MORe DWR：基于增量 POD 的 ROM 的时空目标导向误差控制

Hendrik Fis cher[1,2], Julian Roth[1,2], Thomas Wick[1,2], Ludovic Chamoin[2], 和 Amélie Fau[2]

[1]汉诺威莱布尼茨大学，应用数学研究所，科学计算工作组，Welfengarten 1, 30167 汉诺威，德国[2]巴黎-萨克雷大学，中央理工-高等电力学院，巴黎-萨克雷高等师范学院，法国国家科学研究中心，LMPS - 巴黎-萨克雷力学实验室，91190 吉夫叙尔伊韦特，法国

arXiv:2304.01140v1 [math.NA] 3 Apr 2023

摘要

本文采用对偶加权残差（DWR）方法，构建经过认证的基于增量本征正交分解（POD）的降阶模型。我们提出了一种名为 MORe DWR（基于对偶加权残差误差估计的模型降阶）的新方法。该方法将张量积时空降阶建模、时间分片、增量 POD 基生成与基于对偶加权残差估计的目标导向误差控制相结合。在仿真过程中对目标泛函的误差进行估计；若估计值超过给定阈值，则更新 POD 基。这使得在解的行为发生意料之外的变化时，能够自适应地增广 POD 基；这在许多实际应用中尤为重要。因此，可以跳过离线阶段，直接使用从第一个时间片上的解中提取的 POD 基求解降阶模型，并在必要时于仿真过程中利用高保真有限元解即时增广 POD 基。由此可将全阶模型求解次数降至最低，这一点通过热方程和弹性动力学的数值测试得到了验证。

1 引言

本征正交分解（POD）模型降阶（MOR）已被应用于大量偏微分方程（PDE），以低成本进行代理建模 [70, 46,52, 6, 75, 13, 42, 12, 1, 76, 33]。在此过程中，将动力学投影到一组 POD 模态上，这些模态构成了解流形的近似基，从而降低昂贵的高保真模拟的运行成本。这种基于本征正交分解的降阶建模（POD- ROM）属于真实解近似，因为它能对先验已知的解轨迹进行压缩表示。为避免事先进行这些昂贵的高保真模拟，我们使用误差估计，仅在局部执行高保真计算。

本文采用对偶加权残差法在降阶模型与高保真计算之间进行切换。时空对偶加权残差（DWR）方法是DWR

方法的扩展，适用于定常问题，最早由 [10, 11, 8], 提出，其基础是Johnson及其合作者的开创性工作 [30]。Oden发表了关于将其用于自适应预测性多尺度建模的最新综述 [55]。Schmich (Besier)和Vexler已将时空DWR方法应用于抛物型偏微分方程 [68], Schmich (Besier) [67] 以及Besier和Rannacher的研究中 [16], 作者本人也开展了相关工作 [73, 63]。此外，Rademacher在其学位论文中将该方法应用于双曲型偏微分方程 [59] ，Bangerth等人则将其应用于波动方程 [7]。由于误差估计理论是在时空函数空间中建立的，并且需要时空变分形式，因此我们采用时空有限元方法（FEM）离散化；例如参见 [48]。文献 [68, 66] 研究了热方程的时空有限元，文献 [41, 7]研究了弹性动力学方程的时空有限元。在FEniCS中可以找到类似的时空FEM实现 [51] ， NGSolve中也有相应实现 [50, 58]。

近年来，时空公式已被应用于模型降阶 [21, 43, 72, 25],，其中包括一种旨在提高效率的窗口化时空方法 [69]。时空模型降阶的其他应用包括最优控制 [81] ，以及利用离散经验插值进行的经典误差估计和超降阶估计 [14]。Yano 已开展大量关于采用降阶求积规则进行超降阶的 DWR 误差估计研究 [79, 71]。Meyer 和 Matthies 提出另一种采用目标导向误差估计的降阶建模方法 [53], ，利用这些估计去除与准确计算关注量无关的 POD 基向量。最后，相关方法还包括适当广义分解（PGD） [20] 和分层模型（HiMod）降阶 [57, 9, 56],，后者利用估计值对动力学横向方向上的 POD 进行评估。

本文提出一种不同的 POD-ROM计算方法：仅使用计算成本高昂的全阶模型（FOM）计算解轨迹的一小部分，并在误差估计超过预设容差时即时更新降阶模型（ROM）。为此，我们将 POD-ROM 与增量POD及时空对偶加权残差误差估计相结合。我们详细推导算法细节，并最终提出一种新的增量式降阶模型算法。增量POD方法依赖于奇异值分解的加性秩-b 更新 [17, 18] ，并已成功应用于流体流动的增量模型降阶 [45]。总体而言，我们采用时空框架。具体来说，我们采用基于有限元库 deal.II [2, 3]实现的张量积时空有限元法[63] 。最终算法经过实现，并在多种设置下进行了验证，包括抛物型问题（热方程）和二阶双曲型问题（弹性动力学）。主要目标是在保持数值解精度的同时，展示计算成本的降低。此外，我们还根据有效性对误差估计器和目标泛函进行比较。

本文的结构如下：第2节将针对热方程和弹性动力学建立问题模型，并采用张量积时空有限元对其进行离散。接着，第3节回顾基于POD的降阶建模，并介绍其向张量积时空 POD-ROM 的扩展。随后，第4节阐述时空误差估计和

增量式模型降阶的理论。第5节针对热方程和弹性动力学，在 1+1D、 2+1D 和 3+1D 中开展数值测试。最后，第6节总结本文的研究发现。

2 问题表述与离散化

2.1 模型问题表述

令 $d \in \mathbb{N}$，其中 d 取决于问题是向量值还是标量值，即对于热方程，我们有 $d = 1$；而对于采用 u 形式的弹性动力学（其中 u 表示位移），我们有 $d = d$；对于 (u, v) 形式（其中 u 与前面相同，且 v 表示速度），我们有 $d = 2d$，其中 $d \in \{1, 2, 3\}$ 为空间维数。在问题描述中，$I := (0, T)$ 表示时间域，$\Omega \subset \mathbb{R}^d$ 表示一个足够光滑的空间域。这里，空间边界分为狄利克雷边界 $\Gamma_D \subseteq \partial\Omega$ 和诺伊曼边界 $\Gamma_N \subsetneq \partial\Omega$，且 $\Gamma_D \cap \Gamma_N = \emptyset$。我们考虑如下抽象时变问题：求 $u : \bar{\Omega} \times \bar{I} \to \mathbb{R}^d$ 使得

$$
\begin{aligned}
\partial_t u + \mathcal{A}(u) &= f && \text{in } \Omega \times I, \\
u &= u_D && \text{on } \Gamma_D \times I, \\
\mathcal{B}(u) &= g_N && \text{on } \Gamma_N \times I, \\
u &= u^0 && \text{in } \Omega \times \{0\},
\end{aligned}
\tag{1}
$$

其中，空间算子 $\mathcal{A}$可能是非线性的，边界算子 $\mathcal{B}$ 以及足够正则的右端项f。选取合适的连续空间函数空间 $V := V(\Omega)$、连续时间泛函空间 $X := X(I, \cdot)$以及时变索博列夫空间 $X(I, V(\Omega))$，该空间从 I 映射到 $V(\Omega)$，我们可以定义连续时空变分形式如下：求 $u \in u_D + X(I, V(\Omega))$，使得

$$
\begin{aligned}
A(u)(\varphi) &:= (\!(\partial_t u, \varphi)\!) + (\!(\mathcal{A}(u), \varphi)\!) + (u(0), \varphi(0)) \\
&= (\!(f, \varphi)\!) + \langle\!\langle g_N - \mathcal{B}(u), \varphi \rangle\!\rangle_{\Gamma_N} + (u^0, \varphi(0)) =: F(\varphi) \qquad \forall \varphi \in X(I, V(\Omega)),
\end{aligned}
$$

其中我们采用如下记号

$$
\begin{aligned}
(f, g) &:= (f, g)_{L^2(\Omega)} := \int_\Omega f \cdot g \, \mathrm{d}x, \qquad (\!(f, g)\!) := (f, g)_{L^2(I, L^2(\Omega))} := \int_I (f, g) \, \mathrm{d}t, \\
\langle f, g \rangle &:= \langle f, g \rangle_{L^2(\Gamma)} := \int_\Gamma f \cdot g \, \mathrm{d}s, \qquad \langle\!\langle f, g \rangle\!\rangle := (f, g)_{L^2(I, L^2(\Gamma))} := \int_I \langle f, g \rangle \, \mathrm{d}t.
\end{aligned}
$$

在此记号中，如果 f 和 g 是标量或向量值的量，$f \cdot g$ 表示欧几里得内积；如果 f 和 g 是矩阵，$f \cdot g$ 则表示弗罗贝尼乌斯内积。我们注意到，属于这一框架的某些偏微分方程（PDE）包括热方程，以及更一般的抛物型问题。稍微滥用一下记号，写成一阶时间系统的弹性动力学也可以表示为上述形式；不过，为了确保数学上的严谨性，我们会在下文中对其作出精确定义。

2.1.1 热方程

¯ ¯热方程的强形式如下：求温度 $u: \Omega \times I \to \mathbb{R}$ 使得

$$\partial_t u - \Delta_x u = f \qquad \text{in } \Omega \times I,$$

其中 $\mathcal{A}(u) := -\Delta_x u$ in (1)。初始条件和边界条件为

$$\begin{aligned} u &= u^0 \qquad \text{on } \Omega \times \{0\}, \\ u &= 0 \qquad \text{on } \partial\Omega \times I. \end{aligned}$$

因此，我们得到以下连续变分形式:

公式 2.1 （热方程的连续变分形式）。
求 $u \in X(I, V(\Omega)) := \{v \in L^2(I, H_0^1(\Omega)) \mid \partial_t v \in L^2(I, (H_0^1(\Omega))^*)\}$ 使得

$$A(u)(\varphi) := ((\partial_t u, \varphi)) + ((\nabla_x u, \nabla_x \varphi)) + (u(0), \varphi(0)) = ((f, \varphi)) + (u^0, \varphi(0)) =: F(\varphi) \qquad \forall \varphi \in X(I, V(\Omega)).$$

对于此变分形式，我们使用 $u_0 \in L^2(\Omega)$ 和 $f \in L^2(I, H_0^1(\Omega)^*)$ [78]。这里 $H_0^1(\Omega)^*$ 表示 $H_0^1(\Omega)$ 的对偶空间。

2.1.2 弹性动力学方程

三维空间中的线性弹性动力学强形式为：求位移 $u: \bar{\Omega} \times \bar{I} \to \mathbb{R}^d$，使得

$$\partial_{tt} u - \nabla_x \cdot \sigma(u) = 0 \qquad \text{in } \Omega \times I,$$

with

$$\begin{aligned} \sigma(u) &= 2\mu E(u) + \lambda \operatorname{tr}(E(u)) \mathbb{1}_{d\times d}, && \text{(stress tensor)} \\ E(u) &= \frac{1}{2}(\nabla_x u + (\nabla_x u)^T), && \text{(linearized strain tensor)} \end{aligned}$$

其中 $\mathbb{1}_{d\times d} \in \mathbb{R}^{d\times d}$是单位矩阵，拉梅参数为 $\mu > 0$和 $\lambda > -\frac{2}{3}\mu$。初始条件为 –

$$\begin{aligned} u &= u^0 \qquad \text{on } \Omega \times \{0\}, \\ \partial_t u &= v^0 \qquad \text{on } \Omega \times \{0\}. \end{aligned}$$

边界条件规定为

$$\begin{aligned} u &= 0 \qquad \text{on } \Gamma_D \times I, \\ \mathcal{B}(u) = \sigma(u) \cdot n &= g_N \qquad \text{on } \Gamma_N \times I. \end{aligned}$$

我们将其转化为时间一阶系统，并求解位移 $u: \bar{\Omega} \times \bar{I} \to \mathbb{R}^d$和速度$v: \bar{\Omega} \times \bar{I} \to \mathbb{R}^d$，使得

$$\begin{aligned} \partial_t v - \nabla_x \cdot \sigma(u) &= f \qquad \text{in } \Omega \times I, \\ \partial_t u - v &= 0 \qquad \text{in } \Omega \times I, \end{aligned}$$

其中 $\mathcal{A}(u,\ v) := -\nabla_x \cdot \sigma(u) - v$ 见式 (1)。初始条件和边界条件仍与之前相同，唯一的区别是现在我们有

$$\begin{aligned} v &= v^0 \qquad \text{on } \Omega \times \{0\}, \\ v &= 0 \qquad \text{on } \Gamma_D \times I. \end{aligned}$$

对于变分形式，我们使用 $u_0 \in H^1_{\Gamma_D,0}(\Omega)^d$，这是在 Γ_D, $v_0 \in L^2(\Omega)^d, g_N \in L^2(I, L^2(\Gamma_N)^d)$ 上取零值的弱可微函数空间，以及函数空间

$$\begin{aligned} X(I, V^u(\Omega)) &:= \{v \in L^2(I, H^1_{\Gamma_D,0}(\Omega)^d) \mid \partial_t v \in L^2(I, L^2(\Omega)^d), \partial_t^2 v \in L^2\left(I, (H^1_{\Gamma_D,0}(\Omega)^d)^*\right)\}, \\ X(I, V^v(\Omega)) &:= \{v \in L^2(I, L^2(\Omega)^d) \mid \partial_t v \in L^2\left(I, (H^1_{\Gamma_D,0}(\Omega)^d)^*\right)\}, \\ X(I, V(\Omega)) &:= X(I, V^u(\Omega)) \times X(I, V^v(\Omega)). \end{aligned}$$

因此，我们求解连续变分形式：

公式 2.2 （弹性动力学方程的连续变分形式）。求 $U = (u, v) \in X(I, V(\Omega))$ 使得

$$A(U)(\Phi) = F(\Phi) \qquad \forall \Phi = (\varphi^u, \varphi^v) \in X(I, V(\Omega)),$$

其中

$$\begin{aligned} A(U)(\Phi) &:= ((\partial_t v, \varphi^u)) + ((\sigma(u), \nabla_x \varphi^u)) + (v(0), \varphi^u(0)) + ((\partial_t u, \varphi^v)) - ((v, \varphi^v)) + (u(0), \varphi^v(0)), \\ F(\Phi) &:= (v^0, \varphi^u(0)) + \langle\langle g_N, \varphi^u \rangle\rangle_{\Gamma_N} + (u^0, \varphi^v(0)). \end{aligned}$$

2.2 张量积时空有限元离散

我们沿用近期关于纳维-斯托克斯方程时空自适应的工作 [63] ，采用张量积时空有限元（FEM），其中时间方向使用间断有限元（dG），空间方向使用连续有限元（cG）。采用时间和空间基函数的张量积，是广泛的时空有限元方法类别中的一个特例 [48]。下面我们以热方程为例说明张量积时空有限元方法；函数空间见 [68] ，而逐时间片的张量积时空实现则在 [73]中概述。我们假设空间网格保持固定，这简化了分析和实现。此外，我们还概述了将这一方法推广到弹性动力学的方式。

2.2.1 时间离散化

令 $\mathcal{T}_k := \{I_m := (t_{m-1}, t_m) \mid 1 \le m \le M\}$ 为一个时间划分，即 $\bar{I} = [0, T] = \bigcup_{m=1}^{M} \bar{I}_m$。现在我们引入分片连续层级函数空间

$$\tilde{X}(\mathcal{T}_k, V(\Omega)) := \{v \in L^2(I, L^2(\Omega)) \mid v|_{I_m} \in X(I_m, V(\Omega)) \quad \forall I_m \in \mathcal{T}_k\}$$

用于热方程以及

$$\tilde{X}(\mathcal{T}_k, V^u(\Omega)) := \{v \in L^2(I, L^2(\Omega)^3) \mid v|_{I_m} \in X(I_m, V^u(\Omega)) \quad \forall I_m \in \mathcal{T}_k\},$$
$$\tilde{X}(\mathcal{T}_k, V^v(\Omega)) := \{v \in L^2(I, L^2(\Omega)^3) \mid v|_{I_m} \in X(I_m, V^v(\Omega)) \quad \forall I_m \in \mathcal{T}_k\},$$
$$\tilde{X}(\mathcal{T}_k, V(\Omega)) := \tilde{X}(\mathcal{T}_k, V^u(\Omega)) \times \tilde{X}(\mathcal{T}_k, V^v(\Omega))$$

