi:10.1002/fld.1093。[29] T. J. Hughes、J. A. Cottrell、Y. Bazilevs, 《等几何分析：计算机辅助设计、有限元、非均匀有理B样条、精确几何与网格加密》, 《计算力学与工程中的计算方法》194 (39-41) (2005) 4135–4195。 doi:10.1016/j.cma.2004.10.008。[30] J. A. Cottrell、T. J. Hughes、Y. Bazilevs, 《等几何分析：迈向计算机辅助设计与有限元分析的集成》, 约翰·威立父子公司, 2009 doi:。 。 10.1002/9780470749081.ch7

[31] J. Chung、G. Hulbert, 《一种具有改进数值耗散的结构动力学时间积分算法：广义-α 方法》, 《应用力学杂志》60 (1993) 371。 doi:10.1115/1.2900803。[32] S. Erlicher、L. Bonaventura、O. S. Bursi, 《广义-α 方法在非线性动力学问题中的分析》, 《计算力学》28 (2) (2002) 83–104。 doi:10.1007/s00466-001-0273-z。[33] N. Hosters, 《流固耦合的基于样条的方法》, 博士论文, 亚琛工业大学, 德国亚琛（ 2018 doi:）。。 10.18154/RWTH-2018-223770
[34] M. K. P. Make, 《气动热弹性问题的基于样条的方法》, 博士论文, 亚琛工业大学, 德国亚琛（2021年）。 doi:10.18154/RWTH-2021-03363。[35] ANSYS公司, ANSYS Fluent理论指南, 2019R3（2019年8月）。[36] T. J. Barth、D. Jespersen, 《非结构网格上迎风格式的设计与应用》, 载于：美国航空航天学会（编）, 27th 航空航天科学会议, 美国航空航天学会, 1989 doi:10.2514/6.。 1989-366
。[37] C. M. Rhie 、 W. L. Chow, 《具有后缘分离的翼型湍流数值研究》, 《美国航空航天学会期刊》 21 (11) (1983 年) 1525–1532。 doi:10.2514/3.8284。[38] J. Batina, 《采用非结构动态网格的非定常欧拉翼型解》, 《美国航空航天学会期刊》28 (8) (1990年) 1381–1388。 doi:10.2514/3.25229。[39] V. M. Ferr´andiz、P. Bucher、R. Rossi、R. Zorrilla、J. Cotela、J. Maria、M. A. Celigueta、G. Casas, Kratosmultiphysics/kratos：9.2版, Zenodo（2022年）。 doi:10.5281/ZENODO.3234644。[40] M. Lombardi、N. Parolini、A. Quarteroni, 《流固耦合问题中用于网格间插值与网格运动的径向基函数》, 《计算力学与工程中的计算方法》256 (2013) 117–131。 doi:10.1016/j.cma.2012.12.019。[41] J.-D. M¨uller, 《计算流体力学基础》, 泰勒与弗朗西斯出版集团, 2015年。[42] H. Versteeg、W. Malalasekera, 《计算流体力学导论》, 朗文科技出版社, 埃塞克斯, 英格兰, 1995年。[43] J. Ferziger、M. Peric, 《计算流体力学数值方法》, 施普林格柏林海德堡, 2002年。[44] O. C. Zienkiewicz、R. L. Taylor、J. Z. Zhu, 《有限元法：基础与原理》, 爱思唯尔, 2013 doi:10.1016/。。 C2009-0-24909-9
[45] O. C. Zienkiewicz、R. L. Taylor, 《固体与结构力学有限元法》, 爱思唯尔, 2005年。[46] J. N. Reddy, 《有限元法导论》, 麦格劳-希尔教育, 2019年。[47] J. N. Reddy、D. K. Gartling, 《传热与流体动力学中的有限元法》, CRC出版社, 2010年。[48] Y. Saad、M. H. Schultz, 《广义最小残差法：求解非对称线性系统的广义最小残差算法》, 《SIAM科学与统计计算杂志》7 (3) (1986) 856–869。 doi:10.1137/0907058。[49] K. St
¨ 128 1 2 2001 281 309 doi:10.1016/ s0377-0427(00)00516-1
uben, 《代数多重网格综述》, 《计算与应用数学杂志》 (-) （） –。。[50] J. Degroote, 《流固耦合分区模拟》, 《工程计算方法档案》20 (3) (2013) 185–238。doi:10.1007/s11831-013-9085-5。[51] K. Scheufele、M. Mehl, 《用于并行流固耦合模拟及其他多物理场应用的稳健多割线拟牛顿变体》, 《SIAM科学计算杂志》39 (5) (2017) 404–433。 doi:10.1137/16M1082020。[52] R. Schussnig、D. R. Pacheco、T.-P. Fries, 《涉及不可压缩广义牛顿流的流固耦合高效分步格式》, 《计算机与结构》260 (2022) 106718。 doi:10.1016/j.compstruc.2021.106718。[53] D. P. Mok, 《结构动力学与流固耦合中的分区求解方案》, 博士论文, 斯图加特大学, 德国斯图加特（2001年）。 doi:10.18419/OPUS-147。[54] G. Vald´es, 《包含流固耦合的正交各向异性膜与壳结构非线性分析》, 博士论文, 加泰罗尼亚理工大学, 西班牙巴塞罗那（2007年）。