MULTISCALE MODEL. SIMUL.
Vol. 20, No. 3, pp. 1093-1126

MANIFOLD LEARNING AND NONLINEAR HOMOGENIZATION*

SHI CHEN†, QIN LI‡, JIANFENG LU§, AND STEPHEN J. WRIGHT¶

Abstract. We describe an efficient domain decomposition-based framework for nonlinear multiscale PDE problems. The framework is inspired by manifold learning techniques and exploits the tangent spaces spanned by the nearest neighbors to compress local solution manifolds. Our framework is applied to a semilinear elliptic equation with oscillatory media and a nonlinear radiative transfer equation; in both cases, significant improvements in efficacy are observed. This new method does not rely on a detailed analytical understanding of multiscale PDEs, such as their asymptotic limits, and thus is more versatile for general multiscale problems.

Key words. nonlinear homogenization, multiscale problems, manifold learning, domain decomposition

MSC code. 65N99

DOI. 10.1137/20M1377771

1. Introduction.

Homogenization is a body of theory and methods that studies differential or differential-integro equations with rapidly oscillating coefficients. It can be traceed back to the famous work of Bensoussan, Lions, and Papanicolaou [22] and builds on several other important developments [1, 12, 17, 38, 39, 50, 51, 58]. Generally speaking, the goal of homogenization is to derive asymptotic limiting equations as accurate surrogates of the original equations that do not have scale separations. The core technique is asymptotic analysis.

1.1. Goal.

There are a number of famous examples that use homogenization techniques, such as elliptic equations with rapidly oscillating media [22], the Schrödinger equation with small rescaled Planck constant [48], the neutron transport equation with small Knudsen number [21, 52], the compressible Euler equation with small Mach number [67, 68, 83], and Boltzmann-type equations in the fluid regime [14]. All these examples have the form

$$\mathcal{N}^\epsilon u^\epsilon = f\,, \tag{1.1}$$

* Received by the editors November 2, 2020; accepted for publication (in revised form) March 22, 2022; published electronically September 27, 2022.
https://doi.org/10.1137/20M1377771
Funding: The work of the third author was supported in part by the National Science Foundation via grants DMS-1454939 and DMS-2012286. The work of the first, second, and fourth authors was supported in part by the National Science Foundation via grants CCF-1740707 and DMS-2023239. The work of the fourth author was further supported in part by the National Science Foundation via grant 1934612, by Subcontract 8F-30039 from Argonne National Laboratory, and by Award N660011824020 from the DARPA Lagrange Program. The first and second author's work was further supported in part by Wisconsin Data Science Initiative and by the National Science Foundation via grant DMS-1750488.

† Department of Mathematics, University of Wisconsin-Madison, Madison, WI 53706 USA (schen636@wisc.edu).

‡ Mathematics Department and Discovery Institute, University of Wisconsin-Madison, Madison, WI 53706 USA (qinli@math.wisc.edu).

§ Department of Mathematics, Department of Physics, and Department of Chemistry, Duke University, Durham, NC 27708 USA (jianfeng@math.duke.edu).

¶ Department of Computer Sciences, University of Wisconsin, Madison, WI 53706 USA (swright@cs.wisc.edu).

where $\mathcal{N}^\epsilon$ is a partial differential operator that depends explicitly on the small parameter ϵ. The term f on the right-hand side represents the external information—the source terms, the boundary conditions, the initial conditions, and so on—which does not depend on ϵ. Due to the ϵ-dependence of $\mathcal{N}^\epsilon$, the PDE is rather stiff: the solutions either exhibit high oscillations (such as the Schrödinger equation with small value of the rescaled Planck constant, or the elliptic equation with rough media) or present boundary/initial layers within which solutions change rapidly (such as the Knudsen layer in kinetic systems). The oscillations and layers themselves usually do not carry any interesting physical information; one is more interested in extracting physically meaningful quantities from the solutions directly, with these details omitted. Thus, it is important to evaluate the limiting behavior of (1.1) as $\epsilon \to 0$. There are two contrasting approaches in the literature that enable this task: one is analytical and the other is numerical.

The analytical approach seeks the asymptotic limit of the PDE (1.1) defined as follows:

$$\mathcal{N}^* u^* = f\,. \tag{1.2}$$

The term "asymptotic limit" refers to the fact that for any reasonable f, in a certain space with a certain metric, we have

$$\|u^\epsilon - u^*\| \to 0 \quad \text{as } \epsilon \to 0. \tag{1.3}$$

A classical way to derive this limit is to perform Hilbert expansion in terms of ϵ. Here, we define the ansatz

$$u^\epsilon = u_0 + \epsilon u_1 + \epsilon^2 u_2 + \cdots\,,$$

substitute into (1.1), and then balance the two sides in terms of ϵ. Typically, at some level of the expansion, a closure is performed to derive the effective operator $\mathcal{N}^*$. This framework is highly effective and general; we will give explicit examples in later sections.

On the numerical side, we look for cheap solvers that compute the asymptotic limits. A typical requirement for classical numerical solvers to be accurate is that the discretization has to resolve the smallness of ϵ. This can lead to high numerical and memory costs, which are sometimes beyond reasonable computational resources. The focus of "numerical homogenization" or "asymptotic preserving" is thus to design schemes that capture asymptotic limits of the solutions with relaxed (and thus more efficient) discretization requirements. One technique is to explore analytical results and translate them into the discrete setting: first, the asymptotic limiting equations are derived, and then a "macrosolver" for the limiting equation and a "microsolver" that solves the original equation are combined in some way. This strategy has been applied to deal with Boltzmann-type equations, the Schrödinger equation, and elliptic equations with highly oscillatory media, for the tasks of designing "asymptotic preserving" schemes, finding semiclassical limits, and performing "numerical homogenization." There is a significant drawback to this approach, however: the design of the numerical method is based completely on analytical understanding, so numerical development necessarily lags behind analytical progress. This fact significantly limits the role of multiscale computation.

This observation motivates the question that we address in this paper. Given a system of the form (1.1), knowing it has an asymptotic limit (1.2) but not knowing the specific form of this limit, can we design an efficient, accurate solver?

1.2. Approach. In this paper, we propose a numerical approach based on "compression." Classical methods require the use of $N_\epsilon \sim \frac{1}{\epsilon^\alpha}$ grid points to achieve accuracy and stability in solving (1.1) for some power $\alpha > 0$. Note that N_ϵ blows up to infinity as $\epsilon \to 0$. By contrast, the limiting equation (1.2) is independent of ϵ, so we typically require only N_* grid points (a number that is independent of ϵ) to solve this system. Thus, the information carried in N_ϵ degrees of freedom can potentially be "compressed" into N_* degrees of freedom, provided that we can tolerate an asymptotic error of order ϵ in the solution (see (1.3)).

How can we design an approach to solving (1.1) that exploits compression? Our roadmap consists of three steps: (a) identify the solution set that can be compressed; (b) compress the set into a smaller effective solution set; and (c) for a given new data point $f(x)$, single out the solution from the effective set. We call the steps (a) and (b) the offline stage and step (c) the online stage.

For linear equations, this roadmap has been followed by several authors [23, 26, 27, 28]. When the setup is linear, the solution set is a space, and thus information is entirely coded in representative basis functions. These basis functions can be found in the offline stage, and a Galerkin formulation can then be used to identity the linear combination of the basis for a given $f(x)$ in the online stage. To find the representative basis functions, one can utilize the random sampling technique developed for finding low-rank structures of matrices in [54], where the authors proved that a few random samples are able to reconstruct the low-rank column space with high probability; see [27].

In this article, we develop a roadmap in the nonlinear setting. The extension is not straightforward. Since the solution set is not a space in the nonlinear setup, the notion of "basis function" does not even exist. Instead, we seek an N_*-dimensional approximating manifold in an N_ϵ-dimensional space. For every given $f(x)$, there is a corresponding numerical solution u^ϵ to the original equation (1.1) in the N_ϵ space. Within ϵ distance there exists its homogenized solution u^* to the limiting equation (1.2). Since u^* relies on only N_* degrees of freedom, as $f(x)$ varies, the variations of u^* form a manifold of dimension at most N_*.

By using this argument, we formulate the homogenization problem (in the nonlinear setting)as a manifold-learning problem. Suppose we can generate a few configurations of $f(x)$ and compute the associated numerical solutions; can we learn to represent the solution manifold? Further, given a completely new configuration of $f(x)$, can we quickly identify the corresponding solution? These two questions are addressed in the offline and online stages, respectively.

Many different approaches have been proposed for manifold learning based on observed point clouds. They typically look for key features that the points share either locally (as in local linear embedding (LLE) [82], multiscale SVD [8, 76], local tangent space alignment [88]) or globally (as in the use of heat kernels [19, 33]). The strategy we propose here is not a direct application of any of these ideas, but it uses elements of the LLE and multiscale SVD approaches. Specifically, we seek local linear approximations to the solution map and cover the solution manifold with a number of these tangent space "patches."

We define the solution map as follows:

$$\mathcal{S}^\epsilon : \quad f \in \mathcal{X} \to u^\epsilon \in \mathcal{Y}\,. \tag{1.4}$$

It maps the source term and initial/boundary conditions captured in $f(x)$ to the solution of the equation (1.1). To find the solution manifold, we randomly sample a large number of configurations f_i in $\mathcal{X}$ and compute the solution $u_i^\epsilon = \mathcal{S}^\epsilon f_i \in \mathcal{Y}$

associated with each of these configurations. These solutions form a point cloud in a high-dimensional space $\mathcal{Y}$. We subdivide the set of configurations $\{f_i\}$ into a number of small neighborhoods, and we look for the tangential approximation to the mapping (1.4) on each of these neighborhoods. Given a configuration f, we identify the neighborhood to which it belongs and interpolate linearly to obtain the corresponding solution.

We summarize our online-offline strategy as follows. (Some modifications described in section 2 will reduce the cost of implementation.)

Offline: Randomly sample $f_i(x)$, $i = 1, \ldots, N$, and find solutions $u_i^\epsilon = \mathcal{S}^\epsilon f_i$;
Online: Given $f(x)$:

Step 1: Identify the k-nearest neighbors of $f(x)$, call them $f_{i_j}, j = 1, 2, \ldots, k$, with f_{i_1} being the nearest neighbor;
Step 2: Compute

$$\mathcal{S}^\epsilon \phi \approx u_{i_1}^\epsilon + \mathsf{U} \cdot c\,, \quad \text{with} \quad \mathsf{U} = \begin{bmatrix} | & & | \\ u_{i_2}^\epsilon - u_{i_1}^\epsilon & \ldots & u_{i_k}^\epsilon - u_{i_1}^\epsilon \\ | & & | \end{bmatrix},$$

where c is a set of coefficients that fits $f - f_{i_1}$ with a linear combination of $f_{i_j} - f_{i_1}$ for $j = 2, 3, \ldots, k$.

In **Step 2** we used the fact that the solution manifold is of low dimension locally. To make the strategy mathematically precise, we need to address several questions, including the following.

- How should we sample $f_i(x)$ during the offline step?
- What metric should we use to quantify distance?
- Since computing each solution map $u_i^\epsilon = \mathcal{S}^\epsilon f_i$ is expensive, is there any way to reduce the cost further?

We discuss these questions in the following sections. We stress that the manifold learning technique that we investigate in this paper works best when the intrinsic dimensionality of the problem is significantly smaller than the typical required degrees of freedom, and this holds true for all homogenizable problems where the discretization of the limiting equation eliminates the ϵ-dependence. For problems without ϵ-dependence and in which the dimension of the numerical solution is only moderately large, the approach that we take is not expected to reduce the cost.

1.3. The layout of the paper. We discuss the general recipe of the algorithm in section 2 and then show how the approach can be applied to two examples (a semilinear elliptic equation and a nonlinear radiative transfer equation coupled with a temperature term) in section 3 and section 4, respectively. In both sections, we review the relevant homogenization theory for the equations, study the low-rank structure of the tangential solution spaces, and present numerical evidence for the efficacy of our approach.

2. Framework. Our approach is a domain decomposition algorithm that makes use of Schwarz iteration.

After decomposing the domain into multiple overlapping patches, the Schwarz method solves the PDE in each patch, conditioned on agreement of the solutions in the overlapping regions, which are boundary regions for the adjacent patches. At the initial step, these boundary conditions are unknown, so some initial guess is made. Subsequently, the solution of PDEs on each patch alternates with updates of the solution on the overlapping regions until convergence is obtained with respect to certain

criteria. The cost of the entire process is determined by the number of iterations and by the cost of the local solves, noting that, as with any domain decomposition method, the local solves can be performed in parallel. The approach is efficient when the local solves can be performed much more efficiently on the available computing resources than a solver that does not decompose the domain. The optimal domain partitioning depends on the conditioning of the problem and is often specific to the problem under study. Comprehensive descriptions of the Schwarz method appear in [84, 85].

This basic Schwarz iteration does not fully address the issue of ϵ-dependence that we discussed in section 1, since local solvers still necessarily depend on ϵ. As a step toward making use of compression, we take the viewpoint that the purpose of the local solution step is to implement a boundary-to-boundary map, taking one part of the boundary conditions on a patch and using the solution of the resulting PDE to update the boundary conditions for its neighboring patches. We propose to learn the boundary-to-boundary maps in an "offline" stage by running the local solvers as many times as needed to attain the desired accuracy in this map. This offline stage comes with a high overhead cost, but the computation is done only once, and we hope that the cost of the online stage is greatly reduced by having the boundary-to-boundary maps available. Note that this "offline" learning process is distinct from the offline stage discussed in section 1. With the application of domain decomposition, it is the local behavior that needs to be learned instead of the full u^ϵ.