对于弹性动力学方程。之所以需要这些分片函数空间 [24]，是因为我们希望在时间上进行协调间断伽辽金离散化，因此需要允许时间区间/时间单元之间存在不连续性。由于这些不连续性，我们将函数 f 在时间 t_m 处从上方和下方的极限分别定义为 f

$$f_m^{\pm} := \lim_{\epsilon \searrow 0} f(t_m \pm \epsilon),$$

并将函数值 f 在时间 t_m 处的跳跃定义为

$$[f]_m := f_m^+ - f_m^-.$$

T这些函数空间使我们能够在变分形式中包含不连续性 s:

Formulation 2.3 （热方程的时间间断变分形式）
.˜求 $u \in X(\mathcal{T}_k, V(\Omega))$ 使得

$$\tilde{A}(u)(\varphi) = \tilde{F}(\varphi) \qquad \forall \varphi \in \tilde{X}(\mathcal{T}_k, V(\Omega)),$$

其中

$$\tilde{A}(u)(\varphi) := \sum_{m=1}^{M} \int_{I_m} (\partial_t u, \varphi) + (\nabla_x u, \nabla_x \varphi) \, \mathrm{d}t + \sum_{m=1}^{M-1} ([u]_m, \varphi_m^+) + (u_0^+, \varphi_0^+),$$
$$\tilde{F}(\varphi) := (\!(f, \varphi)\!) + (u^0, \varphi_0^+).$$

公式 2.4 （弹性动力学方程的时间不连续变分形式）。˜求 $U = (u, v) \in X(\mathcal{T}_k, V(\Omega))$ 使得

$$\tilde{A}(U)(\Phi) = \tilde{F}(\Phi) \qquad \forall \Phi = (\varphi^u, \varphi^v) \in \tilde{X}(\mathcal{T}_k, V(\Omega)),$$

其中

$$\begin{aligned}\tilde{A}(U)(\Phi) := \sum_{m=1}^{M} \int_{I_m} &(\partial_t v, \varphi^u) + (\sigma(u), \nabla_x \varphi^u) + (\partial_t u, \varphi^v) - (v, \varphi^v) \, \mathrm{d}t \\ &+ \sum_{m=1}^{M-1} ([v]_m, \varphi_m^{u,+}) + ([u]_m, \varphi_m^{v,+}) + (v_0^+, \varphi_0^{u,+}) + (u_0^+, \varphi_0^{v,+}),\end{aligned}$$
$$\tilde{F}(\Phi) := (v^0, \varphi_0^{u,+}) + \langle\!\langle g_N, \varphi^u \rangle\!\rangle_{\Gamma_N} + (u^0, \varphi_0^{v,+}).$$

˜我们有包含关系 $X(I, \cdot) \subset X(\mathcal{T}_k, \cdot)$，因为对于连续函数，跳跃项会消失，因此变分形式 2.3 和变分形式 2.4 是一致的。

接下来，我们将热方程的半离散空间定义为

$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V(\Omega)) := \left\{ v_k \in L^2(I, L^2(\Omega)) \,\middle|\, v_k|_{I_m} \in P_r(I_m, H_0^1(\Omega)) \right\} \subset \tilde{X}(\mathcal{T}_k, V(\Omega))$$

以及弹性动力学方程为

$$\begin{aligned}
X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^u(\Omega)) &:= \left\{ v_k \in L^2(I, L^2(\Omega)^3) \,\middle|\, v_k|_{I_m} \in P_r(I_m, H^1_{\Gamma_D,0}(\Omega)^3) \right\} \subset \tilde{X}(\mathcal{T}_k, V^u(\Omega)), \\
X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^v(\Omega)) &:= X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^u(\Omega)), \\
X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V(\Omega)) &:= X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^u(\Omega)) \times X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V^v(\Omega)),
\end{aligned}$$

~其中，时空函数空间 $X(\mathcal{T}_k, \cdot)$ 采用 $r \in \mathbb{N}_0$ 阶间断伽辽金方法进行时间离散 $r \in \mathbb{N}_0$ (dG(r))。在我们的工作中，时间次数通常取$r = 1$ 和 $r = 2$。这里， $P_r(I_m, Y)$ 是由时间区间 I_m 映射到空间 Y 的 r 阶多项式空间。图 1 展示了 $r = 1$ 情况下的 dG(r) 时间离散。

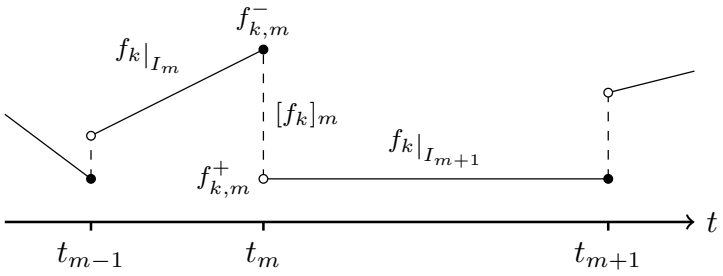


图1：dG(1) 时间离散

时间自由度（DoFs）的位置由求积规则确定。由于时间离散具有不连续性，可以选择多种求积规则，其中最常用的是高斯-洛巴托、高斯-勒让德和高斯-拉杜。在图1中，时间自由度的位置选在各时间区间的端点处，这对应于高斯-洛巴托求积。在第5节中，我们使用高斯-勒让德和高斯-洛巴托时间求积，以展示本方法在时间求积公式选择方面的灵活性。

已在 [23]（另见经典教材 [60, 29]）中推导得出，dG(0) 时间离散化是后向欧拉格式的一种变体。高阶格式也已得到推导，并且已证实， $r_p \in \mathbb{N}_0$ 为多项式次数的 $dG(r_p)$ 离散化通常是隐式且A-稳定的。

2.2.2 空间离散化

对于变分形式的空间离散，我们使用固定网格 $\mathcal{T}_h$，该网格在一维情况下由区间组成，在高维情况下则由四边形（2D）或六面体（3D）单元组成。然后，我们可以使用次数不超过 $s \in \mathbb{N}$ 的逐单元多项式函数作为空间函数空间，即，

$$V_h^s := V_h^s(\mathcal{T}_h) := \left\{ v \in C(\bar{\Omega}) \middle| v|_K \in \mathcal{Q}_s(K) \quad \forall K \in \mathcal{T}_h \right\}$$

以及对于弹性动力学方程

$$V_h^{s,u} := V_h^{s,u}(\mathcal{T}_h) := \left\{ v \in C(\bar{\Omega})^d \middle| v|_K \in (\mathcal{Q}_s(K))^d \quad \forall K \in \mathcal{T}_h \right\} =: V_h^{s,v}(\mathcal{T}_h) =: V_h^{s,v},$$

其中， $\mathcal{Q}_s(K)$ 是通过将主单元 $(0,1)^d$ 上的 s 次张量积多项式映射到单元 K 上构造的。热方程的全离散函数空间为

$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s) := \left\{ v_{kh} \in L^2(I, L^2(\Omega)) \,\middle|\, v_{kh}|_{I_m} \in P_r(I_m, V_h^s) \quad \forall I_m \in \mathcal{T}_k \right\}$$

以及对于弹性动力学方程

$$X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s) := \left\{ v_{kh} \in L^2(I, L^2(\Omega)^{2d}) \,\middle|\, v_{kh}|_{I_m} \in P_r(I_m, V_h^s) \quad \forall I_m \in \mathcal{T}_k \right\},$$
$$V_h^s := V_h^{s,u} \times V_h^{s,v}.$$

因此，热方程的全离散变分形式为:

求 $u_{kh} \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s)$，使得

$$\tilde{A}(u_{kh})(\varphi_{kh}) = \tilde{F}(\varphi_{kh}) \quad \forall \varphi_{kh} \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s).$$

此外，弹性动力学方程的全离散变分形式为：求 $U_{kh} := (u_{kh}, v_{kh}) \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s)$，使得

$$\tilde{A}(U_{kh})(\Phi_{kh}) = \tilde{F}(\Phi_{kh}) \quad \forall \Phi_{kh} = (\varphi_{kh}^u, \varphi_{kh}^v) \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k, V_h^s).$$

2.2.3 分片离散

最后，我们想指出，全离散变分形式不必在整个时空柱 $\Omega \times I$ 上求解，也可以按时空片依次求解

$$S_l^n := \Omega \times \left(\bigcup_{m=l}^{n} I_m \right),$$

其中 $1 \le l \le n \le M$，另请参见 [73][注释 2.1]。如前所述，随后可通过对空间有限元基函数与时间有限元基函数取张量积，在 S_l^n 上得到时空有限元法基。这样可以简化抽象时变问题 (1) 的有限元离散，因为主要前提是拥有一个用于求解 Ω 中定常问题 $\mathcal{A}(u) = f$ 的有限元方法代码。此外，张量积时空有限元方法在时间离散的选择上具有更大的灵活性，因为只需改变时间有限元的多项式次数，即可改变时空离散的时间次数。由于时空有限元基具有张量积结构，因此很容易采用基于本征正交分解（POD）的降阶建模；从抽象层面来看，只需将空间有限元基替换为空间本征正交分解基即可。

对于具有 $n-l+1$ 个时间区间的时空片 S_l^n上的热方程，可得到如下线性方程组：

$$\begin{pmatrix} A & & & & \mathbf{0} \\ B & A & & & \\ & B & A & & \\ & & \ddots & \ddots & \\ \mathbf{0} & & & B & A \end{pmatrix} \begin{pmatrix} U_l \\ U_{l+1} \\ U_{l+2} \\ \vdots \\ U_n \end{pmatrix} = \begin{pmatrix} F_l - BU_{l-1} \\ F_{l+1} \\ F_{l+2} \\ \vdots \\ F_n \end{pmatrix} \tag{2}$$

简言之

$$A_{S_l^n} U_{S_l^n} = F_{S_l^n} \tag{3}$$

with

$$\begin{aligned} A &= C_k \otimes M_h + M_k \otimes K_h, \\ B &= -D_k \otimes M_h, \end{aligned}$$

其中我们使用空间矩阵

$$\begin{aligned} M_h &= \left\{ (\varphi_h^{(j)}, \varphi_h^{(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)}, \\ K_h &= \left\{ (\nabla_x \varphi_h^{(j)}, \nabla_x \varphi_h^{(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)} \end{aligned}$$

以及时间矩阵

$$\begin{aligned} M_k &= \left\{ \int_{I_m} \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)}, \\ C_k &= \left\{ \int_{I_m} \partial_t \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t + \varphi_{k,m-1}^{(j),+} \cdot \varphi_{k,m-1}^{(i),+} \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)}, \\ D_k &= \left\{ \varphi_{k,m-1}^{(j),-} \cdot \varphi_{k,m-1}^{(i),+} \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)}. \end{aligned}$$

注意，U_l，...，U_n本身就是时空向量，其中 $U_m \in \mathbb{R}^{\#\mathrm{DoFs}(I_m)\cdot \#\text{自由度}(\mathcal{T}_h)}$中，$m = l$，...，$n$是解 u_{kh}在时间区间 I_m上的系数向量，即对于时间上的 dG(r) 方法，给定时间求积点 t_1，...，t_{r+1}，我们有

$$U_m = \begin{pmatrix} U_m(t_1) \\ \vdots \\ U_m(t_{r+1}) \end{pmatrix}, \quad m = 1, \ldots, M,$$

其中 M 是时间区间的总数。特别地，如果我们使用仅包含一个时间单元的时空片，那么只需求解线性系统

$$AU_m = F_m - BU_{m-1}$$

对于每个时间片 $S_m \coloneqq S_m^m = \mathcal{T}_h \times I_m$。为提高效率，本文余下部分仅考虑大小为一的此类时间片。

对于弹性动力学方程，可以类似地推导出时空有限元法线性系统。采用时间方向上的高斯-洛巴托求积时，dG(1) 和 dG(2) 的线性系统及时间步进公式见 A。

注2.5. 尽管本节中热方程的线性系统以及附录 *A* 中弹性动力学方程的线性系统都表示为空间矩阵的张量积，但张量积时空有限元方法可应用于范围广得多的问题。例如，当偏微分方程包含依赖于空间和时间的系数时，时空线性系统并不总能分解为这种张量积结构。尽管如此，我们对张量积时空有限元方法的实现足够通用，也能处理这类问题，因为它不依赖于线性系统的张量积结构，只依赖于有限元基的张量积结构。

3 降阶建模

3.1 POD-ROM

过去几十年来计算能力的提升，使得人们能够利用高性能计算开展大规模数值模拟。尽管如此，在某些情形下，例如多物理场问题，高性能计算的成本可能很高，尤其还会带来巨大的碳足迹和能源消耗。在这些情况下，人们希望通过采用模型降阶（MOR）技术获得显著的计算加速比，以满足这些需求。本文主要研究基于投影的降阶基方法（RBM） [38, 40, 15, 49, 37, 65, 54] ，因为该方法旨在通过提供近似求解过程和高效的误差估计 [38]，实现高效处理。这里的关键观察是：不采用具有一般逼近性质的投影空间（例如有限元法），而是选择问题特定的逼近空间，随后用其对原问题进行离散化 [64]。基于这些空间，并假设解在低维解流形上平滑演化（等价于 Kolmogorov N-宽度较小[44, 13, 40]），便可构造降阶模型（ROM），用显著更少的自由度 [64]，以足够精度表征所关注的物理问题。

为构造降阶空间，我们按照第 2.2 节所述，通过全阶模型的解对解流形进行经验探索。随后，对这些高保真解的快照进行本征正交分解（POD），以获得降阶基函数
[49, 13, 47, 61, 70, 77, 22, 36, 19, 5]。定理 3.1 指出，POD 基在最小二乘意义下是最优的。证明由 Gubisch 和 Volkwein 在 [35] 中给出。

定理 3.1（POD基）。设 $Y = [Y_1, \dots, Y_q] := [U_1(t_1) \dots, U_1(t_{r+1})\ U_2(t_1) \dots, U_M(t_{r+1})] \in \mathbb{R}^{n \times q}$ *为快照矩阵，其中每个时间自由度对应一个（空间）列向量，且其自由度数为* $q = M \cdot \#\text{自由度}(I_m)$*、* $n = \#$*自由度(*$\mathcal{T}_h$*)，秩为* $d \leq \min(n, q)$*。此外，设* $Y = \Psi \Sigma \Phi^T$ *为其奇异值分解，其中* $\Sigma = \mathrm{diag}(\sigma_1, \dots, \sigma_d) \in \mathbb{R}^{d \times d}$*，且* $\Psi = [\psi_1, \dots, \psi_d] \in$ *为正交矩阵*

$\mathbb{R}^{n\times d}$, $\Phi = [\phi_1, \ldots, \phi_d] \in \mathbb{R}^{q\times d}$。那么对于 $1 \le N \le d$，优化问题

$$\min_{\tilde{\psi}_1,\ldots,\tilde{\psi}_N \in \mathbb{R}^n} \sum_{j=1}^{q} \left\| Y_j - \sum_{i=1}^{N} \left(Y_j, \tilde{\psi}_i\right)_{\mathbb{R}^n} \tilde{\psi}_i \right\|_{\mathbb{R}^n}^2 \quad s.t. \quad (\tilde{\psi}_i, \tilde{\psi}_j)_{\mathbb{R}^n} = \delta_{ij} \ \forall 1 \le i,j \le N \qquad (\mathbf{P}^N)$$

其中 $\{\tilde{\psi}_i\}_{i=1}^N \subset \mathbb{R}^n$，并且通过左奇异向量 $\{\psi_i\}_{i=1}^N \subset \mathbb{R}^n$ 求解，且有

$$\sum_{j=1}^{q} \left\| Y_j - \sum_{i=1}^{N} (Y_j, \psi_i)_{\mathbb{R}^n} \psi_i \right\|_{\mathbb{R}^n}^2 = \sum_{i=N+1}^{d} \sigma_i^2 = \sum_{i=N+1}^{d} \lambda_i. \tag{4}$$

因此，奇异值的衰减速率对于 POD 方法的可行性至关重要。如果截断奇异值的平方和对于相对较小的N而言足够小，我们就可以利用少数几个基函数 ψ_i 的线性组合，良好地近似高维有限元空间中Y_j 的元素。尽管可以通过公式 (4) 确定所得秩为N 的近似的误差，但这并不能直观地衡量秩判定。因此，常用的启发式标准是通过 POD 基的保留能量或信息含量 $\varepsilon(N)$ 来确定其质量，参见 [34, 35, 49]。后者定义为

$$\varepsilon(N) = \frac{\sum_{\mathrm{i}=1}^{N} \sigma_i^2}{\sum_{i=1}^{d} \sigma_i^2} = \frac{\sum_{i=1}^{N} \sigma_i^2}{\sum_{i=1}^{\mathrm{q}} ||U_i||^2}. \tag{5}$$

接下来介绍 POD 基的构造。算法1中，我们根据快照矩阵的行列比介绍不同的方法。为此，我们部分依据 Gräßle等人于 [13][第 2]章中的工作。

算法1 $\mathbb{R}^n$中的POD基生成

输入： 快照 $\{Y_j\}_{j=1}^q \subset \mathbb{R}^n$ 和能量阈值 $\varepsilon \in [0,1]$。**输出：** POD 基 $\{\boldsymbol{\psi}_i\}_{i=1}^N \subset \mathbb{R}^n$ 和特征值 $\{\lambda_i\}_{i=1}^N$。

Set $Y = [Y_1, \ldots, Y_q] \in \mathbb{R}^{n\times q}$.

如果 $n \approx q$ **则**

计算奇异值分解 $[\Psi, \Sigma, \Phi] =$ 奇异值分解(Y)。计算 $N =$最小 $\{N \in \mathbb{N} \mid \varepsilon(N) \ge \varepsilon, \ 1 \le N \le d\}$。设置 $\lambda_i = \Sigma_{ii}^2$ 和$\boldsymbol{\psi}_i = \Psi_{\cdot,}$ $_i \in \mathbb{R}^n$，其中 $1 \le i \le N$。

否则，如果 $n \ll q$ **则**

  Compute eigenvalue decomposition $[\Psi, \Lambda] = \mathrm{Eig}(YY^T)$, where $YY^T \in \mathbb{R}^{n\times n}$.

  Compute $N = \min\{N \in \mathbb{N} \mid \varepsilon(N) \ge \varepsilon, \ 1 \le N \le d\}$.

  Set $\lambda_i = \Lambda_{ii}$ and $\boldsymbol{\psi}_i = \Psi_{\cdot,i} \in \mathbb{R}^n$ for $1 \le i \le N$.

否则，如果 $q \ll n$ **则**

计算特征值分解 $[\Phi, \Lambda] =$ Eig(Y^TY)，其中 $Y^TY \in \mathbb{R}^{q\times q}$。计算 $N =$最小值 $\{N \in \mathbb{N} \mid \varepsilon(N) \ge \varepsilon, \quad 1 \le N \le d\}$。设定 $\lambda_i = \Lambda_{ii}$ 和 $\boldsymbol{\psi}_i = Y\Phi_{\cdot,\ i}/\sqrt{\lambda_i} \in \mathbb{R}^n$，其中 $1 \le i \le N$。 —

3.2 张量积时空 POD-ROM

为了对第 2.2 节中的时空全阶系统 (2) 进行降阶，将一般的空间有限元空间 V_h替换为特定于问题的低维空间 $V_N = \text{span}\{\varphi_N^1, \ldots, \varphi_N^N\}$，该空间通过˜POD 获得。由此得到降阶变分形式：求 $u_N \in X(\mathcal{T}_k, V_N)$，使得

$$\tilde{A}(u_N)(\varphi) = \tilde{F}(\varphi) \qquad \forall \varphi \in \tilde{X}(\mathcal{T}_k, V_N).$$

降　　阶基矩阵可以通过拼接降阶基向量 vi 形成，　　z.

$$Z_N = \begin{bmatrix} \varphi_N^1 & \ldots & \varphi_N^N \end{bmatrix} \in \mathbb{R}^{\#\text{DoFs}(\mathcal{T}_h) \times N}. \tag{6}$$

随后，具有 $n - l + 1$ 个时间区间的时空片 S_l^n 的分片离散可类比于第 2.2.3 节中的全阶模型获得。对于热方程，我们采用式（2）中描述的线性方程组，并以仿射方式对给定矩阵进行降阶。因此，我们得到

$$\begin{pmatrix} A_N & & & & \mathbf{0} \\ B_N & A_N & & & \\ & B_N & A_N & & \\ & & \ddots & \ddots & \\ \mathbf{0} & & & B_N & A_N \end{pmatrix} \begin{pmatrix} U_{N_l} \\ U_{N_{l+1}} \\ U_{N_{l+2}} \\ \vdots \\ U_{N_n} \end{pmatrix} = \begin{pmatrix} F_{N_l} - B_N U_{N_{l-1}} \\ F_{N_{l+1}} \\ F_{N_{l+2}} \\ \vdots \\ F_{N_n} \end{pmatrix} \tag{7}$$

或简写为

$$A_N U_{N,S_l^n} = F_{N,S_l^n} \tag{8}$$

其中降阶分量为

$$A_N = Z_N^T A Z_N, \tag{9a}$$

$$B_N = Z_N^T B Z_N, \tag{9b}$$

$$F_{N_i} = Z_N^T F_i, \quad l \le i \le n. \tag{9c}$$

4 基于后验误差估计器认证的降阶建模

为便于进一步分析，我们考虑齐次狄利克雷边界条件，以简化表述˜，即 $u_D = 0$。设目标泛函 $J\colon X(\mathcal{T}_k, V(\Omega)) \to \mathbb{R}$ 的形式为

$$J(u) = \int_0^T J_1(u(t))\, \mathrm{d}t + J_2(u(T)), \tag{10}$$

给定，用于表示某个物理关注量（QoI）。这里， T 与前文一样表示终止时间。现在，我们希望减小精细解 u^{fine}与粗略解 u^{coarse}的关注量之差，即

$$J(u^{\text{fine}}) - J(u^{\text{coarse}}) \tag{11}$$

同时满足时变问题（1）的变分形式。精细解和粗略解可以有以下几种选择： $u^{\text{fine}} := u \in X(I, V(\Omega))$ $u^{\text{coarse}} := u_k \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V(\Omega))$，用于控制时间离散引起的误差；或者 $u^{\text{fine}} := u_k \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V(\Omega))$， $u^{\text{coarse}} := u_{kh} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h)$，其中对于热方程， $V_h := V_h^s$；对于弹性动力学方程， $V_h := V_h^s = V_{,h}^{s\ u} \times V_{,h}^{s\ v}$，用于控制空间离散化引起的误差。关于时空误差控制的更多信息，感兴趣的读者可参见 [67, 73, 63]；关于空间误差控制的一般信息，可参见 [10, 11, 8, 26]。作为扩展，本文仅讨论对降阶建模引入的误差进行控制，因此，我们将全阶模型（FOM）解 $u^{\text{fine}} := u_{kh}^{\text{FOM}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V\text{FOM}_h)$作为精细解，将降阶模型（ROM）解 $u^{\text{coarse}} := u_{kh}^{\text{ROM}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V\text{ROM}_h)$作为粗略解，其中 $V_h^{\text{ROM}} \subset V\text{FOM}_h =: V_h$。Meyer和Matthies [53], 率先尝试将对偶加权残差（DWR）方法引入降阶建模：他们计算若干快照并构建降阶基后，使用DWR误差估计器确定误差贡献最大的基向量，并仅将这些向量用于降阶模型。这可以视为对降阶基进行目标导向自适应粗化。本文关注另一项目标，即根据关注量的时间演化增广降阶基。这可以视为对降阶基进行目标导向自适应细化[1]；我们提出该方法，以便在整个时间域内准确且高效地计算解。

4.1 空间—时间对偶加权残差法

对于约束优化问题 (11)，我们将细问题的拉格朗日泛函定义为

$$\begin{aligned}\mathcal{L}_{\text{fine}} : X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}) \times X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}) &\to \mathbb{R}, \\ (u^{\text{fine}}, z^{\text{fine}}) &\mapsto J(u^{\text{fine}}) - \tilde{A}(u^{\text{fine}})(z^{\text{fine}}) + \tilde{F}(z^{\text{fine}}),\end{aligned}$$

粗问题的拉格朗日泛函定义为

$$\begin{aligned}\mathcal{L}_{\text{coarse}} : X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}) \times X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}) &\to \mathbb{R}, \\ (u^{\text{coarse}}, z^{\text{coarse}}) &\mapsto J(u^{\text{coarse}}) - \tilde{A}(u^{\text{coarse}})(z^{\text{coarse}}) + \tilde{F}(z^{\text{coarse}}).\end{aligned}$$

拉格朗日泛函 $\mathcal{L}_{\text{fine}}$和 $\mathcal{L}_{\text{coarse}}$的驻点 $(u^{\text{fine}}, z^{\text{fine}})$ 和 $(u^{\text{coarse}}, z^{\text{coarse}})$ 需要满足 Karush-Kuhn-Tucker 一阶最优性条件。首先，这些驻点是以下方程的解

$$\begin{aligned}\mathcal{L}'_{\text{fine},z}(u^{\text{fine}}, z^{\text{fine}})(\delta z^{\text{fine}}) = 0 \quad \forall \delta z^{\text{fine}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}), \\ \mathcal{L}'_{\text{coarse},z}(u^{\text{coarse}}, z^{\text{coarse}})(\delta z^{\text{coarse}}) = 0 \quad \forall \delta z^{\text{coarse}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}).\end{aligned}$$

[1]原则上也可以进行粗化，但这并非本文的研究目标。若要进行粗化，我们需要遵循 Meyer 和 Matthies [53]的工作。

我们将这些方程称为原始问题，其解称为 u^{fine} 和 u^{coarse} 原始解。其次，驻点还必须满足以下方程

$$\mathcal{L}'_{\text{fine},u}(u^{\text{fine}}, z^{\text{fine}})(\delta u^{\text{fine}}) = 0 \quad \forall \delta u^{\text{fine}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}),$$
$$\mathcal{L}'_{\text{coarse},u}(u^{\text{coarse}}, z^{\text{coarse}})(\delta u^{\text{coarse}}) = 0 \quad \forall \delta u^{\text{coarse}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}).$$

这些方程称为伴随问题或对偶问题，其解 z^{fine} 和 z^{coarse} 是伴随解。

4.1.1 原问题

对拉格朗日泛函 $\mathcal{L}_{\text{fine}}$ 和 $\mathcal{L}_{\text{coarse}}$ 关于伴随解 z求 Gâteaux导数，可得原问题。由于偏微分方程的变分形式对试验函数是线性的，因此有

$$\mathcal{L}'_{\text{fine},z}(u^{\text{fine}}, z^{\text{fine}})(\delta z^{\text{fine}}) = -\tilde{A}(u^{\text{fine}})(\delta z^{\text{fine}}) + \tilde{F}(\delta z^{\text{fine}}) = 0 \quad \forall \delta z^{\text{fine}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}),$$
$$\mathcal{L}'_{\text{coarse},\ z}(u^{\text{coarse}}, z^{\text{coarse}})(\delta z^{\text{coarse}}) = -\tilde{A}(u^{\text{coarse}})(\delta z^{\text{coarse}}) + \tilde{F}(\delta z^{\text{coarse}}) = 0 \quad \forall \delta z^{\text{coarse}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_{\text{降阶模型}h}).$$

我们注意到，可以通过向前推进时间求解原始问题（例如热方程或弹性动力学方程）来获得原始解。

4.1.2 伴随问题

对拉格朗日泛函 $\mathcal{L}_{\text{fine}}$ 和 $\mathcal{L}_{\text{coarse}}$ 关于原始解 u求 Gâteaux 导数，可得

$$\mathcal{L}'_{\text{fine},u}(u^{\text{fine}}, z^{\text{fine}})(\delta u^{\text{fine}}) = J'_u(u^{\text{fine}})(\delta u^{\text{fine}}) - \tilde{A}'_u(u^{\text{fine}})(\delta u^{\text{fine}}, z^{\text{fine}}) = 0$$
$$\forall \delta u^{\text{fine}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{FOM}}),$$
$$\mathcal{L}'_{\text{粗}\quad,u}(u^{\text{coarse}}, z^{\text{coarse}})(\delta u^{\text{coarse}}) = J'_u(u^{\text{coarse}})(\delta u^{\text{coarse}}) - \tilde{A}'_u(u^{\text{coarse}})(\delta u^{\text{coarse}},\ z^{\text{coarse}}) = 0$$
$$\forall \delta u^{\text{coarse}} \in X_k^{\text{dG}(r)}(\mathcal{T}_k, V_h^{\text{ROM}}).$$

因此，要得到伴随解，我们需要求解一个额外的方程，即伴随问题

$$\tilde{A}'_u(u)(\delta u, z) = J'_u(u)(\delta u). \tag{12}$$

请注意，即使对于非线性偏微分方程和目标泛函，伴随问题也是线性的，因为偏微分方程变分形式中的半线性形式关于试验函数是线性的；然而，众所周知，原始解也会参与其中 [11]。

注释 4.1. 对于线性偏微分方程，例如热方程或弹性动力学方程，伴随问题 *(12)* 的左端可简化为

$$\tilde{A}'_u(u)(\delta u, z) = \tilde{A}(\delta u)(z).$$

对于线性目标泛函，例如均值泛函，伴随问题 *(12)* 的右端项简化为

$$J'_u(u)(\delta u) = J(\delta u).$$

特别地，对于线性问题，即线性偏微分方程和目标泛函，我们有伴随问题

$$\tilde{A}(\delta u)(z) = J(\delta u), \tag{13}$$

这不再依赖原始解 u。

根据注释（4.1），热方程的伴随问题为

$$\begin{aligned}
&\tilde{A}(\delta u)(z) = J'_u(u)(\delta u) \\
\Leftrightarrow &\sum_{m=1}^{M} \int_{I_m} (\partial_t \delta u, z) + (\nabla_x \delta u, \nabla_x z)\ \mathrm{d}t + \sum_{m=1}^{M-1} ([\delta u]_m, z_m^+) + (\delta u_0^+, z_0^+) = J'_u(u)(\delta u).
\end{aligned}$$

现在，我们对时间进行分部积分，将时间导数从试验函数 δu 移至伴随解 z，得到

$$\sum_{m=1}^{M} \int_{I_m} (\delta u, -\partial_t z) + (\nabla_x \delta u, \nabla_x z)\ \mathrm{d}t - \sum_{m=1}^{M-1} (\delta u_m^-, [z]_m) + (\delta u_M^-, z_M^-) = J'_u(u)(\delta u).$$

弹性动力学方程的伴随问题可以用类似的方法推导为

$$\begin{aligned}
&\sum_{m=1}^{M} \int_{I_m} (\delta v, -\partial_t z^u) + (\sigma(\delta u), \nabla_x z^u) + (\delta u, -\partial_t z^v) - (\delta v, z^v)\ \mathrm{d}t \\
&\quad - \sum_{m=1}^{M-1} \big((\delta v_m^-, [z^u]_m) + (\delta u_m^-, [z^v]_m) \big) + (\delta v_M^-, z_M^{u,-}) + (\delta u_M^-, z_M^{v,-}) = J'_U(U)(\delta U).
\end{aligned}$$

我们注意到，伴随问题现在是沿时间反向求解的。

4.1.3 线性问题的误差恒等式与时间局部化

为简化起见，我们假设所讨论的是线性偏微分方程和目标泛函。于是有如下误差恒等式

$$J(u^{\mathrm{fine}}) - J(u^{\mathrm{coarse}}) = -\tilde{A}(u^{\mathrm{coarse}})(z^{\mathrm{fine}}) + \tilde{F}(z^{\mathrm{fine}}) =: \eta. \tag{14}$$

该证明既依赖于目标泛函和偏微分方程的线性，也依赖于伴随问题和原始问题的定义：

$$J(u^{\mathrm{fine}}) - J(u^{\mathrm{coarse}}) = J(u^{\mathrm{fine}} - u^{\mathrm{coarse}}) = \tilde{A}(u^{\mathrm{fine}} - u^{\mathrm{coarse}})(z^{\mathrm{fine}}) = -\tilde{A}(u^{\mathrm{coarse}})(z^{\mathrm{fine}}) + \tilde{F}(z^{\mathrm{fine}}).$$

在空间和时间离散误差控制的 DWR 文献中，这类误差恒等式 (14) 毫无用处，因为对于大多数应用，z^{fine} 是未知的先验解析解，用 z^{coarse} 替代它会导致糟糕的误差估计。因此，在有限元方法离散误差控制中，会使用对偶权重 $z^{\mathrm{fine}} - z^{\mathrm{coarse}}$，它们可以通过对偶解的后处理来近似。然而，在我们的情形中，$z^{\mathrm{fine}} := z_{kh}^{\mathrm{FOM}} \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k,\ V_{\text{全阶模型}h})$ 是全阶模型对偶解，虽然可以计算，但代价高昂。此外，在数值实验中我们将观察到，如果对偶降阶基足够大，使用降阶模型对偶解 $z^{\mathrm{coarse}} := z_{kh}^{\mathrm{ROM}} \in X_k^{\mathrm{dG}(r)}(\mathcal{T}_k,\ \tilde{V}_h^{\mathrm{ROM}})$仍能为我们的问题产生出色的误差估计。