In the linear setting, building the boundary-to-boundary maps is quite straightforward. It amounts roughly to finding all discrete Green's functions, with the degree of freedom being determined by the number of grid points on the patch boundary, with one Green's function per grid point. In the nonlinear setting, the boundary-to-boundary map is nonlinear, so we can no longer build a linear basis, and we turn to a manifold-learning approach to approximate the map. Specifically, in the offline stage, we would sample randomly some configurations and find the corresponding image under the map. The resulting point cloud in high-dimensional space can be viewed as samples of the manifold, which we can then learn by means of local approximate tangential planes. In the online stage, these tangential planes are used as surrogates to local boundary-to-boundary maps.

Before presenting details of the offline and online stage computations, we specify the setup and notation. We consider the following nonlinear PDE with Dirichlet boundary conditions in a domain $\Omega \subset \mathbb{R}^2$:

$$\begin{cases} \mathcal{N}^\epsilon u^\epsilon = 0 & \text{in } \Omega, \\ u^\epsilon = \phi & \text{on } \partial\Omega, \end{cases} \tag{2.1}$$

where, as usual, ϵ indicates the small scale of the problem. For simplicity, we will assume throughout a square geometry $\Omega = [0, L]^2$. The domain Ω is decomposed into overlapping rectangular patches defined by

$$\Omega = \bigcup_{m \in J} \Omega_m, \quad \text{with} \quad \Omega_m = (x^{(1)}_{m_1}, x^{(2)}_{m_1}) \times (y^{(1)}_{m_2}, y^{(2)}_{m_2}), \tag{2.2}$$

where $m = (m_1, m_2)$ is a multi-index and J is the collection of the indices

$$J = \{m = (m_1, m_2) : m_1 = 1, \ldots, M_1, m_2 = 1, \ldots, M_2\}.$$

This setup is illustrated in Figure 1. For each patch we define the associated partition-of-unity function χ_m, which has $\chi_m(x) \geq 0$ and

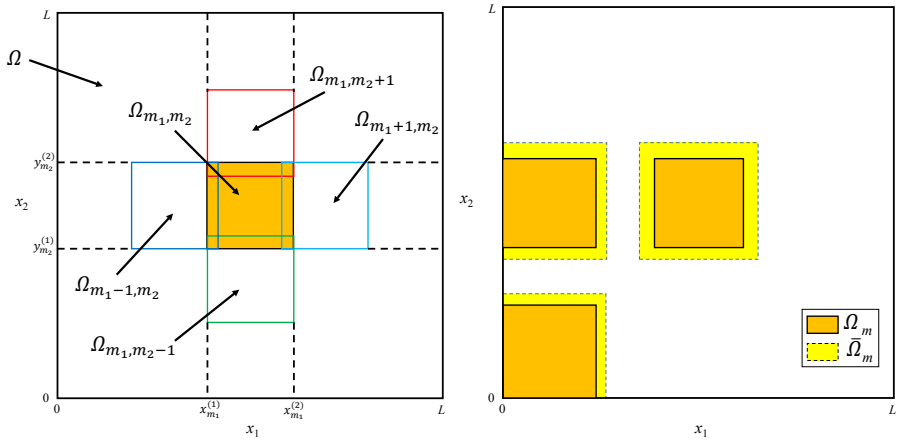


FIG. 1. *The plot on the left shows the domain decomposition for a square geometry. Each patch is labeled by a multi-index $m = (m_1, m_2)$. The adjacent patches of Ω_m are defined to be the patches on its north/south/west/east sides. The plot on the right demonstrates the use of local enlargement to dampen boundary effects.*

$$\chi_m(x) = 0 \quad \text{on } x \in \Omega \backslash \Omega_m\,, \qquad \sum_m \chi_m(x) = 1 \quad \forall x \in \Omega\,. \tag{2.3}$$

We set $\partial\Omega_m$ to be the boundary of patch Ω_m and denote by $\mathcal{N}(m)$ the collection of indices of the neighbors of Ω_m. In this particular 2D case, we have

$$\mathcal{N}(m) = \{(m_1 \pm 1, m_2)\} \cup \{(m_1, m_2 \pm 1)\} \subset J\,. \tag{2.4}$$

Assume that the equation (2.1) is well-posed, meaning that given ϕ in some function space $\mathcal{X}$, there exists a unique solution u^ϵ in another function space $\mathcal{Y}$. Assume further that the local nonlinear equation on patch Ω_m defined by

$$\begin{cases} \mathcal{N}^\epsilon u_m^\epsilon = 0 & \text{in } \Omega_m\,, \\ u_m^\epsilon = \phi_m & \text{on } \partial\Omega_m \end{cases}$$

is well-posed, given local boundary condition ϕ_m in some function space $\mathcal{X}_m$, and that the solution u_m^ϵ lives in space $\mathcal{Y}_m$. We further define the following operators.

- $\mathcal{S}_m^\epsilon$ denotes the solution operator that maps local boundary condition ϕ_m to the local solution u_m^ϵ:

$$\mathcal{S}_m^\epsilon : \mathcal{X}_m \to \mathcal{Y}_m, \quad \mathcal{S}_m^\epsilon \phi_m = u_m^\epsilon\,.$$

- $\mathcal{I}_m^l$ denotes the trace operator for all $l \in \mathcal{N}(m)$:

$$\mathcal{I}_m^l u_l^\epsilon = u_l^\epsilon|_{\partial\Omega_m \cap \Omega_l}, \quad l \in \mathcal{N}(m)\,,$$

  which takes the value of u_l^ϵ restricted on the boundary $\partial\Omega_m \cap \Omega_l$. Here we assume that the space $\mathcal{Y}_l$ allows for trace.

- $\mathcal{P}_m$ denotes the boundary update operator, mapping $\bigoplus_{l\in\mathcal{N}(m)} \mathcal{X}_l$ to $\mathcal{X}_m$:

$$\mathcal{P}_m(\phi_l,\, l \in \mathcal{N}(m)) = \begin{cases} \mathcal{I}_m^l \mathcal{S}_l^\epsilon \phi_l & \text{on } \partial\Omega_m \cap \Omega_l, \quad l \in \mathcal{N}(m)\,, \\ \phi|_{\partial\Omega_m \cap \partial\Omega} & \text{on } \partial\Omega_m \cap \partial\Omega\,. \end{cases}$$

Note that on the points in $\partial\Omega_m \cap \partial\Omega$, the boundary condition from the whole domain Ω is imposed.

The offline and online stages of the algorithm essentially construct and evaluate $\mathcal{P}_m$, as we now show.

2.1. Offline stage. The goal of the offline stage is to construct a dictionary to approximate $\mathcal{P}_m$ for every $m \in J$. To eliminate any boundary layer effect, we enlarge each local patch slightly by adding a margin around its edges (except for the edges that correspond to only part of the boundary of the whole domain). The enlarged domains are denoted by $\widetilde{\Omega}_m$ and illustrated in Figure 1.

We denote by $\widetilde{\mathcal{X}}_m$ the space of boundary conditions on $\partial\widetilde{\Omega}_m$ equipped with norm $\|\cdot\|$, and define a ball in $\widetilde{\mathcal{X}}_m$ as follows:

$$B(R_m;\, \widetilde{\mathcal{X}}_m) = \{\widetilde{\phi} \in \widetilde{\mathcal{X}}_m : \|\widetilde{\phi}\| \le R_m\}\,.$$

First, we draw N samples randomly from the ball, as follows:

$$\widetilde{\phi}_{m,i} \in B(R_m;\, \widetilde{\mathcal{X}}_m)\,, \quad i = 1, \dots, N\,.$$

(The specific measure used in drawing depends on the particular problem being considered; we will make it more precise in the examples below.) For these samples we obtain local solutions $\widetilde{u}_{m,i}$ from the following PDEs:

$$\begin{cases} \mathcal{N}^\epsilon \widetilde{u}^\epsilon_{m,i} = 0 & \text{in } \widetilde{\Omega}_m, \\ \widetilde{u}^\epsilon_{m,i} = \widetilde{\phi}_{m,i} & \text{on } \partial\widetilde{\Omega}_m. \end{cases} \tag{2.5}$$

We build a dictionary from these solutions by confining them in the interior Ω_m and the boundary $\partial\Omega_m$:

$$\mathscr{I}_m = \{\psi_{m,i} = \widetilde{u}_{m,i}|_{\Omega_m}\}_{i=1}^N, \quad \mathscr{B}_m = \{\phi_{m,i} = \widetilde{u}_{m,i}|_{\partial\Omega_m}\}_{i=1}^N\,. \tag{2.6}$$

Since the problems that we consider are homogenizable, meaning that the solution manifold is of low dimension, the value of N can be relatively small.

Remark 2.1. Two remarks are in order.

- How do we sample? That is, how do we find a measure μ_m on $\widetilde{\mathcal{X}}_m$ for drawing samples? To make the setting more precise, we discretize the space $\widetilde{\mathcal{X}}_m$ and equip it with norm $\|\cdot\|_h$, and define a measure μ^h_m on the ball $B(R_m;\, \widetilde{\mathcal{X}}^h_m)$. Denoting the dimension of $\widetilde{\mathcal{X}}_m$ by p, we sample the magnitude and the angle separately, that is, we take the measure as a product $\mu^h_m = \mu_{r,m} \otimes \mu_{S,m}$, with $\mu_{r,m}$ being the radial part on $(0, R_m)$ and $\mu_{S,m}$ being the measure on the unit sphere $S^{p-1} = \{\phi \in \widetilde{\mathcal{X}}^h_m : \|\phi\|_h = 1\} \subset \mathbb{R}^p$. The angular measure $\mu_{S,m}$ is chosen to be the uniform, and the radial part $\mu_{r,m}$ has a density function $f(r) = \frac{D+1}{R_m^D} r^D$. The number D here plays the role of effective dimension; it should depend on the expected dimension of solution manifold. Note that if we take $D = p - 1$, the measure μ^h_m is exactly the uniform measure on the full ball $B(R_m;\, \widetilde{\mathcal{X}}^h_m) \subset \mathbb{R}^p$. The question of selecting D in a rigorous way is left to future research. (See Appendices A and B for further details on this issue.)
- How to prepare the physical boundary condition? To respect the boundary condition on $\partial\Omega$, the boundary patches Ω_m that touch the physical boundary need to be treated differently. For each sample $\widetilde{\phi}_{m,i}$, the physical boundary

condition is enforced on the set $\partial\Omega_m \cap \partial\Omega$. Random sampling is done only on the remaining part of the patch boundary, that is, $\partial\Omega_m \backslash \partial\Omega$. See Appendices A and B for details.

2.2. Online stage. The online stage finds a particular solution u for given boundary data ϕ, based on information accumulated in the offline stage. This process is carried out through a Schwarz iteration to update local boundary conditions on each patch.

Denote by $\phi^{(n)} = [\dots, \phi_m^{(n)}, \dots]$ the collection of local boundary conditions at the nth iteration, with m being the patch index. At each iteration, we need to obtain $\phi_m^{(n+1)} = \mathcal{P}_m \phi^{(n)}$. For each $m \in J$, let $\phi_{m,i_q^{(n)}}$ be the qth L^2-nearest neighbor of $\phi_m^{(n)}$ in $\mathscr{B}_m$, $q = 1, 2, \dots, k$. These neighbors, supported on $\partial\Omega_m$, lie (approximately) on a local tangential plane centered at $\phi_{m,i_1^{(n)}}$:

$$\Phi_m^{(n)} = \begin{bmatrix} | & & | \\ \phi_{m,i_2^{(n)}} - \phi_{m,i_1^{(n)}} & \dots & \phi_{m,i_k^{(n)}} - \phi_{m,i_1^{(n)}} \\ | & & | \end{bmatrix}. \tag{2.7}$$

Also associated with this plane is our formulation of the solution space centered around $\psi_{m,i_1^{(n)}}$:

$$\Psi_m^{(n)} = \begin{bmatrix} | & & | \\ \psi_{m,i_2^{(n)}} - \psi_{m,i_1^{(n)}} & \dots & \psi_{m,i_k^{(n)}} - \psi_{m,i_1^{(n)}} \\ | & & | \end{bmatrix}. \tag{2.8}$$

Locally, the map between these two planes is approximately linear, and thus to find $\phi_m^{(n+1)} = \mathcal{P}_m \phi^{(n)}$, we look for a linear interpolation of $\phi_m^{(n)}$ on $\Phi_m^{(n)}$ and map this interpolation to $\Psi_m^{(n)}$. More precisely, we look for $c_m^{(n)}$ that solves the least-squares problem

$$c_m^{(n)} = \operatorname{argmin}_{v_m \in \mathbb{R}^{k-1}} \left\| \phi_m^{(n)} - \phi_{m,i_1^{(n)}} - \Phi_m^{(n)} v_m \right\|_{L^2(\partial\Omega_m)}, \tag{2.9}$$

and define the approximate solution to be

$$u_m^{(n)} = \mathcal{S}^\epsilon{}_m \phi_m^{(n)} \approx \psi_{m,i_1^{(n)}} + \Psi_m^{(n)} c_m^{(n)}. \tag{2.10}$$

In summary, the map $\mathcal{P}_m \phi^{(n)}$ is a composition of $\mathcal{P}_m(\phi_l^{(n)},\, l \in \mathscr{N}(m))$ with $l \in \mathscr{N}(m)$, where

$$\phi_m^{(n+1)} = \mathcal{P}_m(\phi_l^{(n)},\, l \in \mathscr{N}(m)) = \mathcal{I}_m^l \mathcal{S}^\epsilon{}_l \phi_l, \quad \text{on } \partial\Omega_m \cap \Omega_l. \tag{2.11}$$

Once a preset error tolerance is achieved, at some step n (usually because the local boundary condition barely changes), the global solution is patched up as follows:

$$u^{(n)} = \sum_{m \in J} \chi_m u_m^{(n)}, \tag{2.12}$$

where $u_m^{(n)}$ is the local solution (2.10) and $\chi_m : \Omega \to \mathbb{R}$ is the smooth partition of unity associated with the partition.

We summarize the procedure in Algorithm 2.1.

Algorithm 2.1 Multiscale solver for nonlinear homogenizable equations (2.1).