我们指出，如果要捕捉对偶问题的动力学并获得非零误差估计器，对偶空间降阶模型函数空间 $\tilde{V}_h^{\mathrm{ROM}}$ 必须不同于原始空间降阶模型函数空间 V_h^{ROM}。

为了按时间定位误差，我们只需逐时间片组装原始残差 (14)。具体而言，要将误差定位到每个时间区间 I_m，只需分别组装各时间区间上的原始残差。更具体地说，对于热方程，可以根据原始线性方程组、粗网格原始解和精细对偶解计算该时间区间上的误差，方法如下:

$$\eta|_{I_m} = \sum_{i=1}^{\#\mathrm{DoFs}(I_m)} \left\{ (Z_m^{\mathrm{fine}})^T \left(-AU_m^{\mathrm{coarse}} + F_m - BU_{m-1}^{\mathrm{coarse}} \right) \right\}_i . \tag{15}$$

对于弹性动力学，可以类似地利用原问题的线性系统 (27)，推导时间区间 I_m 上的误差估计器。

为了测试误差估计时是否需要使用精细对偶解，还是可以用粗略对偶解替代，我们使用有效性指数来衡量误差估计器的质量。有效性指数是估计误差与真实误差之比，即

$$\mathrm{I}_{\mathrm{eff}} := \left| \frac{\eta}{J(u^{\mathrm{fine}}) - J(u^{\mathrm{coarse}})} \right| . \tag{16}$$

我们希望 $\mathrm{I}_{\mathrm{eff}} \approx 1$，因为这样误差估计器便能可靠地预测降阶建模误差；第5节的数值测试也验证了这一点。

4.1.4 非线性问题的时空对偶加权残差法 s

对于非线性问题，例如第5.2节中目标泛函为非线性的热方程，我们不再像线性情形那样拥有式(14)中的误差恒等式。根据 [11][命题2.3], 中的证明，我们得到以下误差表示公式。

定理4.2 （非线性问题的误差表示）。

$$J(u^{fine}) - J(u^{coarse}) = -\tilde{A}(u^{coarse})(z^{fine}) + \tilde{F}(z^{fine}) + R,$$

其中二次余项为

$$\begin{aligned} R = \int_0^1 \Big[& \tilde{A}''_{uu}(u^{coarse} + s(u^{fine} - u^{coarse}))(u^{fine} - u^{coarse}, u^{fine} - u^{coarse}, z^{fine}) \\ & - J''_{uu}(u^{coarse} + s(u^{fine} - u^{coarse}))(u^{fine} - u^{coarse}, u^{fine} - u^{coarse}) \Big] \cdot s \, \mathrm{d}s. \end{aligned}$$

证明。 下面我们将证明 $R = J(u^{\mathrm{fine}}) - J(u^{\mathrm{coarse}}) + \tilde{A}(u^{\mathrm{coarse}})(z^{\mathrm{fine}}) - \tilde{F}(z^{\mathrm{fine}})$ 成立。为简洁起见，我们采用记号 $u := u^{\mathrm{fine}}$, $\tilde{u} := u^{\mathrm{coarse}}$ 和 $z := z^{\mathrm{fine}}$。然后，利用分部积分可得

$$\begin{aligned} R &= \int_0^1 \left[\tilde{A}''_{uu}(\tilde{u} + s(u - \tilde{u}))(u - \tilde{u}, u - \tilde{u}, z) - J''_{uu}(\tilde{u} + s(u - \tilde{u}))(u - \tilde{u}, u - \tilde{u}) \right] \cdot s \, \mathrm{d}s \\ &= -\int_0^1 \left[\tilde{A}'_u(\tilde{u} + s(u - \tilde{u}))(u - \tilde{u}, z) - J'_u(\tilde{u} + s(u - \tilde{u}))(u - \tilde{u}) \right] \cdot 1 \, \mathrm{d}s + \left[\tilde{A}'_u(u)(u - \tilde{u}, z) - J'_u(u)(u - \tilde{u}) \right] \cdot 1 - 0. \end{aligned}$$

注意到 $\tilde{A}'_u(u)(u-\tilde{u},\ z) - \tilde{J}'_u(u)(u-\tilde{u}) = 0$，因为 $z := z^{\text{fine}}$ 是精细对偶解˜ ˜问题的解。因此，根据微积分基本定理以及 $A(u)(z) = F(z)$，我们有

$$R = -\left[\tilde{A}(u)(z) - J(u) - \tilde{A}(\tilde{u})(z) + J(\tilde{u})\right] = J(u) - J(\tilde{u}) + \tilde{A}(\tilde{u})(z) - \tilde{F}(z).$$

至此，证明完成。 □

为了使误差估计器可计算，我们忽略二次余项，从而得到与线性问题相同的原始误差估计量（14）。

$$\eta := -\tilde{A}(u^{\text{coarse}})(z^{\text{fine}}) + \tilde{F}(z^{\text{fine}}).$$

与前文类似，我们在误差估计器中用降阶模型的对偶解 z^{coarse}替代全阶对偶解 z^{fine} 。请注意，由于这些近似，对于非线性问题，有效性指数预计不会接近 1。显然，对于高度非线性的问题（例如拟线性或完全非线性问题）以及非线性目标泛函，正如 [28]|[图 4] 和 [27]|[第 6 节5]所示，估计器的两个部分都是必要的。然而，在数值测试中，我们发现估计误差仍能合理地逼近真实误差。

4.2 基于误差估计量的降阶模型更新

本节介绍我们提出的一种新颖的目标导向增量式降阶模型方法。在 MORe DWR 方法中，我们将降阶模型、基于 DWR 的误差估计量以及 POD 算法的增量版本结合起来。MORe DWR 方法针对这样一种情况：降阶模型需要处理在基生成阶段尚未捕捉并纳入的解行为。通常，这会导致全阶解与降阶解之间的误差不断增大。因此，本文提出的方法旨在通过全阶模型或降阶模型，在时间演化过程中检测解行为的变化，或者更准确地说，检测评估所得关注量的差异。当误差增大到难以接受的程度时，该方法便可利用反映新行为的快照，自适应地即时扩充基。因此，可以逐步修改降阶模型，直至误差足够小。

具体而言，我们依托第 3.2 节介绍的时空降阶模型，并应用第 4.1 节关于误差控制的研究成果。与解析误差界相比，采用误差估计具有实际优势，因为它的适用范围更广，即使已知误差界不存在，我们也能使用该方法。此外，为减少计算操作并提高速度，该方法必须采用增量式基生成。增量奇异值分解满足这些要求，只需利用先前的奇异值分解和新的快照即可完成更新。第 4.2.1 节将介绍增量奇异值分解。在此背景下，我们还将介绍作为增量奇异值分解精简版本的增量POD。随后，第 4.2.2 节将阐述完整的 MORe DWR 框架。在此，我们将整合所有组成部分，并介绍最终算法。

总之，我们提出的新方法省去了计算量巨大的离线阶段，直接求解降阶模型。只有在增广基时才需要进行全阶求解，并将其次数降至最低。此外，还可以对关注量的降阶评估进行认证。

4.2.1 增量本征正交分解

本节旨在推导一种算法，根据快照矩阵的修改来更新已有的截断奇异值分解（tSVD），或仅更新其左奇异向量（POD 向量），而无需重新计算整个 tSVD，也无需访问快照矩阵。随后，可通过向快照矩阵追加快照，使用该方法增量式地更新 POD。为此，我们采用加性秩-b 修正这一 SVD 通用方法，该方法主要由 [17, 18]开发，并应用于 [45]中的流体流动模型降阶。尽管该方法支持多种修改，例如调整矩阵尺寸、修改个别数值，或交换行与列，但我们只关注列的更新，即向矩阵添加列，因此将讨论范围限定于此。以下步骤基于 [45][第 2 节。2]

我们从给定的快照矩阵 $Y \in \mathbb{R}^{\#\mathrm{DoFs}(\mathcal{T}_h)\times \tilde{m}}$ 开始，该矩阵包含 $\tilde{m} > 0$ 个快照。通常，$\tilde{m}$等于或与已计算的时间步数相关。此外，我们有矩阵 Y的秩-N截断奇异值分解 USV_{T}。另外，设新计算得到的快照 $b \in \mathbb{N}$ $\{U_1, \ldots, U_b\}$ 存储在批次矩阵中

$$B = \begin{bmatrix} u_1 & \ldots & u_b \end{bmatrix} \in \mathbb{R}^{\#\mathrm{DoFs}(\mathcal{T}_h)\times b}. \tag{17}$$

现在，我们的目标是根据批次矩阵 B 中包含的信息，计算更新后的截断奇异值分解:

$$\tilde{U}\tilde{S}\tilde{V}^T = \tilde{Y} = \begin{bmatrix} Y & B \end{bmatrix}$$

出于性能和内存方面的考虑，我们不显式地重新计算 Y或 Y。这也是 Brand 开展增量奇异值分解研究的初衷，参见 [17, 18]。

因此，我们将列更新写成如下加法形式:

$$\begin{bmatrix} Y & B \end{bmatrix} = \begin{bmatrix} Y & 0_{\#\mathrm{DoFs}(\mathcal{T}_h)\times b} \end{bmatrix} + B \begin{bmatrix} 0_{b\times\tilde{m}} & I_{b\times b} \end{bmatrix} \tag{18}$$

根据 [45] 对奇异值分解应用加性秩-b 修正，从而得到 Y的秩-N 截断奇异值分解 ，其中 $N \leq N + b$ ，并且

$$\tilde{V} = \begin{bmatrix} V & 0 \\ 0 & I \end{bmatrix} V'(:, 1:\tilde{N}) \tag{19}$$

$$\tilde{S} = S'(1:\tilde{N}, 1:\tilde{N}) \tag{20}$$

$$\tilde{U} = \begin{bmatrix} U & Q_{\mathrm{B}} \end{bmatrix} U'(:, 1:\tilde{N}), \tag{21}$$

其中， $F = U'S'V'^T \in \mathbb{R}^{N+b\times N+b}$ 是.....的奇异值分解

$$F = \begin{bmatrix} \Sigma & U^{\mathrm{T}}B \\ 0 & R_{\mathrm{B}} \end{bmatrix} \tag{22}$$

且 $Q_B \in \mathbb{R}^{\#\mathrm{DoFs}(\mathcal{T}_h)\times b}$ 且 $R_B \in \mathbb{R}^{b\times b}$ 由 QR 分解给出

$$Q_{\mathrm{B}}R_{\mathrm{B}} = (I - UU^{\mathrm{T}})B \in \mathbb{R}^{\#\mathrm{DoFs}(\mathcal{T}_h)\times b}. \tag{23}$$

˜对于 POD 基的更新，我们分别将 U 和 U 视为约化基矩阵 Z_N （包括 POD 向量）的更新前版本和更新后版本。我们还忽略了式 (19) 中右奇异向量的更新，因为对于降阶模型而言，除了增加额外的计算工作外，这并不会带来任何额外益处，参见定理 3.1。奇异值用于秩判定，但其获取不会产生任何计算成本。综上，式 (20)–(22) 构成了本文在线计算或增量计算 POD（增量POD）的基础。

补充的技术观察：对于列秩较小的批次矩阵 b，增量POD 算法会频繁调用，而其中可能涉及的代数子空间旋转无法保持正交性，参见 $[18, 32, 4, 31, 80]$。因此，POD 基向量可能出现数值诱发的正交性损失。为解决此问题，建议对 $\begin{bmatrix} U & Q_{\mathrm{B}} \end{bmatrix}$进行额外的正交归一化。算法 2 给出了增量 POD 更新的实现草案。这里， Z_N和 $\Sigma = [\sigma_1, \ldots, \sigma_N] \in \mathbb{R}^N$分别表示式 (6) 中的约化基矩阵及其奇异值。此外，还将式 (17) 中引入的批次矩阵 B （包含 b 个快照）作为输入。约化基所捕获的信息含量由能量阈值 ε决定。

算法 2 增量 POD 更新

输入： 约化基矩阵 $Z_N \in \mathbb{R}^{\#\mathrm{DoFs}(\mathcal{T}_h) \times N}$、奇异值向量 $\Sigma = [\sigma_1, \ldots, \sigma_N] \in \mathbb{R}^N$、批次矩阵 $B \in \mathbb{R}^{\#\mathrm{DoFs}(\mathcal{T}_h) \times b}$以及能量阈值 $\varepsilon \in [0, 1]$。

输出： 约化基矩阵 $Z_N \in \mathbb{R}^{\#\mathrm{DoFs}(\mathcal{T}_h) \times N}$，奇异值向量 $\Sigma = [\sigma_1, \ldots, \sigma_{\tilde{N}}] \in \mathbb{R}^N$

$H = Z_N^T B$

$P = B - Z_N H$

$[Q_P,\, R_P] =$ QR分解(P)

$Q = [Z_N \; Q_P]$

$F = \begin{bmatrix} \mathrm{diag}(\Sigma) & H \\ 0 & R_P \end{bmatrix}$

如果Q 不正交 **则**

    $[Q,\, R] =$QR分解(Q)

    $F = RF$

$[U', \Sigma'] =$ 奇异值分解(F)

$\tilde{N} =$最小化 $\{N \in \mathbb{N} \mid \varepsilon(N) \geq \varepsilon, \;\; 1 \leq N \leq d\}$

$\Sigma =$diag$(\Sigma')(1 : \tilde{N})$

$Z_N = QU'(:, 1 : \tilde{N})$

需要注意的是，检查正交性是否保持可能需要较高的计算成本。因此，我们采用一种启发式方法，仅验证矩阵的第一列与最后一列是否正交。

4.2.2 目标导向认证的增量式降阶模型

本节将第 3.2 节介绍的时空降阶模型和第 4.2.1 节的增量POD，与第 4.1 节关于目标导向误差控制的研究结果结合起来。由此得到一种自适应目标导向增量降阶模型。首先，在时间片定义之外，我们引入

父时间片概念，将其作为对时空域的进一步划分。父时间片将多个在时间上连续的时间片统一起来，其定义如下：

$$P_k^r = \{S_l^n \mid l \geq k \ \wedge \ n \leq r\}.$$

我们的方法无需预先了解或探索解流形，同时力求尽量减少全阶运算。因此，我们旨在按父时间片求解降阶模型，并在必要时借助增量POD，用父时间片的全阶解自适应地增广降阶基，直至降阶基足以满足针对所选代价泛函给定的估计误差容限。为此，我们将DWR方法中引入的精细解和粗略解分别对应于有限元解和降阶基解，并估计父时间片中每个时间片上的误差。用于增广基的全阶解在误差最大的时间片上计算。需要指出的是，原始和对偶全阶解均在该时间片上计算，并分别用于增广原始基和对偶基。因此，每次增广都要进行两次全阶求解。在完成一个父时间片上的这一迭代过程后，将得到的基传递到后续父时间片，并以此作为求解降阶模型的起始基，重复整个过程。因此，如果下一个父时间片上的解行为与已观察到的行为仅略有不同，现有降阶基应能重现其中的大部分行为。这样，只需少量基更新，便有望快速计算降阶解。然而，如果解行为发生显著变化，误差估计将检测到这一点，并进一步细化基，以确保解满足误差容限。我们发现，该过程与Meyer和Matthies在 [53] 中提出的基于DWR估计的自适应基选择完全兼容，后者旨在降低降阶空间的维数。因此，若将其纳入该方法，便可根据问题设定对降阶基进行增广或缩减。

由此得到的方法概述于算法3中。为简单起见，我们将时空柱分解为 K 个长度固定的父时间片 L，并按时间顺序为其编号，即P_1，P_2，……，P_K。为标明某个时间片隶属于哪个父时间片 P_k，其中包含的时间片记为 $S_{P_k}^1$，$S_{P_k}^2$，……，$S_{P_k}^L$，其中 $1 \leq k \leq K$。每个时间片 $S_{P_k}^j$对应的离散原始系统分别由式（3）和式（8）给出，对应完整模型和降阶模型。对于对偶问题

$$A' Z_{S_{P_k}^l} = J_{S_{P_k}^l} \quad \text{and} \tag{24}$$

$$A'_N Z_{N,S_{P_k}^l} = J_{N,S_{P_k}^l} \tag{25}$$

表示伴随问题 (12) 的离散化全阶系统和降阶系统。此外，在时间片 $S_{P_k}^l$ 上的误差估计器 (15) 评估结果为 $\eta_{N,\ S_{P_k}^l}\left(U_{N,S_{P_k}^l}, Z_{N,S_{P_k}^l}\right)$。注意，采用原始和对偶降阶解，可使评估不依赖全阶系统的规模，从而快速评估误差。最后，增量POD (2) 简称为 iPOD，其输入为降阶基、打包在快照矩阵中的新快照以及奇异值，输出为新的 POD 基。

算法3 增量式降阶模型

输入： 初始条件 $U_0 := U(t_0)$、原始和对偶降阶基矩阵 Z_N^p 和 Z_N^d、能量阈值 $\varepsilon \in [0,1]$ 以及误差容限 tol > 0。

输出： 原始和对偶降阶基矩阵 Z_N^p 和Z_N^d，以及对所有 $1 \le m \le M$ 的降阶原始解 $U_{N,\ I_m}$。

1：对于 $k = 1, 2, \ldots, K$，执行 2：当 $\eta_{max} > tol$ 时，执行 3：对于 $l = 1, 2, \ldots, L$，执行 4：求解降阶原始系统 (8)：$A_N U_{N,S_{P_k}^l} = F_{N,S_{P_k}^l}$ 5：对于 $l = L, L-1, \ldots, 1$，执行 6：求解降阶对偶系统 (25)：$A'_N Z_{N,S_{P_k}^l} = J_{N,S_{P_k}^l}$ 7：对于 $l = 1, 2, \ldots, L$，执行 8：计算误差估计：$\eta_{N,S_{P_k}^l}\left(U_{N,S_{P_k}^l}, Z_{N,S_{P_k}^l}\right)$ 9：$\eta_{max} = \max\limits_{1\le l\le L} \left|\eta_{N,S_{P_k}^l}\right|$ 10：如果 $\eta_{max} > tol$，则 11：$l_{max} = \arg\max\limits_{1\le l\le L} \left|\eta_{N,S_{P_k}^l}\right|$ 12：求解原始全阶系统 (3)：$AU_{S_{P_k}^{l_{max}}} = F_{S_{P_k}^{l_{max}}}$ 13：更新原始降阶基：$Z_N^p = \text{iPOD}(Z_N^p, [U_{S_{P_k}^{l_{max}}}(t_1) \ldots, U_{S_{P_k}^{l_{max}}}(t_{r+1})], \Sigma)$ 14：求解对偶全阶系统 (24)：$A' Z_{S_{P_k}^{l_{max}}} = J_{S_{P_k}^{l_{max}}}$ 15：更新对偶降阶基：$Z_N^d = \text{iPOD}(Z_N^d, [Z_{S_{P_k}^{l_{max}}}(t_1) \ldots, Z_{S_{P_k}^{l_{max}}}(t_{r+1})], \Sigma)$ 16：更新降阶系统分量和误差估计器（相对于 (9)） 17：———— 验证循环 ————▷ 这是模型的一种可选验证机制。18：对于 $k = 1, 2, \ldots, K$，执行 19：对于 $l = 1, 2, \ldots, L$，执行 20：求解原始降阶系统：$A_N U_{N,S_{P_k}^l} = F_{N,S_{P_k}^l}$ 21：对于 $k = K, K-1, \ldots, 1$，执行 22：对于 $l = L, L-1, \ldots, 1$，执行 23：求解降阶对偶系统：$A_N^\star Z_{N,S_{P_k}^l} = J_{N,S_{P_k}^l}$ 24：对于 $k = 1, 2, \ldots, K$，执行 25：对于 $l = 1, 2, \ldots, L$，执行 26：计算时间片估计：$\eta_{N,S_{P_k}^l}(U_{N,S_{P_k}^l}, Z_{N,S_{P_k}^l})$

除了前面提到的步骤外，我们还加入了一个可选的验证循环，其目的取决于具体应用。具体而言，该循环使用最终的降阶基重新计算所有降阶解，并再次评估其误差。如果生成的降阶基旨在重复使用，

则对其精度进行额外验证，可确保该降阶基适用于整个时间域内的解近似。这主要适用于优化过程，或使用 MORe DWR 方法进行流形探索的情况。不过，如果唯一目的只是一次性评估某个关注量，出于性能考虑，可以省略验证。

此外，我们注意到，与（4）中的 POD 纯近似误差类似，该误差估计的物理解释并不直观。因此，我们考虑对近似质量进行相对测量。然而，全阶解无法用于误差归一化，因此我们转而采用

$$J\left(U_{S^l_{P_k}}\right) \approx J\left(U_{N,S^l_{P_k}}\right) + \eta_{N,S^l_{P_k}}.$$

由此得到时间片 $\eta^{rel}_{N}{}_{,S^l_{P_k}}$ 上的相对误差估计器 $S^l_{P_k}$，其定义为

$$\eta^{rel}_{N,S^l_{P_k}} = \frac{\eta_{N,S^l_{P_k}}}{J\left(u_{S^l_{P_k}}\right)} \approx \frac{\eta_{N,S^l_{P_k}}}{J\left(u_{N,S^l_{P_k}}\right) + \eta_{N,S^l_{P_k}}}. \tag{26}$$

5 数值测试

为了展示我们的方法，我们针对三种不同的问题配置进行了数值测试。在前两次数值测试中，我们分别对 $1{+}1$D 和 $2{+}1$D 热方程进行计算。对于前者，我们采用线性目标泛函；对于后者，我们采用非线性目标泛函。为了展示时间离散的灵活性，我们对热方程在时间上使用高斯-勒让德求积点，并采用 dG(1) 时间离散。作为第三种问题配置，我们考虑一个 $3{+}1$D 悬臂梁，将其作为弹性动力学中的基准问题。对于该问题，我们在时间上使用高斯-洛巴托求积点，这些求积点是传统时间步进格式的支撑点，并采用 dG(2) 时间离散。

所有计算均在一台配备 Intel i5-7600K CPU @ 3.80GHz $\times$ 4 和 16GB 内存的个人计算机上完成。时空有限元法代码使用 deal.II [2, 3] 编写，降阶建模则使用 NumPy [39]和 SciPy [74]完成。代码之间通过硬盘交换数据。

5.1 $1{+}1$D 热方程

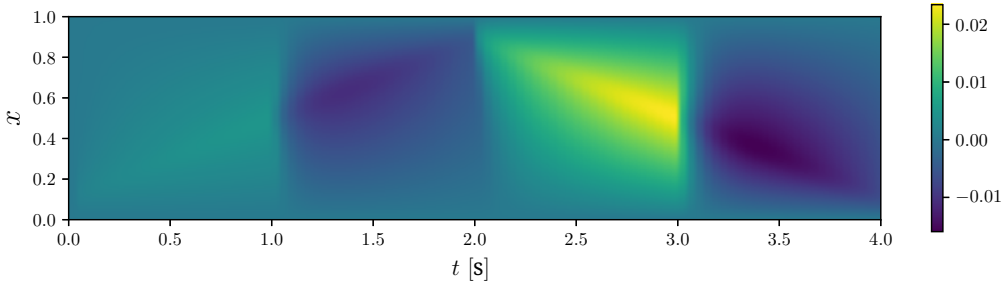
对于第一个数值测试，我们构造一个 $1{+}1$D 热方程问题；见公式 2.1。我们考虑空间域 $\Omega = (0,1)$ 和时间域 $I = (0,4)$。我们采用一个移动热源，每秒改变一次温度，并以 0.1 的加热区间宽度在空间域内从 $x = 0.1$ 移动到 $x = 0.9$，然后返回 $x = 0.1$。为此，我们使用右端函数

$$f(t,x) := \begin{cases} 0.2 & t \in (0,1),\ -0.05 \le x - 0.4t - 0.1 \le 0.05, \\ -0.5 & t \in (1,2),\ -0.05 \le x - 0.4t - 0.1 \le 0.05, \\ 1.0 & t \in (2,3),\ -0.05 \le x + 0.4(t-2) - 0.9 \le 0.05, \\ -0.75 & t \in (3,4),\ -0.05 \le x + 0.4(t-2) - 0.9 \le 0.05. \end{cases}$$

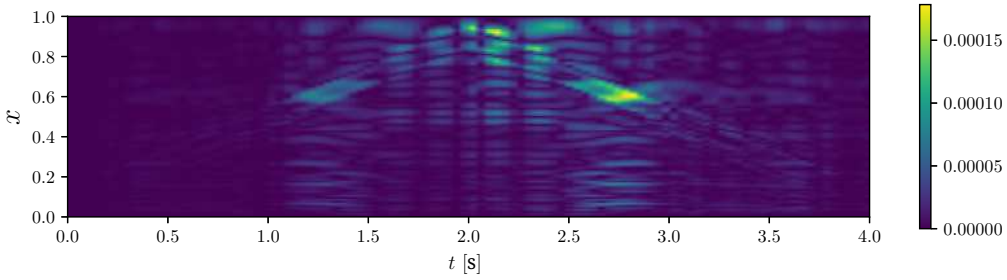
我们采用零初始条件、齐次狄利克雷边界条件，以及时间平均均值目标泛函 $J(u) := \frac{1}{4}\int_0^4 \int_0^{\frac{1}{2}} u(t,x)\,\mathrm{d}x\,\mathrm{d}t$。需要指出的是，目标泛函的支撑并未覆盖整个空间域，而仅位于其下半部分 $(0, 1\,2) \subsetneq \Omega$。

对于降阶模型，我们要求原始约化基和对偶约化基必须保留$\varepsilon = 1 - 10^{-8}$ 的信息。如前所述，我们采用在（26）中提出的相对误差估计 η_N^{rel}，$S_{P_k}^l$，并允许误差达到 1% 的容差。对于许多应用而言，我们认为这是一个合理的容差。全阶模型在空间和时间上分别具有 $n = 8$，193 个和 $q = 10$，240 个自由度。这使我们总共有 $n \cdot q = 83$，896，320 个时空自由度。此外，时间域被划分为 $M = 5$，120 个时间片。对于增量式降阶模型，我们选择总共 $K = 64$ 个父时间片，并将各时间片均匀分配，即$L = 80$。

在图 2 中，我们展示了全阶时空解 u_h 以及全阶时空解 u_h 与使用 MORe DWR 获得的降阶时空解 u_N 之间的真实误差 $u_h - u_N$。观察误差可以发现，与 $x \in (0, 1\,2)$（即目标泛函的空间支集）相比，降阶模型在 $x \in (\frac{1}{2}, 1)$ 处的精度较低。这表明我们的增量POD具有目标导向性。



(a) 时空全阶解 u_h



(b) 时空真实误差 $u_h - u_N$

图 2： 1+1D 热方程的时空解和误差。

在图 3 中，我们比较了全阶时空解 u_h 和降阶时空解 u_N在各时间片上的目标泛函时间轨迹。可以看出，两条轨迹无法区分，这表明即使解的行为发生变化，降阶模型仍能准确捕捉关注量的时间演化。在考察时间平均代价泛函时，也能观察到良好的近似质量。我们得到 $J(u_h) = 2.0608 \cdot 10^{-4}$和 $J(u_N) = 2.0583 \cdot 10^{-4}$，相对误差为

$\eta_{max} = 0.1210\%$。

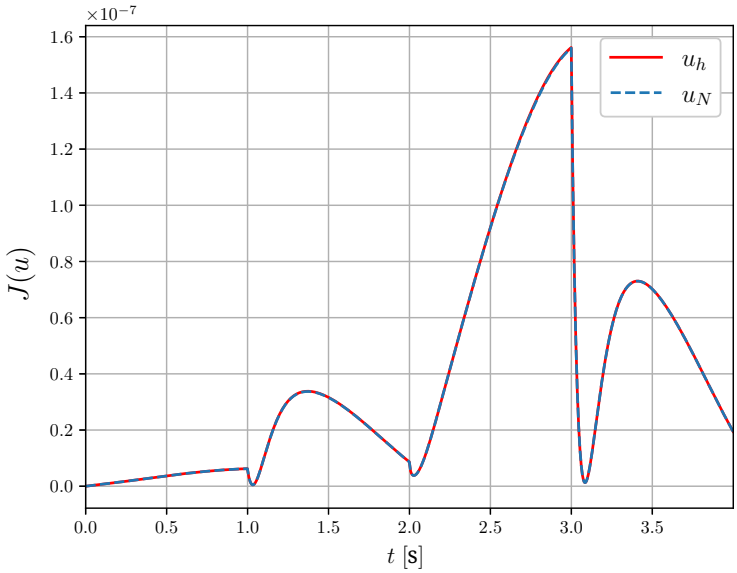


图 3： 1+1D 热方程的代价泛函随时间的演化。

我们在图 4 中比较了各时间片上的时间误差估计与精确时间误差。两条曲线的总体趋势相似。精确误差的平均值比 1% 的误差容限小一个数量级以上（图中以绿色虚线表示）。在 $t = 3\,s$之后的一小段时间内，误差估计超过了容限。这种高估可能导致执行不必要的全阶求解。尽管如此，高估对近似质量的影响较小，因为精确误差仍满足容限。

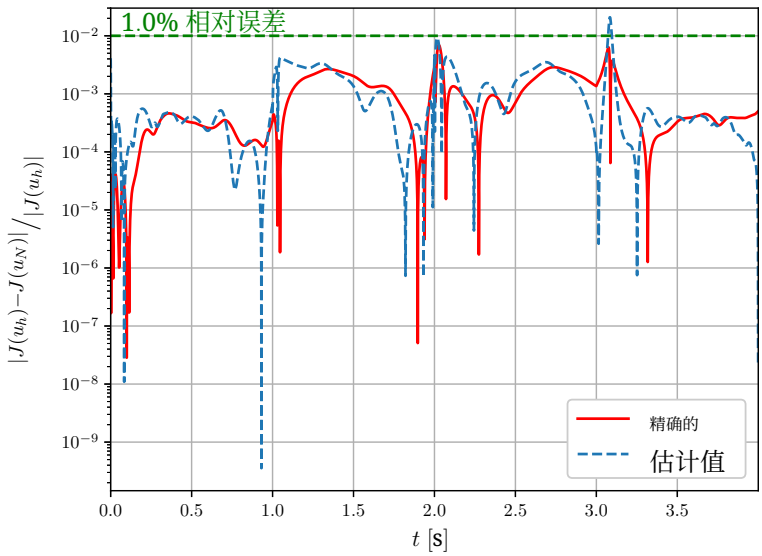


图 4： 1+1D 热方程的逐时间区间误差估计器与真实误差的比较。

表 1 概述了误差容限介于 0.1% 和 10% 之间时的仿真结果。列出的指标包括：相对误差、计算加速比、全阶模型求解次数、原始问题和对偶问题的 POD 基底维数、误差估计器的预测能力，以及式 (16) 中的有效性指数。这里，全阶模型求解次数是原始问题与对偶问题求解次数之和，基底维数按原始/对偶的顺序列出。误差估计器的预测能力通过混淆矩阵表示。每个时间片的预测结果归入以下四种情况之一：

$$\begin{array}{l|l|} \text{error} > \text{tol} \wedge \text{estimate} < \text{tol} & \text{error} < \text{tol} \wedge \text{estimate} > \text{tol} \\ \text{error} > \text{tol} \wedge \text{estimate} > \text{tol} & \text{error} < \text{tol} \wedge \text{estimate} < \text{tol}. \end{array}$$

我们注意到，这四种可能情形按照其发生后果的严重程度排序。因此，前两种情况表示估计器出现了错误预测。第一种情况最不可取，因为此时误差估计器低估了真实误差，可能导致降阶基规模不足。第二种情况的危害较小，因为此时误差估计器高估了真实误差，可能使降阶基略大于所需规模。最后两种情况危害较小，因为估计值能够正确预测误差。不过，第三种情况也并非最优，因为它表明，在验证循环中完成增量基扩充后，仍有一些时间片超出了误差容限。因此，我们预计，对于高效方法，几乎所有时间片都应落入最后一类：既满足误差容限，误差估计值也低于该容限。

我们观察到，随着容差增大，相对误差和加速比也随之增大。请注意，相对误差几乎比容差小一个数量级，这与图4的结果一致。数量级上的差异可以解释为：容差必须在每个时间片上分别满足，而相对误差则是在整个时间域上评估的。加速比的提升源于：对于给定的容差，原始问题和对偶问题的全阶模型求解次数减少，且所需的POD基大小更小。此外，在约 $98-99\%$ 的情况下，估计器能够正确预测误差与容差之间的关系，而大多数错误预测都是高估。同样地，对于有效性指数，随着容差增大，可以看到其略有下降，因为此时在误差估计器中以降阶对偶解替代全阶对偶解会引入额外误差。