Given the radius R_m, the number of nearest neighbors k, the tolerance δ, and the initial guess of boundary conditions $\phi_m^{(0)}$ on each patch $m \in J$.
Domain Decomposition:
  Decompose Ω into overlapping patches: $\Omega = \bigcup_{m\in J} \Omega_m$, and enlarge each patch to obtain $\widetilde{\Omega}_m$.
Offline Stage: Prepare local dictionaries on interior patches Ω_m.
  Step 1: For each $m \in J_i$, generate N samples $\widetilde{\phi}_{m,i}$ from $B(R_m; \widetilde{\mathcal{X}}_m)$;
  Step 2: For all i, call function

$$\widetilde{u}_{m,i} = \text{LocPDESol}(\widetilde{\Omega}_m, \widetilde{\phi}_{m,i})\,;$$

  Step 3: Collect local dictionaries according to (2.6) for $\mathscr{B}_m$ and $\mathscr{I}_m$.
Online Stage: Schwarz iteration.
  while $\sum_m \|\phi_m^{(n)} - \phi_m^{(n-1)}\|_{L^2(\partial\Omega_m)} \geq \delta$ **do**
    for $m \in J$ **do**
      Search for k-nearest neighbors of $\phi_m^{(n)}$ in $\mathscr{B}_m$;
      Solve $c_m^{(n)}$ from the least-squares problem (2.9);
      Update $\phi_m^{(n+1)}$ by (2.11).
    end for
    $n \leftarrow n+1$
  end while
return Global solution $u^{(n)}$ defined by (2.12).

function LocPDESol (Local domain Ω_m, Boundary condition ϕ_m)
  Perform the standard finite difference or finite element methods to solve the local nonlinear equation (2.5);
return Local solution u_m
end function

Remark 2.2. The Johnson–Lindenstrauss lemma [65] indicates that the search for d-dimensional k nearest neighbors in a data set of size N, with distance error δ, can be done in query time $O\Big(kd\frac{\log N}{\delta^2}\Big)$ and storage cost $N^{O(\log(1/\delta)/\delta^2)} + O\Big(d\big(N + \frac{\log N}{\delta^2}\big)\Big)$ [11, 62]. In addition, a cost of $O(k^2 d)$ is incurred at each iteration, due to L^2 minimization for each patch via QR factorization. In our setting, d is equal to the degrees of freedom on the boundary $\partial\Omega_m$.

Remark 2.3. To avoid notational complexity, the discussion above does not consider the physical boundary $\partial\Omega$. If a patch contains part of $\partial\Omega$, then that particular section of the patch is not updated. The true boundary condition ϕ is enforced in every iteration. The derivation is straightforward and is omitted from the discussion.

3. Example 1: Semilinear elliptic equations with highly oscillatory media. In this section, we apply the methodology described above to solve semilinear elliptic equations. Semilinear elliptic equations with multiscale structures arise in a variety of situations, for instance, in nonlinear diffusion generated by nonlinear sources

[66] and in the gravitational equilibrium of stars [24, 75]. As fundamental models in many areas of physics and engineering, these equations have received considerable attention.

We consider the equation

$$\begin{cases} -\nabla_x \cdot \left(A\left(x, \frac{x}{\epsilon}\right) \nabla_x u^\epsilon\right) + f(u^\epsilon) = 0, & x \in \Omega\,, \\ u^\epsilon(x) = \phi(x), & x \in \partial\Omega\,. \end{cases} \tag{3.1}$$

The physical domain is $\Omega \subset \mathbb{R}^d$ with $d \geq 1$, and the Dirichlet boundary condition is given as $\phi(x)$. The permeability $A(x,y) = (a_{ij}(x,y))_{d\times d} : \Omega \times \mathbb{R}^d \to \mathbb{R}^{d\times d}$ depends on both the slow variable x and the fast variable $y = x/\epsilon$ and is highly oscillatory. The function $f : \mathbb{R} \to \mathbb{R}$ describes the nonlinear source term. The solution u^ϵ presents one component in a chemical reaction or one species of a biological system.

The well-posedness of equation (3.1) is classical. We assume that the permeability A is a symmetric matrix with L^∞-coefficients satisfying the standard coercivity condition, and that the nonlinear function f is locally Lipschitz continuous and increasing. Then, assuming the boundary $\partial\Omega$ is smooth enough, given boundary condition $\phi \in H^{1/2}(\partial\Omega) \cap L^\infty(\partial\Omega)$, the problem (3.1) has a unique H^1-solution satisfying the maximum principle. We refer the reader to [30, 49] for details.

3.1. Homogenization limit. The semilinear elliptic equation (3.1) has a homogenization limit as $\epsilon \to 0$. We suppose that $A(x,y)$ is smooth and periodic in y with period $I = [0,1]^d$; then as $\epsilon \to 0$, the solution u^ϵ converges to a limit u^* that satisfies the same class of semilinear elliptic equations with an ϵ-independent effective permeability $A^*(x) = (a^*_{ij}(x))_{d\times d}$:

$$\begin{cases} -\nabla_x \cdot (A^*(x)\nabla_x u^*) + f(u^*) = 0, & x \in \Omega\,, \\ u^*(x) = \phi(x), & x \in \partial\Omega\,. \end{cases} \tag{3.2}$$

This equation (in particular, the effective permeability $A^*(x)$) can be derived by expanding equation (3.1) into different orders of ϵ. Rigorous proofs are given in [20, 22, 80]. We cite the following theorem as a reference.

THEOREM 3.1 (section 16.3 in Chapter 1 of [22]; see also [7]). *Assume the boundary $\partial\Omega$ is smooth. Given $\phi(x) \in H^{1/2}(\partial\Omega) \cap L^\infty(\partial\Omega)$, let u^ϵ be the unique solution to the semilinear elliptic equation* (3.1) *in $H^1(\Omega) \cap L^\infty(\Omega)$. Assume that the permeability $A(x,y)$ is periodic in y with period $I = [0,1]^d$ and that $A(x,\cdot) \in C^1(I)$. Then the solution u^ϵ converges weakly in $H^1(\Omega)$ as $\epsilon \to 0$ to u^* (the solution to* (3.2)*), where the permeability $A^*(x) = (a^*_{ij}(x))_{d\times d}$ is defined by*

$$a^*_{ij}(x) = \int_I \sum_{k,l} a_{kl}(x,y)(\delta_{ki} + \partial_{y_k}\chi_i)(\delta_{lj} + \partial_{y_l}\chi_j)\mathrm{d}y\,. \tag{3.3}$$

Here, for each fixed coordinate $j = 1, 2, \ldots, d$, the function $\chi_j(x,y)$ is the solution of the following cell problem with periodic boundary condition on I:

$$\nabla_y \cdot (A(x,y)\nabla_y(\chi_j(x,y) + y_j)) = 0\,. \tag{3.4}$$

To solve (3.1), the discretization has to resolve ϵ, but in the limit (3.2), the discretization is independent of ϵ. This suggests significant opportunities for cost savings: The information contained in $O(1/\epsilon)$ degrees of freedom can be expressed with $O(1)$ degrees of freedom.

The literature for numerical homogenization is rich, particularly for the linear setting when $f = 0$. Relevant approaches include the multiscale finite element method (MsFEM) [44, 58, 59], the heterogeneous multiscale method (HMM) [4, 39, 40], the generalized finite element method [12, 13], upscaling based on harmonic coordinates [79], elliptic solvers based on $\mathcal{H}$-matrices [18, 53], the reduced basis method [2, 3], the use of localization [77], and the methods based on random SVD [26, 27, 28], to name a few. The analytical understanding of the homogenized equation is essential in the construction of these methods [7]. When randomness presents, one can also look for low-dimensional representation of the solutions in the random space [32, 56, 57, 74].

The literature for nonlinear problems is not as rich. There are several works on quasilinear problems, all of which can be seen as extensions of classical methods, including the MsFEM [29, 42, 43], the HMM [5, 40], the generalized finite element method [41], the local orthogonal decomposition method [55], the reduced basis method [3], and the nonlocal multicontinuum approach [31]. These solvers must be designed carefully for specific nonlinear equations. By contrast, our method makes use of the low-rankness of the solution sets and could be applied with minor modification to different equations.

3.2. Low dimensionality of the tangent space. We now study the structure of the tangent space of the solution manifold, verifying in particular the low dimension assumption. We choose some point $\overline{u}^\epsilon$ on the solution manifold and then randomly pick a neighboring solution point u^ϵ. These two points are solutions to (3.1) computed from distinct nearby boundary configurations $\overline{\phi}$ and ϕ, that is,

$$\overline{u}^\epsilon|_{\partial\Omega} = \overline{\phi}\,, \quad u^\epsilon|_{\partial\Omega} = \phi\,, \quad \text{with} \quad \|\overline{\phi} - \phi\|_{L^\infty(\partial\Omega)} = \mathcal{O}(\delta)\,. \tag{3.5}$$

By varying ϕ around $\overline{\phi}$, one can build a small point cloud around $\overline{u}^\epsilon$. Denoting $\delta u^\epsilon := u^\epsilon - \overline{u}^\epsilon$, we have immediately that

$$\begin{cases} -\nabla_x \cdot \left(A\left(x, \frac{x}{\epsilon}\right) \nabla_x \delta u^\epsilon\right) + f(\overline{u}^\epsilon + \delta u^\epsilon) - f(\overline{u}^\epsilon) = 0, & x \in \Omega\,, \\ \delta u^\epsilon(x) = \phi(x) - \overline{\phi}(x), & x \in \partial\Omega\,. \end{cases} \tag{3.6}$$

In the small-δ regime, this collection of solution differences δu^ϵ spans the tangent plane. We claim this tangent plane is low-dimensional, so that it inherits the homogenization effect of the original equation. We have the following result.

THEOREM 3.2. *Let δu^ϵ solve* (3.6). *Assume $A(x, y) = (a_{ij}(x, y))_{d\times d}$ is periodic in y with period $I = [0, 1]^d$. The equation has homogenization limit when $\epsilon \to 0$, meaning there exists a limiting permeability $A^*(x) = (a^*_{ij}(x))_{d\times d}$, determined by $A(x, y)$ via* (3.3) *and* (3.4), *so that $\delta u^\epsilon \to \delta u^*$ and δu^* solves*

$$\begin{cases} -\nabla_x \cdot (A^*(x)\nabla_x \delta u^*) + f(\overline{u}^* + \delta u^*) - f(\overline{u}^*) = 0, & x \in \Omega\,, \\ \delta u^*(x) = \phi(x) - \overline{\phi}(x), & x \in \partial\Omega\,, \end{cases} \tag{3.7}$$

where $\overline{u}^*$ solves

$$\begin{cases} -\nabla_x \cdot (A^*(x)\nabla_x \overline{u}^*) + f(\overline{u}^*) = 0, & x \in \Omega\,, \\ u^*(x) = \overline{\phi}(x), & x \in \partial\Omega\,. \end{cases} \tag{3.8}$$

Further, for small δ, (3.7), *in the leading order of δ, becomes*

$$-\nabla_x \cdot (A^*(x)\nabla_x \delta u^*) + f'(\overline{u}^*(x))\delta u^* = 0\,. \tag{3.9}$$

Proof. By applying Theorem 3.1 to the equation for $\overline{u}^\epsilon$, which is

$$\begin{cases} -\nabla_x \cdot \left(A\left(x, \frac{x}{\epsilon}\right) \nabla_x \delta\overline{u}^\epsilon\right) + f(\overline{u}^\epsilon) = 0, & x \in \Omega\,, \\ \delta\overline{u}^\epsilon(x) = \overline{\phi}(x), & x \in \partial\Omega\,, \end{cases}$$

we have, by comparing with equation (3.1) for u^ϵ, that $\overline{u}^\epsilon$ converges weakly to $\overline{u}^*$, which solves (3.8), and that u^ϵ converges weakly to u^*, which solves (3.2). From the definition $\delta u^\epsilon = u^\epsilon - \overline{u}^\epsilon$, we find that δu^ϵ converges to δu^*, which solves (3.7). ∎

This theorem suggests that for the discretized equation, because of the existence of the homogenized limit, the tangent plane of the discrete solution is approximately low-rank. The space spanned by $\{\delta u^\epsilon\}$ can be approximately spanned by $\{\delta u^*\}$, which solves the limiting equation (3.7) without dependence on small scales.