容差	相对误差	加速比	全阶模型求解次数	基大小	预测	有效性
0.1%	0.0130%	10.9	68	39 | 36	0 | 38 | 0 | 5082	1.0065
1%	0。1210%	12.2	40	25 | 22	0 | 31 | 0 | 5089	1.0071
2%	0.3370%	13.2	38	24 | 21	0 | 41 | 0 | 5079	1.0074
5%	1.2019年%	15.2	32	21 | 18	28 | 48 | 18 | 5026	1.0125
10%	1。7645%	18.6	30	20 | 17	0 | 73 | 0 | 5047	1.0404

表 1：MORe DWR 在目标泛函容差不同的情况下求解 $1+1$D 热方程的性能。

最后，我们在图5中展示了 MORe DWR 方法的增量特性。在这

一背景下，我们通过绘制时间域内原始降阶基和对偶降阶基的大小，来说明在线基生成，并比较其在1%和10%容差下的变化。结果表明，容差越小，原始基和对偶基大小的增长越陡峭且越细致。不过，我们观察到，在约2秒之后，所有基在各种容差下的基大小都保持稳定。如果将热源的移动考虑在内，这恰好是热源穿过一次空间域所需的时间。因此，此后系统中不再加入会触发进一步基增广的新信息。

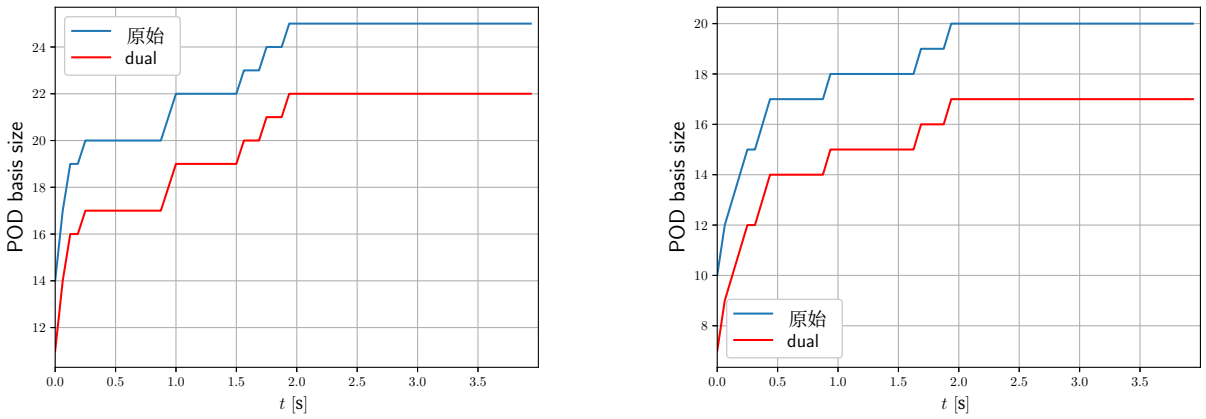


图5： 1+1D 热方程在相对误差容限为 1%（左）和 10%（右）时，降阶基大小随时间的演化。

5.2 2+1D 热方程

在第二个数值实验中，我们在一个 2+1D 热方程问题上测试 MORe DWR。我们考虑空间域 $\Omega = (0,\ 1)^2$ 和时间域 $I = (0,\ 10)$。我们构造一个振荡温度的移动热源，使其围绕空间域 Ω 的中点旋转，如图6所示。为此，我们使用右端函数

$$f(t,x) := \begin{cases} \sin(4\pi t) & \text{if } (x_1 - p_1)^2 + (x_2 - p_2)^2 < r^2, \\ 0 & \text{else,} \end{cases}$$

其中 $x = (x_1, x_2)$、中点 $p = (p_1, p_2) = (\frac{1}{2} + \frac{1}{4}\cos(2\pi t)\ \frac{1}{2} + \frac{1}{4}\sin(2\pi t))$ 以及轨迹半径 $r = 0.125$。此外，采用零初始条件和齐次狄利克雷边界条件。与第5.1节中的目标泛函不同，我们针对非线性代价泛函 $J(u) := \frac{1}{10}\int_0^{10}\int_\Omega u(t,x)^2\,\mathrm{d}x\,\mathrm{d}t$ 测试该方法。

—

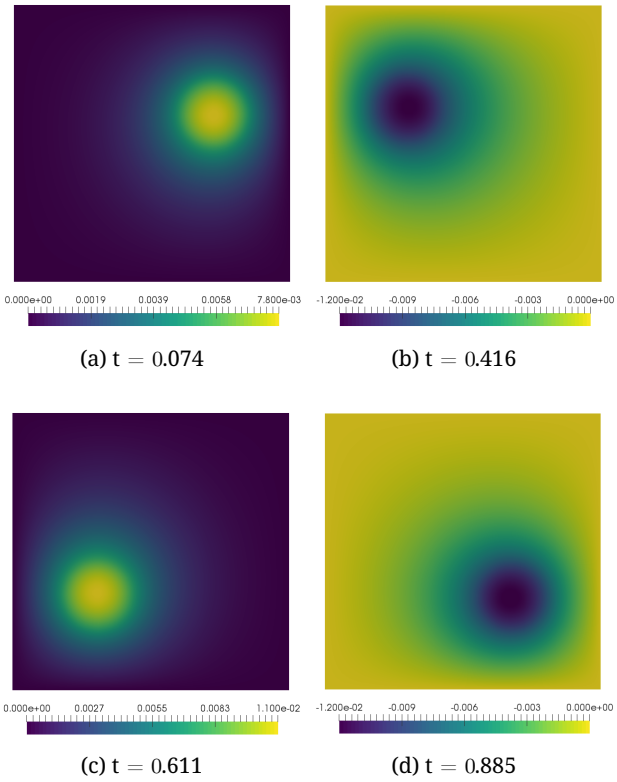


图6： 2+1D 热方程的全阶解快照。

对于降阶模型，我们要求原始与对偶约化基保留$\varepsilon = 1 - 10^{-8}$ 的信息。与之前的一维情形类似，我们采用相对误差估计 η_N^{rel} ,$S_{P_k}^l$ ，并允许误差容差最高为 1%。全阶模型在空间和时间上分别具有 $n = 4{,}225$ 和 $q = 4{,}096$ 个自由度。因此，总计有 $n \cdot q = 17{,}305{,}600$ 个时空自由度。此外，时间域被划分为$M = 2{,}048$ 个时间片。对于增量式降阶模型，我们选择总计 $K = 128$ 个父时间片，并将时间片均匀分布其上，即 $L = 16$。

首先，在图 7 中，我们比较了全阶时空解 u_h 和降阶时空解 u_N在每个时间片上的目标泛函时间轨迹。结果表明，尽管解的行为持续变化，两条轨迹仍难以区分。此外，从时间平均代价泛函来看，也能观察到良好的近似质量。我们得到 $J(u_h) = 6.4578 \cdot 10^{-5}$和 $J(u_N) = 6.4577 \cdot 10^{-5}$，相对误差为 $\eta_{max} = 0.0016\%$。这表明，增量式降阶模型能够在给定容差内复现非线性代价泛函。

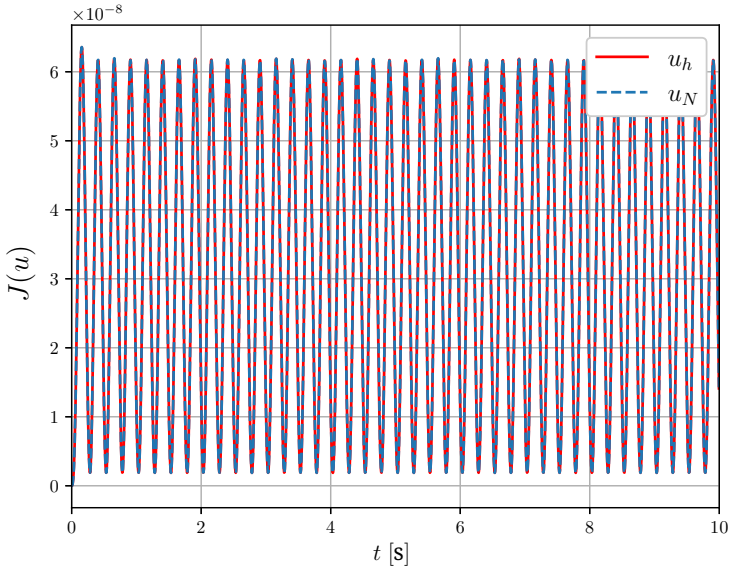


图 7： 2+1D 热方程代价泛函的时间演化。

图 8 对比了每个时间片上的精确时间误差及其估计值。此外，为便于说明，图中标出了 1% 的误差容限。结果表明，在所有时间片上，精确误差和估计误差都满足给定的误差容限。总体而言，估计值呈现出与精确误差相似的轨迹。不过，我们可以观察到精确误差中的尖峰并未被估计值完全覆盖。尽管如此，这些偏差并未造成影响。

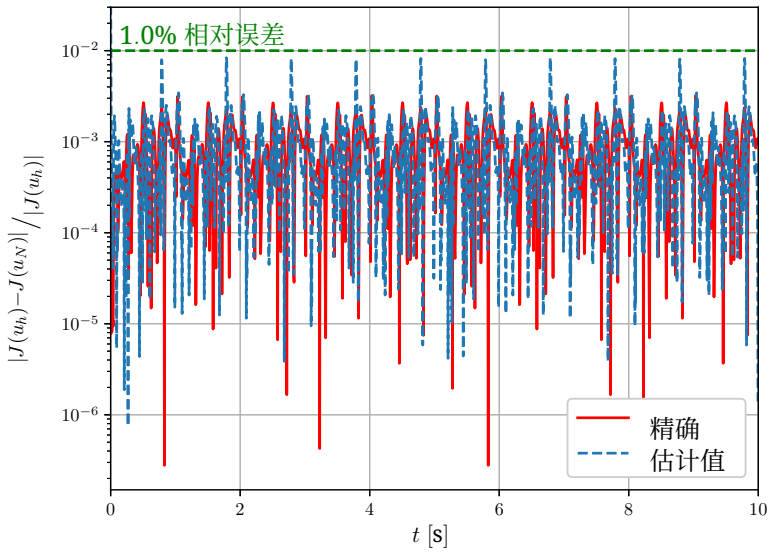


图 8： 2+1D 热方程的时间区间逐段相对误差估计器与真实误差的时间演化。

表 2 展示了不同误差容限下的仿真结果。我们考察的量包括：相对误差、计算加速比、全阶模型求解总次数、原始问题和对偶问题的 POD 基维数、误差估计器的预测能力以及有效性

指数。有关这些量的定义，请参见第5.1节。可以看出，随着容差增大，相对误差和加速比也随之增大。相对误差仍远小于容差。与一维线性情形相比，放宽误差容限对加速比的影响更大。这可以从全阶模型求解次数和 POD 基维数随给定容差的变化中得到解释。此外，在约 $94-99\%$ 的情况下，估计器能够正确预测误差与容差之间的关系，而错误预测大多是高估。一个例外是 tol $=10\%$ 的情况，其预测能力下降了 5%，这表明对偶基过小，无法准确估计误差。调整对偶基信息含量的容差或许可以解决这一问题。尽管如此，所得降阶代价泛函仍满足误差容限。与线性情形相比，最大的差异在于有效性指数的评估。我们发现，与线性情形相比，有效性指数在 1 附近波动更大，这符合非线性代价泛函下的预期。不过，有效性指数仍处于可接受范围内，结果良好。最后，我们发现，当容差较大，为 10% 时，会出现少量错误预测：79 个时间片上，真实误差大于容差而估计误差小于容差；28 个时间片上，误差估计器给出的误差大于容差而真实误差小于容差。此外，在该容差下，还有 17 个时间片的真实误差和估计误差均大于容差。MORe DWR 性能的这种下降，可以用 DWR 误差估计器中以粗略对偶解 z^{coarse} 替代精细对偶解 z^{fine} 来解释。如果原始问题和对偶问题的 POD 基过小，这种近似可能会引入额外误差，导致我们的方法性能变差。

容差	相对误差	加速比	全阶模型求解次数	基大小	预测	有效性
0.1%	0.0019%	7.7	150	$92 \mid 78$	$0 \mid 35 \mid 0 \mid 2013$	0.7524
1%	0.0017%	27.5	80	$55 \mid 44$	$0 \mid 1 \mid 0 \mid 2047$	0.2771
2%	0.0628%	29.6	66	$47 \mid 36$	$0 \mid 9 \mid 0 \mid 2039$	3.9181
5%	0.9162%	44.8	44	$33 \mid 25$	$0 \mid 1 \mid 0 \mid 2047$	1.2254
10%	0.9243%	50.0	38	$31 \mid 23$	$79 \mid 28 \mid 17 \mid 1924$	1.5474

表 2：针对目标泛函容差的 $2{+}1$D 热方程增量降阶建模总结。

最后，图9展示了 MORe DWR 方法的增量特性。通过绘制原始降阶基和对偶降阶基在时间域内的大小，并比较其在 1% 和 10% 容差下的变化，展示了在线基生成。结果表明，在仿真的第一秒内，原始基和对偶基的大小都急剧增加，这反映了振荡热源在空间域内完成了一次往返。同样，容差越严格，降阶基大小增长得越快。热源完成第一次往返后，基大小几乎保持不变；仅在约 $t=4$ s 处，容差为 1% 时发生了一次基增广。这是因为所选数值实验具有周期性行为，不会向系统提供更多信息。因此，为满足给定的误差容限，无须再进行基增广，或只需进行少量增广。

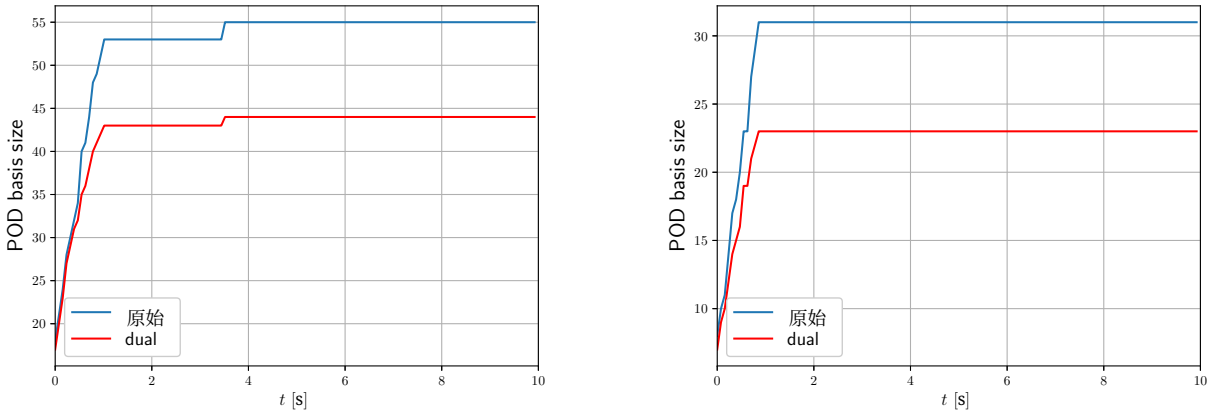


图9：对于 2+1维热方程，当相对误差容限为1%（左）和10%（右）时，降阶基大小的时间演化。

5.3 3+1维弹性动力学方程

在第三次数值实验中，我们选择公式 2.2，并在 a 3+1D 弹性动力学问题上考察该方法。我们考虑一个跨越空间域 $\Omega = (0, 6) \times (0, 1) \times (0, 1)$ 的矩形梁。此外，还考虑时间域 $I = (0, 40)$。通过定义作用于梁上边界 $\Gamma_{\text{up}} = (0, 6) \times (0, 1) \times \{x_3 = 1\}$ 上的力 $f(t, x)$，使梁沿竖直方向振荡。在实验的第一阶段，如图10所示，作用力将梁抬起。随后，逐渐撤去该力，使梁开始摆动。

为此，我们使用

$$g(t) := \begin{cases} f_{\max} \frac{t}{t_1} & x_3 = 1 \ \wedge \ t \leq t_1, \\ f_{\max} \left(1 - \frac{t - t_1}{t_2 - t_1}\right) & x_3 = 1 \ \wedge \ t_1 < t \leq t_2, \\ 0 & \text{else}, \end{cases}$$