3.3. Implementation. We apply Algorithm 2.1 to equation (3.1) with $f(u) = u^3$ and $\Omega = [0, L]^2 \subset \mathbb{R}^2$, that is,

$$\begin{cases} -\nabla_x \cdot \left(a\left(x, \frac{x}{\epsilon}\right) \nabla_x u\right) + u^3 = 0\,, & x \in \Omega = [0, L]^2, \\ u(x) = \phi(x), & x \in \partial\Omega. \end{cases} \tag{3.10}$$

We use the domain decomposition strategy of section 2 to solve this system. Since Ω is convex and the coefficient $a(x, x/\epsilon)$ belongs to $L^\infty(\Omega)$, we can show, using the monotone method [9, 30], that the equation is well-posed, having a unique solution if we set

$$\mathcal{X} = H^{1/2}(\partial\Omega) \cap L^\infty(\partial\Omega), \quad \mathcal{Y} = H^1(\Omega) \cap L^\infty(\Omega)\,.$$

In the offline stage, we generate N samples for each enlarged patch $\widetilde{\Omega}_m$, as follows:

$$\widetilde{\phi}_{m,i} \in B(R_m;\, \widetilde{\mathcal{X}}_m)\,, \quad i = 1, \ldots, N\,.$$

(The measure we use for sampling is discussed in Appendix A.) We equip the ball with the $H^{1/2}$-norm:

$$B(R_m;\, \widetilde{\mathcal{X}}_m) = \{\widetilde{\phi} \in \widetilde{\mathcal{X}}_m : \|\widetilde{\phi}\|_{H^{1/2}(\partial\Omega_m)} \le R_m\}\,. \tag{3.11}$$

We compute the $H^{1/2}(\partial\Omega)$-norm numerically using the Gagliardo seminorm [35]:

$$\|\phi\|_{H^{1/2}(\partial\Omega)} = \sqrt{\int_{\partial\Omega} |\phi(x)|^2 \mathrm{d}x + \iint_{\partial\Omega\times\partial\Omega} \frac{|\phi(x) - \phi(y)|^2}{|x - y|^2} \mathrm{d}x\mathrm{d}y}\,.$$

For these boundary configurations, we solve the equation

$$\begin{cases} -\nabla_x \cdot \left(a\left(x, \frac{x}{\epsilon}\right) \nabla_x \widetilde{u}_{m,i}\right) + \widetilde{u}_{m,i}^3 = 0\,, & x \in \widetilde{\Omega}_m\,, \\ \widetilde{u}_{m,i}(x) = \widetilde{\phi}_{m,i}(x)\,, & x \in \partial\widetilde{\Omega}_m\,, \end{cases} \tag{3.12}$$

and build two sets of dictionaries by confining the solutions in the interior Ω_m and the boundary $\partial\Omega_m$ as follows:

$$\mathscr{I}_m = \{\psi_{m,i} = \widetilde{u}_{m,i}|_{\Omega_m}\}_{i=1}^N, \quad \mathscr{B}_m = \{\phi_{m,i} = \widetilde{u}_{m,i}|_{\partial\Omega_m}\}_{i=1}^N\,. \tag{3.13}$$

In the online stage, local boundary conditions are updated according to (2.10) at each iteration, with coefficients computed from (2.9). The local tangent space is found by

searching for the k-nearest neighbors in the dictionary $\mathscr{B}_m$, mapped to the dictionary $\mathscr{I}_m$ (see (3.13)). We use the L^2-norm to measure the distance between the newly generated solutions and the older solution set.

Once a preset error tolerance is achieved (at step n, say), the global solution is patched up from the local pieces, as follows:

$$u^{(n)} = \sum_{m\in J} \chi_m u_m^{(n)}\,, \tag{3.14}$$

where $u_m^{(n)}$ is the local solution on Ω_m at the nth step, and $\chi_m : \Omega \to \mathbb{R}$ is a smooth partition of unity.

3.4. Numerical tests. We present numerical results for (3.10) in this subsection. We use $L = 1$, yielding the domain $\Omega = [0,1]^2$, and define the oscillatory media as follows:

$$a(x,y,x/\epsilon,y/\epsilon) = 2 + \sin(2\pi x)\cos(2\pi y) + \frac{2+1.8\sin(2\pi x/\epsilon)}{2+1.8\cos(2\pi y/\epsilon)} + \frac{2+\sin(2\pi y/\epsilon)}{2+1.8\cos(2\pi x/\epsilon)}\,.$$

The boundary condition is

$$\begin{aligned} &\phi(x,0) = -\sin(2\pi x)\,, && \phi(x,1) = \sin(2\pi x)\,,\\ &\phi(0,y) = \sin(2\pi y)\,, && \phi(1,y) = -\sin(2\pi y)\,. \end{aligned}$$

To form the partitioning, the whole domain Ω is divided equally into 4×4 nonoverlapping squares, and then each square is enlarged by $\Delta x_{\rm o} = .0625$ on the sides that do not intersect with $\partial\Omega$ to create overlap. We thus have $M_1 = M_2 = 4$, with Ω_m for $m = (m_1, m_2)$, $m_1 = 1,2,3,4$ and $m_2 = 1,2,3,4$, defined by

$$\begin{aligned} \Omega_m = &\left[\max\left(\tfrac{m_1-1}{M_1} - \Delta x_{\rm o}, 0\right), \min\left(\tfrac{m_1}{M_1} + \Delta x_{\rm o}, 1\right)\right]\\ &\times \left[\max\left(\tfrac{m_2-1}{M_2} - \Delta x_{\rm o}, 0\right), \min\left(\tfrac{m_2}{M_2} + \Delta x_{\rm o}, 1\right)\right], \quad m = (m_1,m_2) \in J\,. \end{aligned}$$

Denote $\Omega_m = [x_m^{(1)}, x_m^{(2)}] \times [y_m^{(1)}, y_m^{(2)}]$. The partition of unity function χ_m is defined by normalizing the bump functions on the overlapping domains. More precisely, we first define a bump function $f_m : \Omega \to \mathbb{R}$ supported on Ω_m as follows:

$$f_m(x,y) = \begin{cases} \exp\left(-\frac{1}{1-|x-x_m|/\alpha_m} - \frac{1}{1-|y-y_m|/\beta_m}\right), & (x,y)\in\Omega_m,\\ 0 \quad \text{otherwise}, \end{cases}$$

where $x_m = \frac{x_m^{(2)}-x_m^{(1)}}{2}$, $y_m = \frac{y_m^{(2)}-y_m^{(1)}}{2}$, $\alpha_m = \frac{x_m^{(1)}+x_m^{(2)}}{2}$, and $\beta_m = \frac{y_m^{(1)}+y_m^{(2)}}{2}$. The partition of unity $\chi_m : \Omega \to \mathbb{R}$ is then obtained by

$$\chi_m(x,y) = \frac{f_m(x,y)}{\sum_{m\in J} f_m(x,y)}\,.$$

A standard finite-volume scheme with uniform grid is used for discretization, the corresponding nonlinear discrete system being solved by Newton's method. The reference solutions are computed on the fine mesh with $h = 2^{-12} = \frac{1}{4096}$. Unless otherwise

specified, other computations are performed with mesh size $h = 2^{-9} = \frac{1}{512}$. Denoting the numerical solution by $u_{ij} \approx u(x_i, y_j)$, we use the classical discrete L^2-norm,

$$\|u\|_{L^2} = h\sqrt{\sum_{i,j=0}^{p} |u_{ij}|^2},$$

and the energy norm,

$$\|u\|_{\mathcal{E}} = h\sqrt{\sum_{j=0}^{p}\sum_{i=0}^{p-1} a_{i+1/2,j}\left|\frac{u_{i+1,j}-u_{ij}}{h}\right|^2 + \sum_{i=0}^{p}\sum_{j=0}^{p-1} a_{i,j+1/2}\left|\frac{u_{i,j+1}-u_{ij}}{h}\right|^2},$$

and define the relative errors accordingly by

$$\text{relative } L^2 \text{ error } = \frac{\|u_{\text{ref}} - u_{\text{approx}}\|_{L^2}}{\|u_{\text{ref}}\|_{L^2}}, \quad \text{relative energy error} = \frac{\|u_{\text{ref}} - u_{\text{approx}}\|_{\mathcal{E}}}{\|u_{\text{ref}}\|_{\mathcal{E}}}.$$

We first describe numerical experience with the offline stage. Each interior patch Ω_m is enlarged by a margin Δx_{b} to dampen the boundary effects. The resulting buffered patch $\widetilde{\Omega}_m$ is concentric with Ω_m; see Figure 2. In the plots shown below, we study the patch indexed by $m = (2, 2)$.

To build the local dictionary, we generate 64 samples randomly in $B(R_{2,2}, \widetilde{X}_{2,2})$, where $R_{2,2} = 20$. (The sampling scheme is discussed in Appendix A.) We compute the local solutions with these boundary conditions on $\widetilde{\Omega}_{2,2}$, for several choices of buffer size Δx_{b}, and subtract the solutions from the reference solution, confined to $\Omega_{2,2}$. This procedure forms the tangent space centered around the reference solution in this particular patch. In Figure 3a we plot the singular value decay of this tangent space for $\epsilon = 2^{-4}$. It is clear that the singular values decay exponentially, with a larger buffer margin Δx_{b} leading to a faster decay rate. This observation suggests that the tangent space is approximately low dimensional. We then project the reference solution onto the space spanned by its closest neighbors. As the number of neighbors increases, the relative error decays exponentially, as seen in Figure 3b. When the buffer margin is $\Delta x_{\text{b}} = 2^{-4}$, we achieve 99% accuracy with 30 neighbors. By comparison, the degrees of freedom for this patch are determined by the total number of grid points on the boundary of this patch — 768 in this particular case.

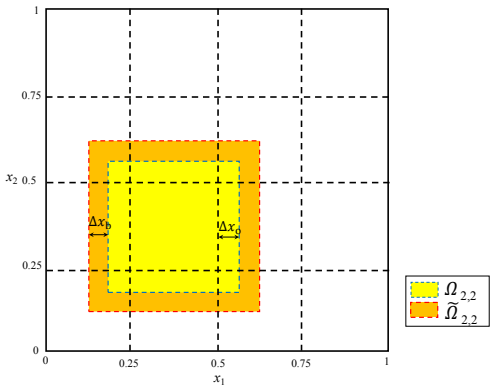


FIG. 2. *Buffered domain decomposition.*

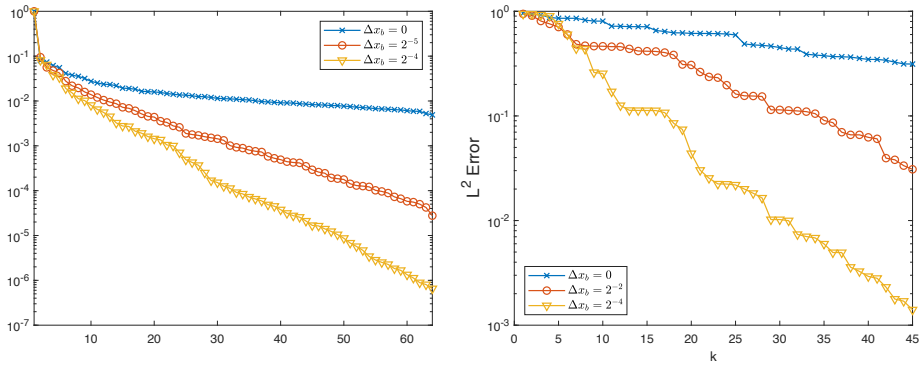


FIG. 3. (a) *The singular value decay of the tangent space (centered around the reference solution) on patch* $\Omega_{2,2}$ *for different values of the buffer margin* Δx_b. (b) *The relative error of the projection of the reference solution onto the space spanned by the nearest* k *neighbors on* $\Omega_{2,2}$. *The distance is measured in* $L^2(\Omega_{2,2})$. $\epsilon = 2^{-4}$ *in both plots.*

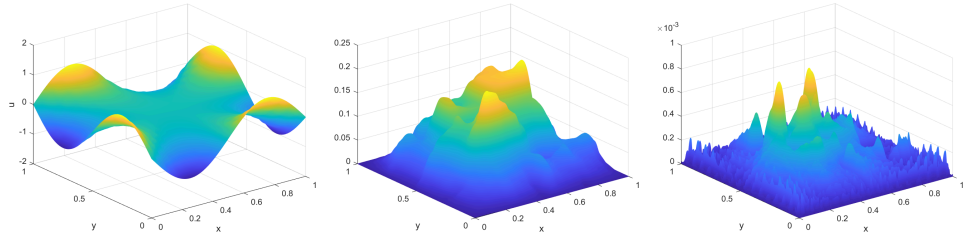


FIG. 4. *Computed solutions. Left panel shows the reference solution obtained with fine grids of width* $h = 2^{-12}$. *Middle and right panels show the numerical error* $|u - u_{ref}|$ *obtained with* $k = 5$ *and* $k = 30$, *respectively.*

In the online stage, we set the stopping criterion to be

$$\sum_m \|\phi_m^{(n)} - \phi_m^{(n-1)}\|_{L^2(\partial\Omega_m)} < 10^{-5},$$

where the upper index (n) indicates the evaluation of the solution in the nth iteration on $\mathcal{X}_m$, which is the boundary of Ω_m. The initial guesses for all local boundary conditions are chosen (trivially) to be $\phi_m^{(0)}|_{\partial\Omega_m \setminus \partial\Omega} = 0$ and $\phi_m^{(0)}|_{\partial\Omega_m \cap \partial\Omega} = \phi|_{\partial\Omega_m \cap \partial\Omega}$.

In Figure 4, we compare the numerical solutions using the space spanned by $k = 5$ and $k = 40$ nearest neighbors. The buffer margin is $\Delta x_{\mathrm{b}} = 2^{-4}$, and we set $\epsilon = 2^{-4}$. We also document the error behavior as a function of k, ϵ, and Δx_{b}. In Figure 5, we plot the error decay as a function of k (the number of neighbors used in the online stage) for different values of ϵ and Δx_{b}. The decay is independent of ϵ, indicating the rank structure is not influenced by small scales in the equation. As the number of neighbors k increases, the global relative L^2 and energy errors decay exponentially provided a buffer zone is present. When $\Delta x_{\mathrm{b}} = 0$ (no buffer), the boundary layer effect is strong, and convergence is not obtained, meaning that the local solution cannot be well approximated from the dictionary.

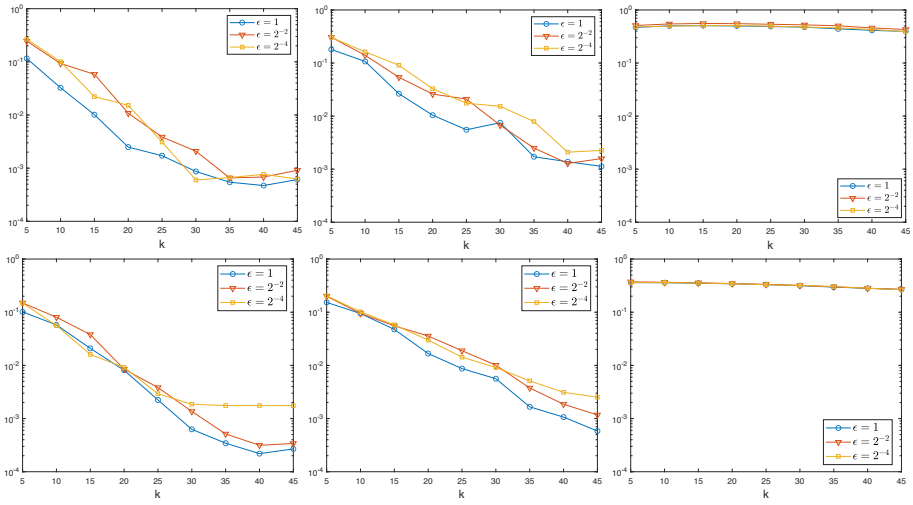


FIG. 5. *The top row of plots shows the global L^2 error as a function of k with different ϵ and buffer zone size Δx_b. The bottom row of plots shows the global energy error. The three columns of plots represent $\Delta x_b = 2^{-4}, 2^{-5}, 0$, respectively.*

TABLE 1

CPU time comparison between our reduced method with $k = 5, 10, 20, 30, 40$ and classical Schwarz method.

	CPU Time (s) ($\epsilon = 2^{-4}$)	
	Offline	Online
Reduced model $k = 5$	135.6914	0.173712
Reduced model $k = 10$		0.305707
Reduced model $k = 20$		0.462857
Reduced model $k = 30$		0.696785
Reduced model $k = 40$		1.124082
Classical Schwarz	—	187.7705

We show CPU times in Table 1, comparing the reduced model for different values of k with the classical Schwarz iteration for $\epsilon = 2^{-4}$ and $\Delta x_{\mathrm{b}} = 2^{-4}$. The same stopping criterion is used for all variants. The online stage of each reduced model is significantly faster than the classical Schwarz iteration. Even with $k = 40$ neighbors involved in the local solution reconstruction, our method requires 1.12s, compared to 187.8s required by the classical Schwarz method. While the offline preparation is expensive in general, it is still cheaper in this example than the classical Schwarz iteration for solving a single problem. Because the dictionary can be reused, our method has a strong advantage in situations where many solutions corresponding to different boundary conditions are needed. This is a typical situation in inverse problems, where in order to determine the unknown media, many boundary configurations are imposed and numerical solutions are computed to compare with measurements [25].

4. Example 2: Nonlinear radiative transfer equation. Here we study the application of Algorithm 2.1 to a nonlinear radiative transfer equation. Radiative transfer is the physical phenomenon of energy transfer in the form of electromagnetic

radiation, and the radiative transfer equations describe the absorption or scattering of radiation as it propagates through a medium. The equations are important in optics, astrophysics, atmospheric science, remote sensing [78], and other applications.

We denote by $I^\epsilon(x,v)$ the distribution function of photon particles at location x moving with velocity v in the physical domain $\mathcal{D} \subset \mathbb{R}^3$ and the velocity domain $\mathcal{V} = \mathbb{S}^2$. Also denote by $T^\epsilon(x)$ the temperature profile across domain $\mathcal{D}$. We consider a nonlinear system of equations that couples the photon particle distribution with the temperature profile. The steady state equations are

$$\begin{cases} \epsilon v \cdot \nabla_x I^\epsilon = B(T^\epsilon) - I^\epsilon \ \text{ for } (x,v) \in \mathcal{K} = \mathcal{D} \times \mathcal{V}, \\ \epsilon^2 \Delta_x T^\epsilon = B(T^\epsilon) - \langle I^\epsilon \rangle \ \text{ for } x \in \mathcal{D}\,, \end{cases} \tag{4.1}$$

with the velocity-averaged intensity given by

$$\langle I \rangle(x) = \int_{\mathcal{V}} I(x,v) \mathrm{d}\mu(v)\,. \tag{4.2}$$

Here, $\mu(v)$ is a normalized uniform measure on $\mathcal{V}$, and $B(T)$ is a nonlinear function of T, typically defined as

$$B(T) = \sigma T^4, \tag{4.3}$$

where σ is a scattering coefficient [70, 86]. The parameter ϵ is called the Knudsen number, standing for the ratio of the mean free path and the typical domain length. When the medium is highly scattering and optically thick, the mean free path is small, with $\epsilon \ll 1$. The scattering coefficient σ is independent of ϵ.

We consider a slab geometry. We assume the y and z directions to be homogeneous; then since $v = (\cos\theta, \sin\theta\sin\varphi, \sin\theta\cos\varphi)$, the v_x component becomes $\cos\theta \in [-1,1]$. The problem is simplified to

$$\begin{cases} \epsilon v \partial_x I^\epsilon = B(T^\epsilon) - I^\epsilon \\ \epsilon^2 \partial_x^2 T^\epsilon = B(T^\epsilon) - \langle I^\epsilon \rangle \end{cases}, \quad (x,v) \in \mathcal{K} = [a,b] \times [-1,1]\,, \tag{4.4}$$

with $\langle I \rangle(x) = \frac{1}{2}\int_{-1}^{1} I(x,v)\mathrm{d}v$.

We provide incoming boundary conditions that specify the distribution of photons entering the domain. The boundary condition itself has no ϵ-dependence; we have

$$I^\epsilon(x,v) = I_b(x,v) \quad \text{on } \Gamma_-\,, \quad T^\epsilon(x) = T_b(x) \quad \text{on } \partial\mathcal{D}\,. \tag{4.5}$$

Here $\Gamma_\pm$ collect the coordinates at the boundary with velocity pointing into or out of the domain,

$$\Gamma_\pm = \{(x,v) : x \in \partial\mathcal{D}\,, \pm v \cdot n_x > 0\}\,,$$

and n_x denotes the unit outer normal vector at $x \in \partial\Omega$.

4.1. Homogenization limit. Equations (4.1) have a homogenization limit. As $\epsilon \to 0$, the right-hand side of the equations dominates, and by balancing the scales we obtain

$$I^\epsilon \sim \langle I^\epsilon \rangle \sim \sigma (T^\epsilon)^4 \sim I^* \sim \sigma (T^*)^4\,.$$

To find the equation satisfied by T^*, we expand equations (4.1) up to second order in ϵ. Rigorous results are shown in [15, 16, 69, 72]. We cite the following theorem, which captures the results needed here.

THEOREM 4.1 (modification of Theorem 3.2 in [69]). *Let $\mathcal{D} \subset \mathbb{R}^3$ be bounded, and let $\partial\mathcal{D}$ be smooth. We assume that the boundary conditions* (4.5) *are positive and that $T_b \in H^{1/2}(\partial\mathcal{D}) \cap L^\infty(\partial\mathcal{D})$ and $I_b \in L^\infty(\Gamma_-)$; then the nonlinear radiative transfer equation* (4.1) *has a unique positive solution $(I^\epsilon, T^\epsilon) \in L^\infty(\mathcal{K}) \times L^\infty(\mathcal{D})$. If we assume further that $(I_b, T_b) \geq \gamma > 0$ and $I_b = B(T_b)$ a.e. on Γ_-, then the solution in the limit as $\epsilon \to 0$ converges weakly to $(B(T^*), T^*)$, where the limiting temperature T^* is the unique positive solution to the following PDE:*

$$\Delta_x(T^* + B(T^*)/3) = 0 \quad \text{for} \quad x \in \mathcal{D}\,, \tag{4.6}$$

equipped with Dirichlet boundary data $T^*|_{\partial\mathcal{D}} = T_b$. The convergence of T^ϵ is in $H^1(\mathcal{D})$ weak, and the convergence of I^ϵ is in $L^\infty(\mathcal{K})$ weak-$*$.

Remark 4.2. Without appropriate boundary conditions $I_b = B(T_b)$, boundary layers of width $O(\epsilon)$ may emerge as $\epsilon \to 0$. It is conjectured in [69] that the boundary layers in the neighborhood of each point $\hat{x} \in \partial\mathcal{D}$ can be characterized by the following one-dimensional Milne problem for $y \in [0, \infty)$:

$$\begin{aligned} -(v \cdot n_{\hat{x}})\partial_y \hat{I} &= B(\hat{T}) - \hat{I}\,, \\ \partial_y^2 \hat{T} &= B(\hat{T}) - \langle \hat{I} \rangle\,, \\ \hat{T}(0) = T_b(\hat{x}), \quad \hat{I}(0, v) &= I_b(\hat{x}, v) \ \text{ for } v \cdot n_{\hat{x}} < 0\,, \end{aligned}$$

where $y = \frac{(x-\hat{x})\cdot n_{\hat{x}}}{\epsilon}$ represents a rescaling of the layer. The solutions that are bounded at infinity are used to form the Dirichlet boundary conditions for (4.6): At the limit as $y \to \infty$, $B(\hat{T}) = \langle \hat{I} \rangle = \hat{I}$, and one uses $T(\hat{x}) = \hat{T}(\infty)$.

According to Theorem 4.1, in the zero limit of ϵ, I^ϵ loses its velocity dependence and is proportional to $(T^\epsilon)^4$ that satisfies a semilinear elliptic equation. Since the information in the velocity domain is lost, we expect low dimensionality of the (discretized) solution set. For the slab problem for RTE (radiative transfer equation) (4.4), the number of grid points needed for a satisfactory numerical result is $N_x N_v$, with both N_x and N_v scaling as $O(\frac{1}{\epsilon})$ for numerical accuracy. Thus, for every given configuration of boundary conditions, the numerical solution is one data point in an $N_x N_v$-dimensional space — a space of very high dimension. However, when ϵ is small, the solutions are approximately given by the limiting elliptic equation (4.6), and the number of grid points needed is a number N_x^* that has no dependence on ϵ. This implies that the point clouds in the $O(1/\epsilon^2)$-dimensional space can be essentially represented using $O(1)$ degrees of freedom: The solution manifold is approximately low dimensional. (Savings are even greater for problems with higher physical/ velocity dimensions.)

The use of a limiting equation to speed up the computation of kinetic equations is not new. For Boltzmann-type equations (for which RTE serves as a typical example), one is interested in designing algorithms that automatically reconstruct the limiting solutions with low computational cost. The algorithms that achieve this property are called "asymptotic-preserving" (AP) methods [34, 36, 37, 46, 47, 60, 63, 64, 71, 73], because the asymptotic limits are preserved automatically. There are many successful examples of AP schemes, but most of them depend strongly on the analytical understanding of the limiting equation. The solver of the limiting equation is built

into the Boltzmann solver in order to drag the numerical solution to its macroscopic description. Such a design scheme limits the application of AP methods significantly. Many kinetic equations have unknown limiting behavior, making the use of AP designs impossible. By contrast, Algorithm 2.1 does not rely on any explicit information about the limiting equation and is able to deal with general kinetic equations with small scales.

4.2. Low dimensionality of the tangent space. As for the example of section 3, we start by studying some basic properties of the local solution manifold and its tangential plane.

We first randomly pick a point $(\overline{I}^\epsilon, \overline{T}^\epsilon)$ on the solution manifold around which to perform tangential approximation. Nearby points (I^ϵ, T^ϵ) are obtained by solutions to the RTE (4.4) with respect to perturbed boundary conditions. The boundary conditions for $(\overline{I}^\epsilon, \overline{T}^\epsilon)$ and (I^ϵ, T^ϵ), respectively, are

$$(\overline{I}^\epsilon|_{\Gamma_-}, \overline{T}^\epsilon|_{\partial\mathcal{D}}) = (\overline{I}_b, \overline{T}_b), \quad (I^\epsilon|_{\Gamma_-}, T^\epsilon|_{\partial\mathcal{D}}) = (I_b, T_b), \tag{4.7}$$

and we assume close proximity, in the sense that

$$\|\overline{I}_b - I_b\|_{L^2(\Gamma_{m,-})} + \|\overline{T}_b - T_b\|_2 = O(\delta). \tag{4.8}$$

Using the notation $\delta I^\epsilon := I^\epsilon - \overline{I}^\epsilon$ and $\delta T^\epsilon := T^\epsilon - \overline{T}^\epsilon$ for the difference of the two solutions, we find that this difference satisfies the equations

$$\begin{cases} \epsilon v \partial_x \delta I^\epsilon = B(\overline{T}^\epsilon + \delta T^\epsilon) - B(\overline{T}^\epsilon) - \delta I^\epsilon, \\ \epsilon^2 \partial_x^2 \delta T^\epsilon = B(\overline{T}^\epsilon + \delta T^\epsilon) - B(\overline{T}^\epsilon) - \langle \delta I^\epsilon \rangle, \end{cases} \tag{4.9}$$

with boundary conditions

$$\delta I^\epsilon|_{\Gamma_-} = \overline{I}_b - I_b, \quad \delta T^\epsilon|_{\partial\mathcal{D}} = \overline{T}_b - T_b.$$

By varying I_b and T_b (subject to (4.8)), we obtain a list of solutions $(\delta I^\epsilon, \delta T^\epsilon)$ that spans the tangent plane of the solution manifold surrounding $(\overline{I}^\epsilon, \overline{T}^\epsilon)$. It will be shown below that this plane is low dimensional. We have the following result.