其中最大作用力为 $f_{\max} = 0.5$ ， $t_1 = 5$ 和 $t_2 = 6$ 分别为作用力增大或减小的时间点。结合梁在边界处固定 $\Gamma_{\text{clamped}} = \{x_1 = 0\} \times (0, 1) \times (0, 1)$，可得第 2.1.2 节中的边界条件

$$\begin{aligned} u &= 0 && \text{in } I \times \Gamma_{\text{clamped}}, \\ v &= 0 && \text{in } I \times \Gamma_{\text{clamped}}, \\ \sigma(u) \cdot n &= 0 && \text{in } I \times \partial\Omega \setminus (\Gamma_{\text{clamped}} \cup \Gamma_{\text{up}}), \\ \sigma(u) \cdot n &= g(t) && \text{in } I \times \Gamma_{up}. \end{aligned}$$

此外，齐次初始条件为

$$\begin{aligned} u(0) &= 0 && \text{in } \Omega, \\ v(0) &= 0 && \text{in } \Omega. \end{aligned}$$

我们选择作用于夹持边界 Γ_{clamped} （记为 $J(u) := \frac{1}{40}\int_0^{40}\int_\Omega \sigma(u(t,x)) \cdot n \,\mathrm{d}x\,\mathrm{d}t$ ）的时间平均应力作为代价泛函。对于降阶模型，我们决定原始和对偶约化基必须保留 $\varepsilon = 1 - 10^{-11}$ 的信息。我们再次采用相对误差估计 η_N^{rel} ，并将误差容差设为 1%。全阶模型在空间和时间上分别具有 $n = 702$ 和 $q = 4{,}800$ 个自由度。因此，总共有 $n \cdot q = 561{,}600$ 个时空自由度。此外，时间域被划分为 $M = 1{,}600$ 个时间片。对于增量式降阶模型，我们选择总计 $K = 80$ 个父时间片，并将时间片均匀分布在这些父时间片上，即 $L = 20$。

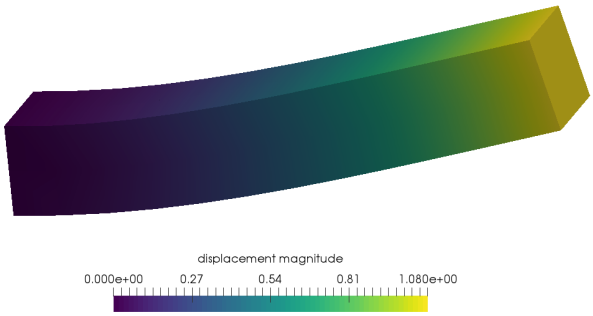


图10：弹性动力学方程在 t = 5.75 时的全阶解快照。

图11比较了全阶时空解 u_h 和降阶时空解 u_N 在每个时间片上的目标泛函时间轨迹。结果表明，两条轨迹无法区分，降阶代价泛函能够模拟其振荡行为。此外，从时间平均代价泛函也可以观察到良好的近似质量。我们得到 $J(u_h) = -3.1114$ 和 $J(u_N) = -3.1115$，相对误差为 $\eta_{max} = 0.0035\%$，小于所需的 1% 误差容限。

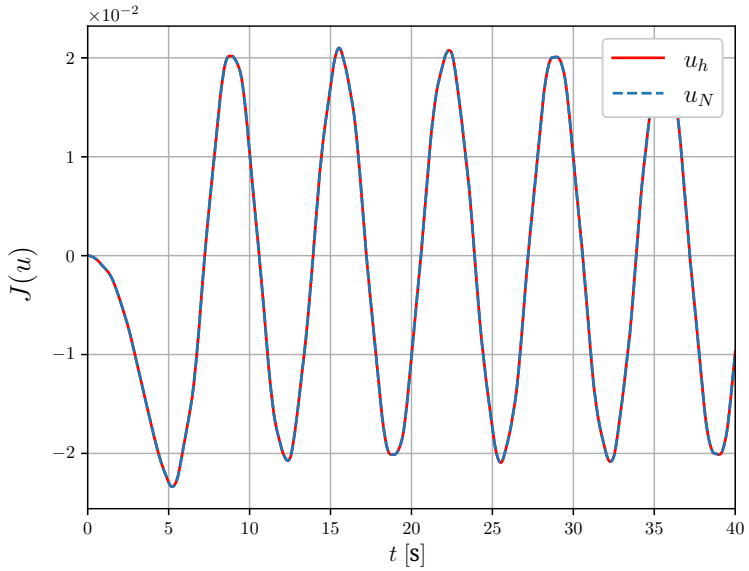


图11： 3+1D 弹性动力学方程的目标泛函的时间演化。

在图12中，我们绘出了每个时间片上的精确时间误差及其估计值，以供比较。为便于说明，图中标出了 1% 的误差容限。结果表明，这两个量的平均量级相同，而实际误差的标准差似乎更大。因此，误差中存在未被误差估计捕捉到的尖峰。大多数时候，这不会造成影响，但有一个时间片上的误差略微超过了误差容限。

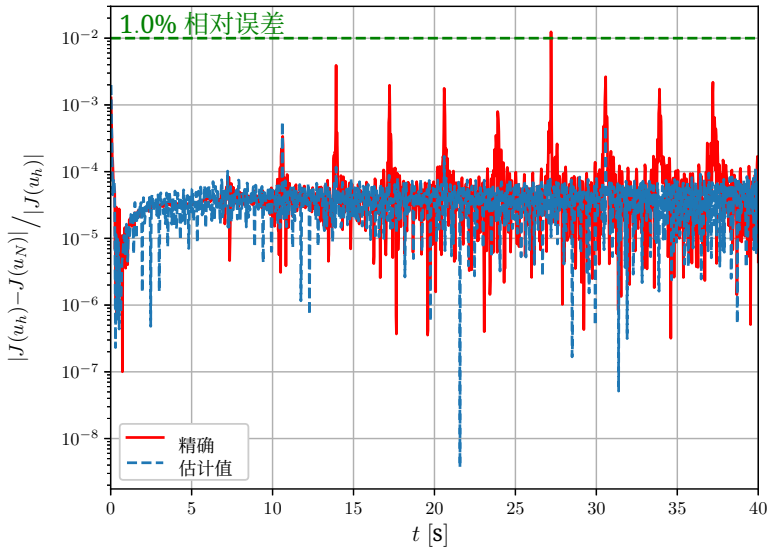


图12： 3+1D 弹性动力学方程的时间区间逐段相对误差估计器与真实误差的时间演化。

为了探究误差容限被违反的情况，我们在表3中给出了不同误差容限下的仿真结果。因此，我们展示了相对误差、预测指标和效能指数，以考察误差估计。此外，还列出了计算加速比、全阶模型求解次数总计，以及原始问题和对偶问题的 POD 基底维数。有关这些量的定义，请参阅第5.1节。可以看出，尽管大多数错误预测都是对误差的严重低估，但这类情况并不多。此外，效能指数接近最优值 1，且所有情况下的相对误差均满足容限。不过，对于容限较大的降阶模型，效能指数略有下降。另外，回顾性能测量结果时，我们发现其与之前的热问题有所不同。所有容限下的加速比和全阶模型求解次数都基本保持不变。只有在容限为 10% 时，加速比和基大小才有进一步改善。我们还观察到，与热方程情形不同，全阶模型求解次数总计和 POD 基的大小不会随误差容限增大而单调递减。这种现象可能与误差本身的变化有关。采用较小的容限可能会导致较早进行更多降阶基扩充。较大的容限会使初始降阶基较小，因此之后会因基大小不足而产生更大误差；这一问题会在后续阶段通过扩充基来弥补。

容差	相对误差	加速比	全阶模型求解次数	基大小	预测	有效性
0.1%	0。0042%	11.2	44	98 | 79	26 | 0 | 1 | 1573	1.0029
1%	0.0035%	12.1	46	107 | 82	1 | 0 | 0 | 1599	1.0001
2%	0.0001%	10.6	46	113 | 82	0 | 0 | 0 | 1600	0.9953
5%	0.0040%	12.6	48	101 | 86	0 | 0 | 0 | 1600	0.9999
10%	0.0200%	14.9	38	89 | 84	44 | 1 | 0 | 1555	0.9801

表3：取决于目标泛函容差的 3+1D弹性动力学方程增量降阶建模总结。

最后，图13展示了我们的 MORe DWR 方法的增量特性。我们通过绘制原始降阶基和对偶降阶基在时间域内的大小，并比较容限为 1% 和 10% 时的变化情况，来展现在线基生成过程。与之前的情形类似，我们观察到，在仿真开始时，原始基和对偶基的大小都迅速增加。此外，仿真后半段降阶基的大小也发生了变化。我们再次看到，较严格的容限会产生更大的原始降阶基，也就是说，降阶基经过了更多次细化。这次数值实验的另一个有趣之处是，当容限为 1% 时，由于执行了一次增量POD更新，纳入了一个包含大量信息的快照，对偶降阶基在 $t \approx 10$ s 时略有缩小。

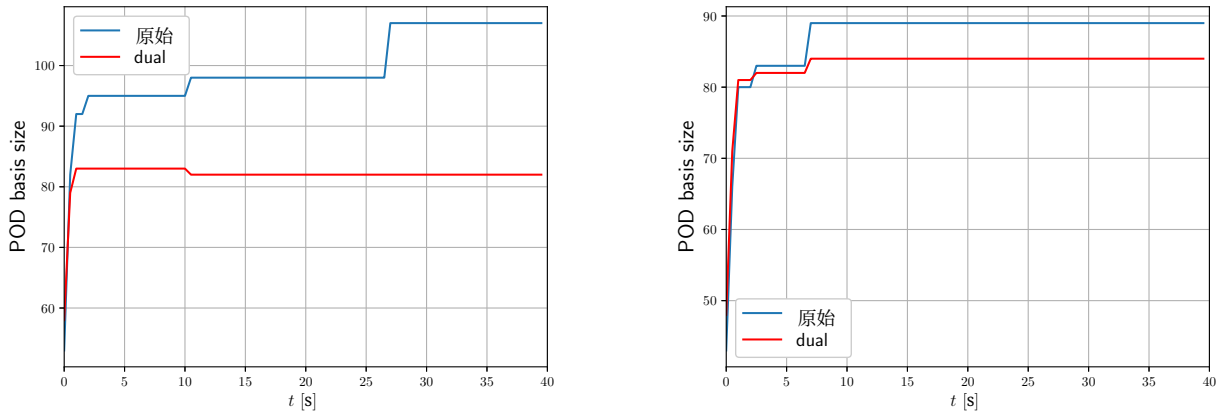


图13： 3+1D弹性动力学方程在相对误差容限为1%（左）和10%（右）时降阶基大小的时间演化。

6 结论与展望

本文针对线性偏微分方程（即热方程和弹性动力学方程）以及线性与非线性目标泛函，提出了一种基于时空 DWR 误差估计、能够即时增广基的增量式 POD-ROM 方法。第5节已证明该方法效率良好，因此可应用于广泛的问题。对于线性问题，有效性指数几乎恰好为1，这使误差估计在实践中可靠；对于非线性目标泛函，有效性指数也较好。对于非线性偏微分方程和目标泛函，可能需要完整的 DWR。

此外，加速比最高可达50，同时全阶模型与降阶模型解之间的误差仍保持在预设容差范围内。因此，例如，降阶基方法离线阶段耗费高昂的高保真计算可由我们的增量POD方法取代。未来值得探索的一个方向是将该方法扩展至动态自适应空间网格，以进一步加快计算。

致谢

作者感谢德国研究联合会（DFG）通过其计算力学高维技术国际研究生院框架提供资助，项目编号为GRK 2657，资助编号为 433082294。此外，我们感谢 Hendrik Geisler （汉诺威莱布尼茨大学，GRK 2657）进行富有成效的讨论并提出意见。我们也感谢法德大学通过法德博士学院“先进数值与测试方法”（CDFA-DFDK 19-04）提供支持。

弹性动力学的时空线性系统与 dG(r) 时间步进形式

本附录讨论了在一个时间片上采用单个时间单元，并在时间方向采用 dG(r) 离散化的弹性动力学方程的时空离散。利用弹性动力学的全离散变分形式 2.4，可得线性方程组

$$\left[C_k \otimes M_h + M_k \otimes K_h + D_k^1 \otimes M_h\right] U_m = F_m + \left[D_k^2 \otimes M_h\right] U_{m-1} \tag{27}$$

其中，空间矩阵为

$$\begin{aligned} M_h &= \left\{ (\varphi_h^{v,(j)}, \varphi_h^{u,(i)}) + (\varphi_h^{u,(j)}, \varphi_h^{v,(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)}, \\ K_h &= \left\{ (\sigma(\varphi_h^{u,(j)}), \nabla_x \varphi_h^{u,(i)}) + (\varphi_h^{v,(j)}, \varphi_h^{v,(i)}) \right\}_{i,j=1}^{\#\mathrm{DoFs}(\mathcal{T}_h)} \end{aligned}$$

以及时间矩阵

$$\begin{aligned} M_k &= \left\{ \int_{I_m} \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)}, \\ C_k &= \left\{ \int_{I_m} \partial_t \varphi_k^{(j)} \cdot \varphi_k^{(i)} \, \mathrm{d}t \right\}_{i,j=1}^{\#\mathrm{DoFs}(I_m)}, \\ D_k^1 &= \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & & \\ \vdots & & \ddots & \\ 0 & & & 0 \end{pmatrix}, \qquad D_k^2 = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ & & 0 & 0 \\ & \iddots & & \vdots \\ 0 & & & 0 \end{pmatrix}. \end{aligned}$$

这里，采用时间求积点 $t_1,\dots,t_{r+1}$ 的时间 dG(r) 方法的解向量 U_m 表示为

$$U_m = \begin{pmatrix} U_m(t_1) \\ \vdots \\ U_m(t_{r+1}) \end{pmatrix} = \begin{pmatrix} u_m(t_1) \\ v_m(t_1) \\ \vdots \\ u_m(t_{r+1}) \\ v_m(t_{r+1}) \end{pmatrix}.$$

为了推导 dG(r) 时间步进公式，我们现在只需通过在 (0，k) 上积分来计算时间矩阵M_k和 C_k，其中 $k := t_m - t_{m-1}$ 是时间步长，并代入 (0，k) 上的 dG$-Q^r$ 基函数；由于我们仅有一个单元并使用高斯-洛巴托求积，这些基函数与 Q^r 基函数相同。

A.1 弹性动力学的 dG(1) 公式

将基函数 $\varphi_k^{(1)} = 1 - \frac{t}{k}$，$\varphi_k^{(2)} = \frac{t}{k}$代入时间矩阵 M_k和 C_k，得到

$$\int_0^k \varphi_k^{(1)} \cdot \varphi_k^{(1)} \,\mathrm{d}t = \int_0^k \left(1 - \frac{t}{k}\right)^2 \,\mathrm{d}t = \int_0^k 1 - \frac{2t}{k} + \frac{t^2}{k^2} \,\mathrm{d}t = \frac{k}{3} = \int_0^k \varphi_k^{(2)} \cdot \varphi_k^{(2)} \,\mathrm{d}t,$$
$$\int_0^k \varphi_k^{(1)} \cdot \varphi_k^{(2)} \,\mathrm{d}t = \int_0^k \varphi_k^{(2)} \cdot \varphi_k^{(1)} \,\mathrm{d}t = \int_0^k \left(1 - \frac{t}{k}\right) \cdot \frac{t}{k} \,\mathrm{d}t = \int_0^k \frac{t}{k} - \frac{t^2}{k^2} \,\mathrm{d}t = \frac{k}{6},$$

以及

$$\int_0^k \partial_t\varphi_k^{(1)} \cdot \varphi_k^{(2)} \,\mathrm{d}t = \int_0^k \partial_t \left(1 - \frac{t}{k}\right) \cdot \frac{t}{k} \,\mathrm{d}t = \int_0^k -\frac{t}{k^2} \,\mathrm{d}t = -\frac{1}{2} = \int_0^k \partial_t\varphi_k^{(1)} \cdot \varphi_k^{(1)} \,\mathrm{d}t,$$
$$\int_0^k \partial_t\varphi_k^{(2)} \cdot \varphi_k^{(2)} \,\mathrm{d}t = \int_0^k \partial_t \left(\frac{t}{k}\right) \cdot \frac{t}{k} \,\mathrm{d}t = \int_0^k \frac{t}{k^2} \,\mathrm{d}t = \frac{1}{2} = \int_0^k \partial_t\varphi_k^{(2)} \cdot \varphi_k^{(1)} \,\mathrm{d}t.$$