THEOREM 4.3. *Let $(\delta I^\epsilon, \delta T^\epsilon)$ solve (4.9). As $\epsilon \to 0$, we have $(\delta I^\epsilon, \delta T^\epsilon) \to (\delta I^*, \delta T^*)$ so that $\delta I^* = \langle \delta I^* \rangle = B(\overline{T}^* + \delta T^*) - B(\overline{T}^*)$, and δT^* solves*

$$\partial_x^2 \left[\delta T^* + \tfrac{1}{3} B(\overline{T}^* + \delta T^*) - \tfrac{1}{3} B(\overline{T}^*) \right] = 0. \tag{4.10}$$

Here the reference state $\overline{T}^*$ solves

$$\partial_x^2 \left[\overline{T}^* + \tfrac{1}{3} B(\overline{T}^*) \right] = 0. \tag{4.11}$$

Both equations are equipped with appropriate Dirichlet-type boundary conditions. Furthermore, for small δ, the leading order equation is

$$\Delta_x \left[\left(1 + \tfrac{1}{3} B'(\overline{T}^*) \right) \delta T^* \right] = 0. \tag{4.12}$$

Proof. Apply Theorem 4.1 (in one dimension) to the equation for $(\overline{I}^\epsilon, \overline{T}^\epsilon)$ to obtain

$$\begin{cases} \epsilon v \partial_x \overline{I}^\epsilon = B(\overline{I}^\epsilon) - \overline{I}^\epsilon, \\ \epsilon^2 \partial_x^2 \overline{T}^\epsilon = B(\overline{I}^\epsilon) - \langle \overline{I}^\epsilon \rangle, \end{cases}$$

and equation (4.4) for (I^ϵ, T^ϵ). Together, these equations show that $(\overline{I}^\epsilon, \overline{T}^\epsilon)$ converges weakly to $(\overline{I}^*, \overline{T}^*)$ which solves (4.11), and also that (I^ϵ, T^ϵ) converges weakly to (I^*, T^*) which solves (4.6). Taking the difference for $(\overline{I}^\epsilon, \overline{T}^\epsilon)$ and (I^ϵ, T^ϵ), we find that $(\delta I^\epsilon, \delta T^\epsilon)$ converges to $(\delta I^*, \delta T^*)$, which solves (4.10). ∎

In one dimension, the elliptic problem only has two degrees of freedom, determined by the two Dirichlet boundary conditions. This suggests that in the limit as $\epsilon \to 0$, for relatively small δ, the tangent plane spanned by $(\delta I^\epsilon, \delta T^\epsilon)$ is asymptotically two-dimensional and is parameterized by the two boundary conditions for δT^ϵ. (A similar reduction holds in higher dimensions, but we leave the implementation to future work.)

4.3. Implementation of the algorithm. In RTE, the domain setup needs some extra care, and we need to re-perform partitioning. The physical boundaries are no longer the boundaries at which the Dirichlet conditions are imposed, and the general framework in section 2 for the PDE with Dirichlet boundary conditions on the physical boundaries has to be changed accordingly. For the $(1+1)$D case, we set

$$\mathcal{K} = \mathcal{D} \times \mathcal{V} = [0, L] \times [-1, 1]\,; \text{ then } \Gamma_- = \{(0, v) : v > 0\} \cup \{(1, v) : v < 0\}\,,$$

with boundary conditions

$$I^\epsilon|_{\Gamma_-} = g = (g^{(1)}(0, \cdot), g^{(2)}(L, \cdot)), \quad T^\epsilon(0) = \theta^{(1)}, \quad T^\epsilon(L) = \theta^{(2)}\,,$$

where $g^{(1)}$ is supported only on $v > 0$, while $g^{(2)}$ is supported only on $v < 0$. For notational simplicity, we write

$$u := (I^\epsilon, T^\epsilon)\,, \quad u|_{\Gamma_-} := \phi = (g^{(1)}(0, \cdot), g^{(2)}(L, \cdot), \theta^{(1)}, \theta^{(2)})\,.$$

To partition the domain, we divide $\mathcal{K}$ into M overlapping patches:

$$\mathcal{K} = \bigcup_{m=1}^{M} \mathcal{K}_m, \quad \text{with} \quad \mathcal{K}_m = \mathcal{D}_m \times \mathcal{V} = [t_m, s_m] \times [-1, 1], \tag{4.13}$$

where t_m and s_m are left and right boundaries for the mth patch, satisfying

$$0 = t_1 < t_2 < s_1 < t_3 < \cdots < s_{M-2} < t_M < s_{M-1} < s_M = L\,.$$

The size of the mth patch in the x direction is denoted as $d_m = t_m - s_m$. For each patch, we define the local incoming boundary coordinates as follows:

$$\Gamma_{m,-} = \{(t_m, v) : v > 0\} \cup \{(s_m, v) : v < 0\}\,. \tag{4.14}$$

See Figure 6 for an illustration of the configuration.

In this particular setup, according to [72], if ϕ is in the space

$$\mathcal{X} = L^2(\Gamma_-) \times \mathbb{R}^2_+ = \left\{ \left(g, \theta^{(1)}, \theta^{(2)}\right) \mid g \in L^2(\Gamma_-);\ \theta^{(1)}, \theta^{(2)} \geq 0 \right\}\,,$$

then there exists a unique positive solution in the space

$$\mathcal{Y} = H^1_2(\mathcal{K}) \times H^1(\mathcal{D}) = \left\{(I, T) \mid I \in H^1_2(\mathcal{K}), T \in H^1(\mathcal{D})\right\}\,,$$

where $H^1_2(\mathcal{K})$ is the space of functions for which the following norm is finite:

$$\|I\|_{H^1_2(\mathcal{K})} = \|I\|_{L^2(\mathcal{K})} + \|v\partial_x I\|_{L^2(\mathcal{K})}\,.$$

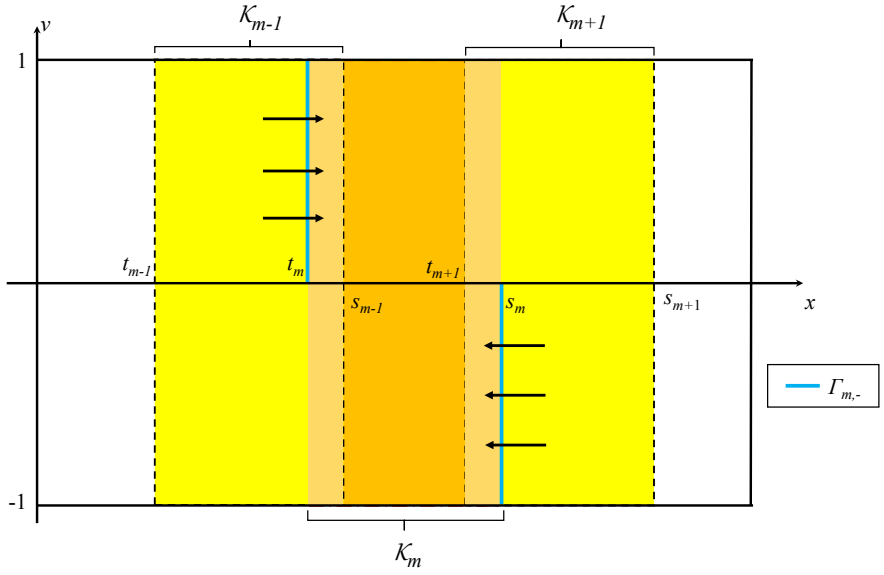


FIG. 6. *Domain decomposition for nonlinear RTE and the incoming boundary of the local patch.*

Note that the trace operators $T_\pm u = u|_{\Gamma_\pm}$ are well-defined maps from $H_2^1(\mathcal{K})$ to $L^2(\Gamma_\pm)$ (see, for example, [6]).

To proceed, we define several operators. We denote spaces associated with each patch m as follows:

$$\mathcal{X}_m := L^2(\Gamma_{m,-}) \times \mathbb{R}_+^2 = \left\{ \left(g, \theta^{(1)}, \theta^{(2)} \right) \mid g \in L^2(\Gamma_{m,-}),\ \theta^{(1)}, \theta^{(2)} \geq 0 \right\},$$
$$\mathcal{Y}_m := H_2^1(\mathcal{K}_m) \times H^1(\mathcal{D}_m) = \left\{ (I, T) \mid I \in H_2^1(\mathcal{K}_m),\ T \in H^1(\mathcal{D}_m) \right\}.$$

Then we have the following operator definitions for each patch m. (For simplicity of notation, we set $\sigma \equiv 1$ in the definition (4.3) of $B(T)$.)

- The solution operator $\mathcal{S}_m : \mathcal{X}_m \to \mathcal{Y}_m$ satisfies $\mathcal{S}_m \phi_m = u_m$, where $u_m = (I_m^\epsilon, T_m^\epsilon)$ solves the RTE on patch $\mathcal{K}_m$ with boundary condition $\phi_m = (g_m, \theta_m^{(1)}, \theta_m^{(1)})$:

$$\begin{cases} \epsilon v \partial_x I_m^\epsilon &= (T_m^\epsilon)^4 - I_m^\epsilon, \\ \epsilon^2 \partial_x^2 T_m^\epsilon &= (T_m^\epsilon)^4 - \langle I_m^\epsilon \rangle, \end{cases} \quad (x, v) \in \mathcal{K}_m,$$

  with $T_m^\epsilon(t_m) = \theta_m^{(1)}$, $T_m^\epsilon(s_m) = \theta_m^{(2)}$, and

$$I_m^\epsilon|_{\Gamma_{m,-}} = g_m(x, v) = (g_m^{(1)}(x, v), g_m^{(2)}(x, v)).$$

- The restriction operator $\mathcal{I}_{m\pm1}^m$ from patch $\mathcal{K}_m$ to the boundaries of adjacent patches, namely, $\mathcal{K}_m \cap \Gamma_{m\pm1,-}$ and $\mathcal{D}_m \cap \partial\mathcal{D}_{m\pm1,-}$, is defined as follows:

$$\mathcal{I}_{m+1}^m u_m = (I_m^\epsilon|_{\mathcal{K}_m \cap \Gamma_{m+1,-}}, T_m^\epsilon|_{\mathcal{D}_m \cap \partial\mathcal{D}_{m+1,-}}), \quad m = 1, \ldots, M-1,$$
$$\mathcal{I}_{m-1}^m u_m = (I_m^\epsilon|_{\mathcal{K}_m \cap \Gamma_{m-1,-}}, T_m^\epsilon|_{\mathcal{D}_m \cap \partial\mathcal{D}_{m-1,-}}), \quad m = 2, \ldots, M.$$

- The boundary update operator $\mathcal{P}_m : \mathcal{X}_{m-1} \oplus \mathcal{X}_{m+1} \to \mathcal{X}_m$ is defined for $m \neq 1$ and $m \neq M$ by

$$\mathcal{P}_m(\phi_{m-1}, \phi_{m+1}) = (\mathcal{I}_m^{m-1} \mathcal{S}_{m-1} \phi_{m-1}, \mathcal{I}_m^{m+1} \mathcal{S}_{m+1} \phi_{m+1}). \tag{4.15}$$

For the two "end" patches $\mathcal{K}_1$ and $\mathcal{K}_M$ that intersect with physical boundary Γ_-, boundary conditions are updated only in the interior of the domain:

$$\mathcal{P}_1 : \mathcal{X} \times \mathcal{X}_2 \to \mathcal{X}_1, \qquad \mathcal{P}_1(\phi, \phi_2) = (\phi|_{\Gamma_- \cap \Gamma_{1,-}}, \mathcal{I}_1^2 \mathcal{S}_2 \phi_2)\,,$$
$$\mathcal{P}_M : \mathcal{X}_{M-1} \times \mathcal{X} \to \mathcal{X}_M, \qquad \mathcal{P}_M(\phi_{M-1}, \phi) = (\mathcal{I}_M^{M-1} \mathcal{S}_{M-1} \phi_{M-1}, \phi|_{\Gamma_- \cap \Gamma_{M,-}})\,.$$

As suggested by Algorithm 2.1, in the offline stage, we construct local dictionaries on interior patches from a few random samples, enlarging each interior patch slightly to eliminate the boundary layer effect. Define $\widetilde{\mathcal{K}}_m$ and $\widetilde{\mathcal{D}}_m$ such that

$$\mathcal{K}_m \subset \widetilde{\mathcal{K}}_m = \widetilde{\mathcal{D}}_m \times \mathcal{V}\,,$$

where $\mathcal{D}_m \subset \widetilde{\mathcal{D}}_m \subset \mathcal{D}$ expands the boundary of $\mathcal{D}_m$ to both sides by a margin of Δx_{b}. Denoting by $\widetilde{\Gamma}_{m,-}$ the boundary coordinates corresponding to $\widetilde{\mathcal{D}}_m$, we let $\widetilde{\mathcal{X}}_m = L^2(\widetilde{\Gamma}_{m,-}) \times \mathbb{R}^2$ capture the boundary conditions on $\partial \widetilde{\mathcal{D}}_m$.