因此，弹性动力学的 dG(1) 时间步进公式为

$$\left[\frac{1}{2}\begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} \otimes M_h + \frac{k}{6}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix} \otimes K_h\right] \begin{pmatrix} U_m(t_{m-1}) \\ U_m(t_m) \end{pmatrix} = \begin{pmatrix} F_m(t_{m-1}) + U_{m-1}(t_{m-1})M_h \\ F_m(t_m) \end{pmatrix},$$

其中我们利用了 dG(1) 的时间求积点为 t_{m-1} 和 t_m 这一事实。

A.2 弹性动力学的 dG(2) 公式

对二次基函数重复 A.1 节中的过程，我们得到弹性动力学的 dG(2) 时间步进公式

$$\left[\frac{1}{6}\begin{pmatrix} 3 & 4 & -1 \\ -4 & 0 & 4 \\ 1 & -4 & 3 \end{pmatrix} \otimes M_h + \frac{k}{30}\begin{pmatrix} 4 & 2 & -1 \\ 2 & 16 & 2 \\ -1 & 2 & 4 \end{pmatrix} \otimes K_h\right] \begin{pmatrix} U_m(t_{m-1}) \\ U_m(t_{m-\frac{1}{2}}) \\ U_m(t_m) \end{pmatrix} = \begin{pmatrix} F_m(t_{m-1}) + U_{m-1}(t_{m-1})M_h \\ F_m(t_{m-\frac{1}{2}}) \\ F_m(t_m) \end{pmatrix},$$

其中我们利用了 dG(2) 的时间求积点为 t_{m-1}、$t_{m-\frac{1}{2}} := t_{m-1} + \frac{k}{2}$ 和 t_m 这一事实。

备注 A.1. dG(1)和dG(2)格式也可在*[62]*中关于常微分方程模型的第 *7.1* 节和第 *7.2* 节找到。

参考文献

[1] M. Abbaszadeh、M. Dehghan、A. Khodadadian、N. Noii、C. Heitzinger 和 T. Wick。基于适当正交分解的降阶变分多尺度插值无网格伽辽金技术，用于求解与传热方程耦合的纳维–斯托克斯方程：非定常不可压缩布西内斯克方程。《*J. Comput. Phys.*》，426:109875，2021年。[2] D. Arndt、W. Bangerth、D. Davydov、T. Heister、L. Heltai、M. Kronbichler、M. Maier、J.-P. Pelteret、B. Turcksin 和 D. Wells。deal.II 有限元库：设计、特性与见解。《计算数学及其应用》，81:407–422，2021年。[3] D. Arndt、W. Bangerth、M. Feder、M. Fehling、R. Gassmöller、T. Heister、L. Heltai、M. Kronbichler、M. Maier、P. Munch、J.-P. Pelteret、S. Sticko、B. Turcksin 和 D. Wells。deal.II 库，9.4版。《数值数学杂志》，30(3):231–246，2022年。[4] 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。《*Int. J. Numer. Methods Eng.*》，118(4):209–241，2019年。[5] J. Baiges、R. Codina 和 S. Idelsohn。用于不可压缩纳维–斯托克斯方程稳定化有限元近似的显式降阶模型。国际数值方法流体杂志，72(12):1219–1243，2013年。[6] F. Ballarin、A. Manzoni、A. Quarteroni 和 G. Rozza。参数化定常不可压缩纳维–斯托克斯方程 POD–伽辽金近似的上确界稳定化。《*Int. J. Numer. Methods Eng.*》，102(5):1136–1161，2015年。[7] W. Bangerth、M. Geiger 和 R. Rannacher。波动方程的自适应伽辽金有限元方法。计算方法与应用数学，10(1):3–48，2010年。[8] W. Bangerth 和 R. Rannacher。微分方程的自适应有限元方法。Birkhäuser 出版社„ 2003年。[9] D. Baroli、C. M. Cova、S. Perotto、L. Sala 和 A. Veneziani。参数化流体动力学问题的 Hi-POD 求解：初步结果。研究报告，MOX，数学系，米兰理工大学，2016年7月，https://doi.org/10.13140/RG.2.1.2993.0482。[10] R. Becker 和 R. Rannacher。有限元方法中误差控制的反馈方法：基本分析与示例。东西方数值数学杂志，4:237–264，1996年。[11] R. Becker 和 R. Rannacher。有限元方法后验误差估计的最优控制方法。*Acta Numer.*，10:1 – 102，05 2001年。[12] P. Benner、A. Cohen、M. Ohlberger 和 K. Willcox。模型降阶与近似：理论与算法。工业与应用数学学会，费城，2015年。

[13] P. Benner, W. Schilders, S. Grivet-Talocia, A. Quarteroni, G. Rozza, 和 L. Miguel Silveira。《模型降阶：第2卷：基于快照的方法与算法》。De Gruyter，2020年。[14] M. Bernreuther 和 S. Volkwein。针对非光滑抛物型偏微分方程的自适应认证时空降阶基方法，2022年，https://doi.org/10.48550/ARXIV.2212. 13744。[15] L. Bertagna 和 A. Veneziani。通过求解逆流固耦合问题，采用变分估计方法对血管顺应性进行模型降阶。《Inverse Probl.》，30(5):055006，2014年。[16] M. Besier 和 R. Rannacher。有限元伽辽金法中面向非定常不可压缩流计算的目标导向时空自适应。《Int. J. Num. Meth. Fluids》，70:1139–1166，2012年。[17] M. Brand。含缺失值的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。[18] M. Brand。瘦奇异值分解的快速低秩修正。《Linear Algebra Appl.》，415(1):20–30，2006年。[19] 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研究。《J. Comput. Phys.》，259:598–616，2014年。[20] F. Chinesta, P. Ladevèze, 和 E. Cueto。基于适当广义分解的模型降阶简述。《Arch. Comput. Methods Eng.》，18(4):395–404，2011年。[21] Y. Choi, P. N. Brown, B. Arrighi, 和 R. Anderson。用于大型线性动力系统的时空降阶模型及其在玻尔兹曼输运问题中的应用。《J. Comput. Phys.》，424:109845，2019年。[22] E. A. Christensen, M. Brøns, 和 J. N. Sørensen。对应用于参数依赖的非湍流的本征正交分解类分解技术进行评估。《SIAM《科学计算杂志》》，21(4):1419–1434，1999年。[23] M. Delfour, W. Hager, 和 F. Trochu。常微分方程的间断伽辽金方法。《Math. Comp.》，36:455–473，1981年。[24] D. Di Pietro 和 A. Ern。《间断伽辽金方法的数学方面》。Mathematiques et Applications。施普林格柏林海德堡，2011年。[25] F. Ekre, F. Larsson, K. Runesson, 和 R. Jänicke。计算均匀化中多孔介质数值模型降阶的后验误差估计。《Int. J. Numer. Methods Eng.》，121(23):5350–5380，2020年。[26] B. Endtmayer。《非线性偏微分方程的多目标导向后验误差估计》。博士论文，林茨约翰内斯·开普勒大学，2021年。

[27] B. Endtmayer, U. Langer, 和 T. Wick。非线性问题的多目标导向误差估计。《数值数学杂志》，27(4):215–236，2019年。[28] B. Endtmayer, U. Langer, 和 T. Wick。对偶加权残差法的双侧后验误差估计。SIAM《科学计算杂志》，42(1):A371–A394，2020年。[29] K. Eriksson, K. Eriksson, D. Estep, P. Hansbo, 和 C. Johnson。《计算微分方程》。计算微分方程。剑桥大学出版社，1996年。[30] K. Eriksson, D. Estep, P. Hansbo, 和 C. Johnson。微分方程自适应方法简介。《Acta Numer.》，4:105–158，1995年。[31] H. Fareed 和 J. R. Singler。关于连续时间数据增量POD算法的说明。《应用数值数学》，144:223–233，2019年。[32] H. Fareed, J. R. Singler, Y. Zhang, 和 J. Shen。偏微分方程模拟数据的增量式本征正交分解。《计算数学及其应用》，75(6):1942–1960，2018年。[33]M. Girfoglio, A. Quaini, 和 G. Rozza。纳维–斯托克斯方程流函数–涡量形式的POD-Galerkin降阶模型。《计算流体力学》，244:105536，2022年。[34] C. Gräßle 和 M. Hinze。利用任意有限元离散对演化方程进行POD降阶建模。《计算数学进展》，44(6):1941–1978，2018年。[35] M. Gubisch 和 S. Volkwein。第1章：线性二次最优控制的本征正交分解：理论与算法。载于《模型降阶与近似》，第3–63页。工业与应用数学学会，2017年7月。[36] M. D. Gunzburger, J. S. Peterson, 和 J. N. Shadid。边界数据含多个参数的时变偏微分方程降阶建模。《计算方法与应用力学和工程》，196(4-6):1030–1047，2007年。[37] B. Haasdonk。第2章：参数化偏微分方程的降阶基方法——定常与非定常问题教程。载于《模型降阶与近似》，第65–136页。工业与应用数学学会，2017年。[38] B. Haasdonk 和 M. Ohlberger。参数化线性演化方程有限体积近似的降阶基方法。《ESAIM数学建模与数值分析》，42(2):277–302，2008年。[39] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, 和 T. E. Oliphant。使用 NumPy 进行数组编程。《自然》，585(7825):357–362，2020年9月。

[40] J. S. Hesthaven、G. Rozza 和 B. Stamm。参数化微分方程。载于 J. S. Hesthaven、G. Rozza 和 B. Stamm（编），参数化偏微分方程的认证降阶基方法，SpringerBriefs in Mathematics，第 15–25 页。Springer 国际出版公司，2016年。[41] T. Hughes 和 G. Hulbert。弹性动力学的时空有限元方法：公式与误差估计。《计算方法与应用力学和工程》，66：339–363，1988年2月。[42] G. Kerschen、J.-C. Golinval、A. F. Vakakis 和 L. A. Bergman。用于机械系统动力学表征与降阶的本征正交分解方法：综述。非线性动力学，41(1)：147–169，2005年8月。[43] Y. Kim、K. Wang 和 Y. Choi。使用 Python 以不到 120 行代码实现线性动力系统的高效时空降阶模型。数学，9(14)：1690，2021年。[44] A. Kolmogoroff。关于给定函数类中函数的最佳逼近。*Ann. Math.*，第 107–110 页，1936。[45] N. Kühl、H. Fischer、M. Hinze 和 T. Rung。增量奇异值分解方法：用于大规模空间并行、分布式但时间串行的数据，并应用于工程流动，2023，https://doi.org/10.48550/ARXIV.2302.09149。[46] K. Kunisch 和 S. Volkwein。抛物型问题的 Galerkin 正交适当分解方法。数值数学，90(1)：117–148，2001年11月。[47] K. Kunisch 和 S. Volkwein。流体动力学中一般方程的 Galerkin 正交适当分解方法。*SIAM* 数值分析期刊，40(2)：492–515，2002年。[48] U. Langer 和 O. Steinbach（编）。时空方法：偏微分方程中的应用。Radon 计算与应用数学丛书第 25 卷，柏林。De Gruyter，2019年。[49] 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。载于用于建模与计算降阶的降阶方法，第 9 卷，第 235–273 页。Springer 国际出版公司，2014年。[50] C. Lehrenfeld、F. Heimann、J. Preuß 和 H. von Wahl。“ngsxfem”：NGSolve 的扩展插件，用于几何非拟合有限元离散。开源软件期刊，6(64)：3237，2021年。[51] M. Loveland、E. Valseth、M. Lukac 和 C. Dawson。扩展 FEniCS，以利用张量积有限元处理更高维问题。计算机科学期刊，64：101831，2022年。[52] Z. Luo、J. Chen、P. Sun 和 X. Yang。基于本征正交分解的抛物型方程有限元公式。中国科学：数学，52(3)：585–596，2009年3月。

[53]M. Meyer 和 H. G. Matthies。利用 Karhunen-Loève 展开和双权重残差方法实现非线性动力学的高效模型降阶。计算力学，31(1)：179–191，2003年5月。[54] N.-C. Nguyen、G. Rozza 和 A. T. Patera。时变黏性 Burgers 方程的降阶基近似与后验误差估计。*Calcolo*，46(3)：157–185，2009年。[55] J. T. Oden。自适应多尺度预测建模。*Acta Numer.*，27：353–450，2018年。[56] S. Perotto、M. G. Carlino 和 F. Ballarin。通过变量分离进行模型降阶：分层模型降阶与适当广义分解的比较。载于 S. J. Sherwin、D. Moxey、J. Peiró、P. E. Vincent 和 C. Schwab（编），偏微分方程谱方法与高阶方法会议 *ICOSAHOM 2018*，第 134 卷，第 61–77 页。Springer，2020年8月。[57] S. Perotto 和 A. Zilio。抛物型方程的时空自适应分层模型降阶。*Adv. Model. Simul. Eng. Sci.*，2，2015年12月。[58] J. Preuß。移动区域上的高阶非拟合等参时空有限元法。硕士论文，哥廷根乔治-奥古斯特大学，2018年。[59] A. Rademacher。二阶非线性双曲问题的自适应有限元方法。博士论文，多特蒙德工业大学，2009年。[60] R. Rannacher。数值分析*1*：常微分方程数值分析。海德堡大学出版社，2017年。[61] S. S. Ravindran。利用本征正交分解对流体进行最优控制的降阶方法。国际数值方法流体杂志，34(5)：425–448，2000年。[62] T. Richter、A. Springer 和 B. Vexler。用于抛物型问题时间离散的间断伽辽金方法的高效数值实现。数值数学，124(1)：151–182，2013年5月。[63] J. Roth、J. P. Thiele、U. Köcher 和 T. Wick。面向非定常流动问题的张量积时空目标导向误差控制与自适应：采用单位分解双权重残差，2022年，https://doi.org/10.48550/ARXIV.2210.02965。[64] G. Rozza。基于最优流动控制与降阶基技术的形状设计。博士论文，2005年。[65] G. Rozza、D. B. P. Huynh 和 A. T. Patera。仿射参数化椭圆强制偏微分方程的降阶基近似与后验误差估计。《*Arch. Comput. Methods Eng.*》，15(3)：229–275，2008年。[66] A. Schafelner。时空有限元方法。博士论文，林茨约翰内斯·开普勒大学，2021年。

[67]M. Schmich. 用于计算非定常不可压缩流的自适应有限元方法. 博士论文，海德堡大学，2009年。[68] M. Schmich and B. Vexler. 用于抛物型方程时空有限元离散的动态网格自适应。*SIAM*《科学计算杂志》，30(1):369–393，2008年。[69] Y. S. Shimizu and E. J. Parish. 面向非线性动力系统的窗口化时空最小二乘 Petrov–Galerkin 模型降阶。《计算方法与应用力学和工程》，386:114050，2021年。[70] 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。[71] M. K. Sleeman and M. Yano. 面向参数化时变非线性偏微分方程的目标导向模型降阶。《计算方法与应用力学和工程》，388:114206，2022年。[72] R. Tenderini, N. Mueller, and S. Deparis. 参数化非定常 Stokes 方程的时空降阶基方法，2022年，https://doi.org/10.48550/ARXIV.2206.12198。[73]J. P. Thiele and T. Wick. 抛物型问题中时空对偶加权残差估计子的变分单位分解局部化，2022年，https://doi.org/10.48550/ARXIV. 2207.04764。[74] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0：Python 科学计算基础算法。《自然·方法》，17:261–272，2020年。[75] S. Volkwein.《本征正交分解：理论与降阶建模》讲义。2013年8月，http://www.math.uni-konstanz.de/numerik/personen/volkwein/ teaching/POD-Book.pdf。[76] Q. Wang, N. Ripamonti, and J. S. Hesthaven. 基于 Mori-Zwanzig 形式的参数化 POD-Galerkin 降阶模型的循环神经网络闭合。《*J. Comput. Phys.*》，410:109402，2020年。[77] K. Willcox and J. Peraire. 基于本征正交分解的平衡模型降阶。《*AIAA* 杂志》，40(11):2323–2330，2002年。[78] J. Wloka, C. Thomas, and M. Thomas. 偏微分方程。剑桥大学出版社，1987年。[79] M. Yano. 参数化非线性偏微分方程的目标导向模型降阶：在空气动力学中的应用。《*Int. J. Numer. Methods Eng.*》，121(23):5200–5226，2020年。

[80] Y. Zhang. 关于增量奇异值分解中一个未决问题的解答，2022年，https://doi.org/10.48550/ARXIV.2204.05398。

[81] F. Zoccolan、M. Strazzullo 和 G. Rozza。用于最优控制下对流占优偏微分方程的流线迎风 Petrov-Galerkin 降阶方法，2023年，https://doi.org/10.48550/ARXIV.2301.01973。