We draw N samples $\widetilde{\phi}_{m,i}$, $i = 1, 2, \ldots, N$, randomly from the set

$$B_+(R_m; \widetilde{\mathcal{X}}_m) := \{\widetilde{\phi} = (\widetilde{I}_B, \widetilde{T}_B) \in \widetilde{\mathcal{X}}_m : \|\widetilde{\phi}\|_{\widetilde{\mathcal{X}}_m} \le R_m,\ \widetilde{I}_B \ge 0,\ \widetilde{T}_B \ge 0\}\,.$$

(The sampling procedure is discussed in in Appendix B.) The local solutions $\widetilde{u}_{m,i} = (\widetilde{I}^\epsilon_{m,i}, \widetilde{T}^\epsilon_{m,i})$ solve

$$\begin{cases} \epsilon v \partial_x \widetilde{I}^\epsilon_{m,i} = (\widetilde{T}^\epsilon_{m,i})^4 - \widetilde{I}^\epsilon_{m,i}, \\ \epsilon^2 \partial_x^2 \widetilde{T}^\epsilon_{m,i} = (\widetilde{T}^\epsilon_{m,i})^4 - \langle \widetilde{I}^\epsilon_{m,i} \rangle, \end{cases} \qquad (x, v) \in \widetilde{\mathcal{K}}_m\,, \tag{4.16}$$
$$(\widetilde{I}^\epsilon_{m,i}|_{\widetilde{\Gamma}_{m,-}}, \widetilde{T}^\epsilon_{m,i}|_{\partial \widetilde{\mathcal{D}}_m}) = \widetilde{\phi}_{m,i}\,, \quad i = 1, 2, \ldots, N.$$

The solutions to these equations, confined to the original patch $\mathcal{K}_m$ and its boundary Γ_m, are used to construct two dictionaries,

$$\mathscr{I}_m = \{\psi_{m,i}\}_{i=1}^N, \quad \mathscr{B}_m = \{\phi_{m,i}\}_{i=1}^N, \tag{4.17}$$

where

$$\psi_{m,i} = (\widetilde{I}^\epsilon_{m,i}|_{\mathcal{K}_m}, \widetilde{T}^\epsilon_{m,i}|_{\mathcal{D}_m})\,, \quad \phi_{m,i} = (\widetilde{I}^\epsilon_{m,i}|_{\Gamma_{m,-}}, \widetilde{T}^\epsilon_{m,i}|_{\partial \mathcal{D}_m})\,.$$

In the online stage, at each iteration we seek neighbors to interpolate for local solutions. We use the L^2-norm to measure the distance between the newly generated solutions and the older solution set. Denote by $\phi_m^{(n)}$ the solution at the nth iteration in patch $\mathcal{K}_m$, and define by

$$\{\phi_{m,i_q^{(n)}}\,, \quad q = 1, 2, \ldots, k\}$$

its k-nearest neighbors in $\mathscr{B}_m$, for some chosen positive integer k, with the indices $i_q^{(n)}$ being ordered so that $\phi_{m,i_1^{(n)}}$ is the nearest neighbor. Then we define the local tangential approximation $\mathcal{S}_m \phi_m^{(n)}$ by

$$u_m^{(n)} = \mathcal{S}_m \phi_m^{(n)} = \psi_{m,i_1^{(n)}} + \Psi_m^{(n)} c_m^{(n)}\,, \tag{4.18}$$

where $\Psi_m^{(n)}$ and $c_m^{(n)}$ are defined as in (2.8) and (2.9). The local solution is then updated as follows:

$$\phi_m^{(n+1)} = \mathcal{P}_m(\phi_{m-1}^{(n)}, \phi_{m+1}^{(n)}) = (\mathcal{I}_m^{m-1} \mathcal{S}_{m-1} \phi_{m-1}^{(n)}, \mathcal{I}_m^{m+1} \mathcal{S}_{m+1} \phi_{m+1}^{(n)})\,. \tag{4.19}$$

For $m = 1$ and $m = M$, to avoid updating the physical boundary, we set

$$\mathcal{P}_1(\phi, \phi_2^{(n)}) = (\phi|_{\Gamma_- \cap \Gamma_{1,-}}, \mathcal{I}_1^2 \mathcal{S}_2 \phi_2^{(n)}),$$
$$\mathcal{P}_M(\phi_{M-1}^{(n)}, \phi) = (\mathcal{I}_M^{M-1} \mathcal{S}_{M-1} \phi_{M-1}^{(n)}, \phi|_{\Gamma_- \cap \Gamma_{M,-}}).$$

Once the convergence is achieved (at iteration n, say), we assemble the final solution as

$$u_{\text{final}} = u^{(n)} = \sum_{m=1}^{M} \chi_m u_m^{(n)}, \tag{4.20}$$

with $\chi_m : \Omega \to \mathbb{R}$ being the smooth partition of unity associated with the partition of $\mathcal{K}$.

4.4. Numerical tests. In the numerical tests, we take the domain to be

$$\mathcal{K} = \mathcal{D} \times \mathcal{V} = [0, L] \times [-1, 1] = [0, 3] \times [-1, 1].$$

To form the patch $\mathcal{K}_m = \mathcal{D}_m \times \mathcal{V}$, the domain $\mathcal{D}$ is divided into $M = 7$ nonoverlapping patches whose widths are $d_1 = d_7 = \frac{L}{2(M-1)} = 0.25$ and $d_i = \frac{L}{M-1} = 0.5$, $i = 2, \ldots, M-1$. Each patch is then enlarged by $\Delta x_{\text{o}} = .125$ on both sides (except the ones adjacent to the physical boundary, which are enlarged only on the "internal" sides), so we have

$$\mathcal{D}_m = \left(\tfrac{L(2m-1)}{2(M-1)} - \Delta x_{\text{o}}, \tfrac{L(2m-1)}{M} + \Delta x_{\text{o}} \right), \quad m = 2, \ldots, M-1,$$
$$\mathcal{D}_1 = \left(0, \tfrac{L}{2(M-1)} + \Delta x_{\text{o}} \right), \quad \mathcal{D}_M = \left(L - \tfrac{3}{2(M-1)} - \Delta x_{\text{o}}, 3 \right).$$

The region of overlap between adjacent patches $\mathcal{K}_m$ has size $2\Delta x_{\text{o}} \times [-1, 1]$. The partition of unity functions over each patch $\mathcal{K}_m$ are obtained using the method of subsection 3.4

Denote the spatial grid points by $0 = x_0 < x_1 < \cdots < x_{N_x - 1} < x_{N_x} = L$, which is a uniform grid with step size $\Delta x = \frac{L}{N_x}$. The velocity grid points are denoted by $-1 < v_1 < v_2 < \cdots < v_{N_v - 1} < v_{N_v} < 1$ for some even value of N_v. We use the Gauss–Legendre quadrature points for the v_i. The numerical solutions are denoted by $I^{ij} \approx I(x_i, v_j)$ and $T^i \approx T(x_i)$. To quantify the numerical error, we denote the discrete L^2-norm of $u = ([I^{ij}], [T^i])$ by

$$\|u\|_2^2 = \sum_{j=1}^{N_v} w_j \frac{\Delta x}{2} |I^{0j}|^2 + \sum_{j=1}^{N_v} w_j \frac{\Delta x}{2} |I^{iN_x}|^2 + \sum_{i=1}^{N_x - 1} \sum_{j=1}^{N_v} w_j \Delta x |I^{ij}|^2$$
$$+ \frac{\Delta x}{2} |T^0|^2 + \frac{\Delta x}{2} |T^{N_x}|^2 + \sum_{i=1}^{N_x - 1} \Delta x |T^i|^2,$$

where w_j is the Gauss–Legendre weight, and the relative error u_{ref} between a reference solution and an approximate solution u_{approx} is defined by

$$\text{relative } L^2 \text{ error} = \frac{\|u_{\text{ref}} - u_{\text{approx}}\|_2}{\|u_{\text{ref}}\|_2}.$$

We solve the PDE using finite differences. The intensity equation is discretized in space by a classical second-order exponential finite difference scheme [61, 81], and the temperature equation is approximated by the standard three-point scheme. The

resulting nonlinear system is then solved by fixed point iteration [69, 72], where in each evaluation of the fixed point map, the monotone iterative method is exploited to solve the semilinear elliptic equation. For computations with $\epsilon = 2^{-4}$ and $\epsilon = 2^{-6}$, we further use Anderson acceleration to boost the convergence of fixed point iteration [10, 45, 87].

We use extremely fine discretization with $\Delta x = 2^{-14} = \frac{1}{16384}$ and $N_v = 2^{10} = 1024$. The discretization is fine enough for us to view it as the reference solution. All other computations are done with coarser mesh $\Delta x = 2^{-11} = \frac{1}{2048}$ and $N_v = 2^7 = 128$.

The boundary condition $\phi = (g^{(1)}, g^{(2)}, \theta^{(1)}, \theta^{(2)})$ is defined as follows:

$$g^{(1)}(0, v > 0) = 3 + \sin(2\pi v)\,, \quad g^{(2)}(L = 3, v < 0) = 2 + \sin(2\pi v)\,,$$
$$\theta(0) = \theta^{(1)} = 2\,, \quad \theta(L) = \theta^{(2)} = 3\,.$$

The enlarged patches needed in the offline stage, denoted by $\widetilde{\mathcal{K}}_m$, are obtained by enlarging each respective $\mathcal{K}_m$ by the quantity Δx_{b}. The configuration of the domain and the partition are seen in Figure 7, where $\Delta x_{\mathrm{b}} = .125$.

On the buffered interior patch $\widetilde{\mathcal{K}}_m$, we sample $N = 64$ configurations of boundary conditions in $B_+(R_m; \widetilde{\mathcal{X}}_m)$. On the discrete level, this process finds 64 boundary conditions $\widetilde{\phi}$ so that

$$\|\widetilde{\phi}\|^2 = \sum_{j=1}^{\frac{N_v}{2}} w_j |\widetilde{g}^{(2)}(s, v_j)|^2 + \sum_{j=\frac{N_v}{2}+1}^{N_v} w_j |\widetilde{g}^{(1)}(t, v_j)|^2 + |\widetilde{\theta}^{(1)}|^2 + |\widetilde{\theta}^{(2)}|^2 < R_m\,.$$

We set $R_m = 25$ in our experiments.

To demonstrate the linearity of the updating map $\mathcal{P}_m$, we choose the patch $\mathcal{K}_3 = [0.625, 1.375] \times [-1, 1]$, which overlaps $\mathcal{K}_2$ at $[0.625, 0.875] \times [-1, 1]$. For $\Delta x_{\mathrm{b}} = 2^{-3}$ and $\epsilon = 2^{-6}$, we compute local solutions on the buffered domain $\widetilde{\mathcal{K}}_3$ with 64 different configurations and evaluate T at 0.625 and 1.375 (the two ending points of $\mathcal{K}_3$) and at 0.875 (the point that intersects with $\partial\mathcal{K}_2$). In Figure 8, we plot $T(0.875)$ as a function of $T(0.625)$ and $T(1.375)$. We observe that it is a slowly varying two-dimensional manifold and is locally almost linear. Thus, $T(0.875)$ can be determined uniquely by the pair of values $(T(0.625), T(1.375))$. Further, we plot $\frac{\langle |I(x,\cdot) - \langle I\rangle(x)|^2\rangle}{\langle I\rangle(x)^2}$

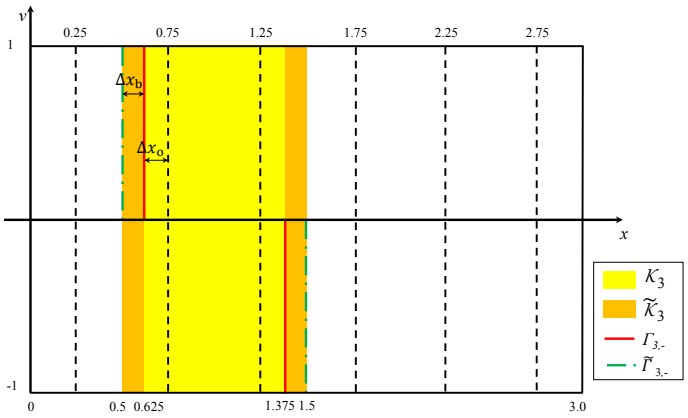


FIG. 7. *Configuration of patches (including enlarged patches) in the decomposed domain.*

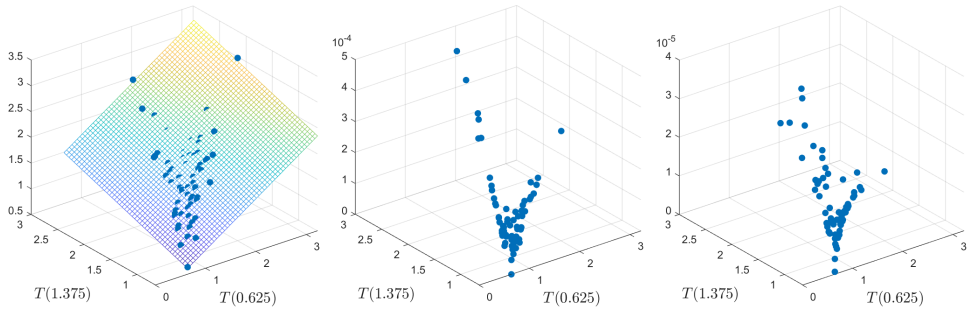
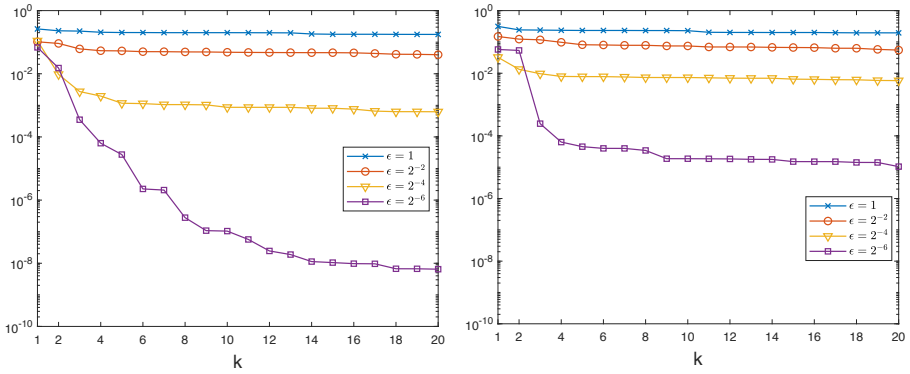


FIG. 8. *The plot on the left shows the point cloud* $(T(0.625), T(1.375), T(0.875))$ *and its fitting plane. We observe that the manifold is approximately two-dimensional, so that* $T(0.875)$ *can be uniquely determined by* $(T(0.625), T(1.375))$. *The middle and right panels show the quantities* $\frac{\langle |I(x,\cdot)-\langle I\rangle(x)|^2\rangle}{\langle I\rangle(x)^2}$ *and* $\frac{\langle I\rangle(x)-T(x)^4}{T(x)^4}$ *at* $x = 0.875$, *respectively, showing that the solution is nearly constant, with* $I = T^4$.



(a) $\Delta x_{\mathrm{b}} = 2^{-2}$ (b) $\Delta x_{\mathrm{b}} = 2^{-3}$

FIG. 9. *The relative error of the* L^2 *projection of the reference solution onto the space spanned by the nearest* k *modes on the patch* $\mathcal{K}_2$.

and $\frac{\langle I\rangle(x)-T^4(x)}{T^4(x)}$ at $x = 0.875$, showing that the relative variation is nearly zero. This means that I is essentially constant at $x = 0.875$, with $I = T^4$. These calculations suggest that the entire solution on this patch is uniquely determined by $T(0.625)$ and $T(1.375)$, implying that the local degrees of freedom for the solution in the entire patch is only two, so that the local solution manifold is approximately two-dimensional.

To verify that the local dictionary represents the solution manifold adequately, we confine the reference solution in patch $\mathcal{K}_2$ and project it onto the space spanned by its nearest k modes in the local dictionary. We evaluate the resulting relative error as a function of k, plotting the result in Figure 9. For $\epsilon = 2^{-6}$ and $\Delta x_{\mathrm{b}} = .125$, we observe a sharp decay in the error when $k \geq 3$, meaning that the local reference solution can be represented to acceptable accuracy by two local dictionary modes, and suggesting once again that the local solution manifold is two-dimensional.

The sample number N and the radius R_m are two crucial parameters that influence the effectiveness of the method. We check how the approximation capability of the local dictionary depends on the two parameters over the local patch $\mathcal{K}_2$. In

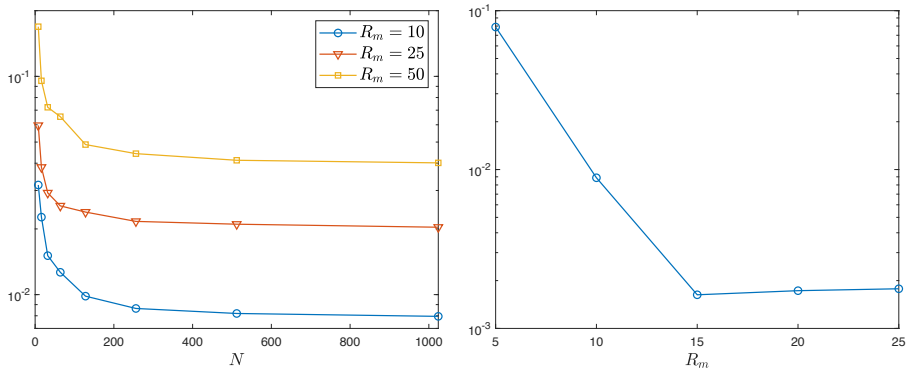


FIG. 10. *The plot on the left shows the average error of the L^2 projection of 100 test samples onto the space spanned by the nearest five modes. The test samples are generated from the same distribution as the dictionary. The plot on the right shows the relative error of the L^2 projection of the reference solution onto the space spanned by the nearest five modes on patch $\mathcal{K}_2$. The number of samples is $N = 64$ for all R_m.*

Figure 10a, we show the projection error as N increases for different R_m. The error of the dictionary saturates as N increases, and it can be used as a criterion for deciding the size of the local dictionary. In Figure 10b, we show the relative projection error of the reference solution onto the local tangent space using dictionaries with different R_m. It can be seen that the radius R_m must be large enough to obtain a good local basis.

In the online computation, we set the stopping criterion to be

$$\sum_m \|\phi_m^{(n)} - \phi_m^{(n-1)}\| < 10^{-3},$$

where $\phi_m^{(n)}$ is the boundary condition on the patch $\mathcal{K}_m$ at the nth iteration. We take the initial boundary condition on each patch to be trivial, setting $\phi_m^{(0)}|_{\Gamma_{m,-}\setminus\Gamma_-} = 0$, except on the real physical boundary condition, where it is set to the prescribed Dirichlet conditions.

In Figure 11, we compare the reference solution with our numerical solution computed using $k = 5$ and buffer zone $\Delta x_{\mathrm{b}} = 2^{-3}$. When $\epsilon = 1$, the equation is far away from its homogenization limit, and the numerical solution is far from the reference, but for $\epsilon = 2^{-6}$ the numerical solution is captured rather well using just $k = 5$ neighbors.

In Figure 12 we document the relative error for various values of k and Δx_{b}. When ϵ is small, and for buffer width Δx_{b} sufficiently large, we need only $k = 2$ neighbors to produce a solution of acceptable accuracy. Without the buffer zone to dampen the boundary layer effect, however, the low dimensionality of the solution manifold cannot be captured, even for small ϵ.

We also compare the cost of our reduced method with the classical Schwarz iteration. CPU times for both methods are summarized in Table 2 for $\epsilon = 2^{-4}$ and $\epsilon = 2^{-6}$, with buffer size $\Delta x_{\mathrm{b}} = .125$. The online cost of the reduced method is about 1000 times cheaper than the classical Schwarz iteration when $\epsilon = 2^{-4}$ and 4000 times cheaper when $\epsilon = 2^{-6}$. Even considering the large overhead cost in the offline stage, the reduced order method is still cheaper than Schwarz iteration.

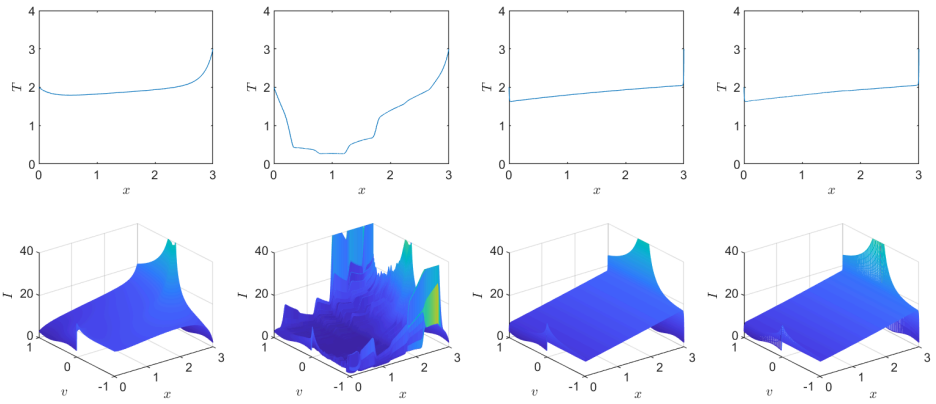
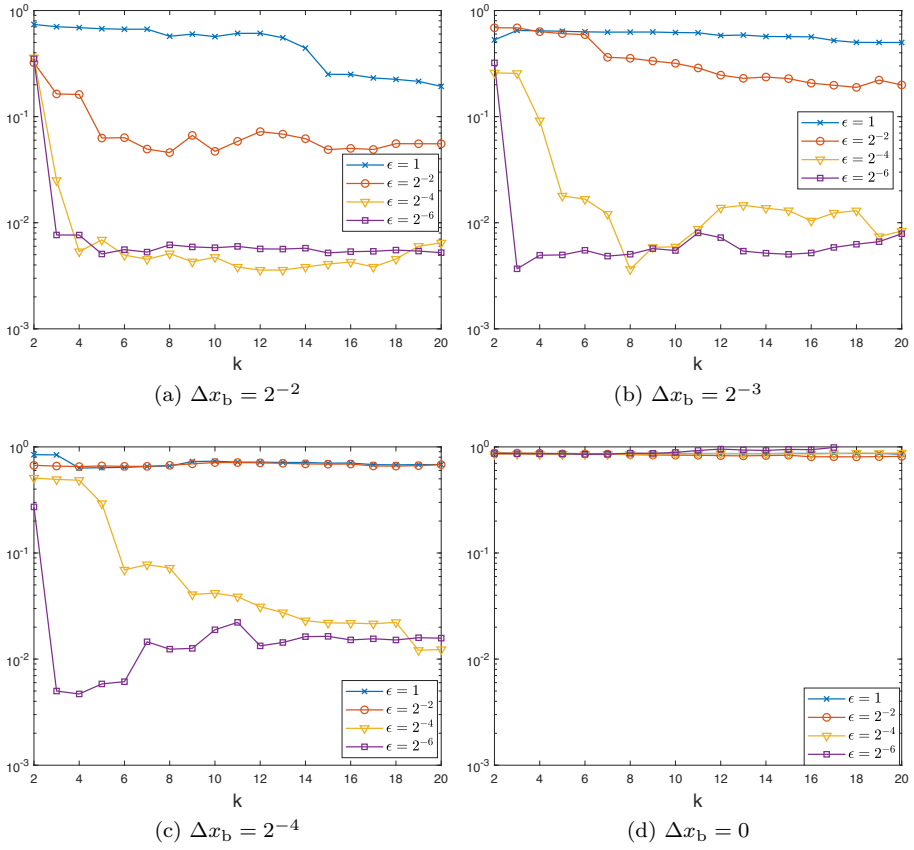


FIG. 11. *The first two columns of plots show the reference solution and numerical solution for* $\epsilon = 1$*, and the last two columns compare the solutions for* $\epsilon = 2^{-6}$.



(a) $\Delta x_{\mathrm{b}} = 2^{-2}$ (b) $\Delta x_{\mathrm{b}} = 2^{-3}$

(c) $\Delta x_{\mathrm{b}} = 2^{-4}$ (d) $\Delta x_{\mathrm{b}} = 0$

FIG. 12. *The relative* L^2 *error in one trial as a function of* k*, for various values of* Δx_b *and* ϵ.

Finally, we reiterate that due to the nonlinear nature of the equations, the concept of "basis function" is not well defined. The reduced model method for linear equations was proposed in [26, 28], where random sampling is used to construct the

TABLE 2
CPU time comparison between reduced model methods with $k = 3, 5, 10, 15, 20$ (size of each local dictionary $N = 64$).

	CPU Time (s)			
	$\epsilon = 2^{-4}$		$\epsilon = 2^{-6}$	
	Offline	Online	Offline	Online
Reduced model $k = 3$	394.3911	0.181324	904.7498	0.215390
Reduced model $k = 5$		0.301761		0.222538
Reduced model $k = 10$		0.379348		0.282070
Reduced model $k = 15$		0.548689		0.346633
Reduced model $k = 20$		0.586276		0.532603
Classical Schwarz	—	458.0987	—	2183.7079

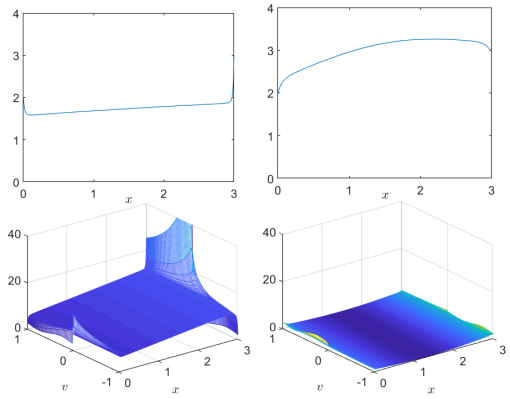


FIG. 13. *The left column of plots is the solution with $\epsilon = 2^{-4}$, and the right column is the solution from a linear combination of the full set of "Green's functions."*

boundary-to-boundary map $\mathcal{P}$, by following the idea of randomized SVD [54]. If we translate this approach into nonlinear homogenization, using Green's functions in a brute-force manner, the numerical results are poor. By the "Green's functions" we mean the solution to the equation with delta boundary conditions (counterparts of Green's functions in the linear setting). The numerical results are presented in Figure 13, which compares the ground-truth solution with the Green's function interpolation.

5. Conclusion. Multiscale physical phenomena are often described by PDEs that contain small parameters. it is generally expensive to capture small-scale effects using numerical solvers. There is a vast literature on improving numerical performance of PDE solvers in this context, but most algorithms are equation-specific, requiring analytical understanding to be built into algorithmic design.

We have described numerical methods that can capture the homogenization limit of nonlinear PDEs with small scales automatically, without analytical prior knowledge. This work can be seen as a nonlinear extension of our earlier work [27] for linear PDEs. Elements of our algorithm include a domain decomposition framework and Schwarz iteration. The method is decomposed into offline and online stages, where in the offline stage, random sampling is employed to learn the low-rank structure of the solution manifold, while in the online stage, the reduced manifolds serve as surrogates

of local solvers in the Schwarz iteration. Since the manifolds are prepared offline and are of low dimension, the method exhibits significant speedup over naive approaches, as we demonstrate using computational results on two examples.

Appendix A. Sampling method for the semilinear elliptic equation. We explain here the sampling method for the semilinear elliptic equation in subsection 3.4. To enforce the boundary condition on the physical boundary, patches that intersect this boundary should be treated differently from patches inside the domain. (We call the patch $\widetilde{\Omega}_m$ an "interior patch" if it satisfies $\partial\widetilde{\Omega}_m \cap \partial\Omega = \varnothing$; otherwise, we call it a "boundary patch.")

A.1. Sampling for interior patches. For the interior patch $\partial\widetilde{\Omega}_m$, each sample in $B(R_m; \widetilde{\mathcal{X}}_m)$ is decomposed into radial and angular parts $\widetilde{\phi} = rX$, with the two parts r and X sampled independently. The radial part r is generated so that $(\frac{r}{R_m})^D$ is uniformly distributed in the unit interval $[0,1]$, where D is a preset integer. (We choose $D = 5$ and $R_m = R = 20$ in our tests.) The angular part X is an N_m-dimensional vector uniformly distributed in the set $\{X \in \mathbb{R}^{N_m} : \|X\|_{1/2} = 1\}$, where N_m is the number of grid points on $\partial\widetilde{\Omega}_m$, and the norm $\|\cdot\|_{1/2}$ is defined by

$$\|\widetilde{\phi}\|_{1/2} = \sqrt{h\sum_{i=1}^{N_m} |\widetilde{\phi}_i|^2 + h^2 \sum_{\substack{i,j=1 \\ i\neq j}}^{N_m} \frac{|\widetilde{\phi}_i - \widetilde{\phi}_j|^2}{|z_i - z_j|^2}}\,.$$

Here $\widetilde{\phi} = (\widetilde{\phi}_i)_{i=1}^{N_m}$ is any discrete boundary condition, and z_i denotes the grid point on $\partial\widetilde{\Omega}_m$.

In order to generate X, let $Y_1, \ldots, Y_{N_m} \sim \mathcal{N}(0,1)$ be independent and identically distributed (i.i.d.(standard Gaussian random variables. Define the weight matrix $W = (W_{ij})_{N_m \times N_m}$ by

$$W_{ii} = h + \sum_{\substack{j=1 \\ j\neq i}}^{N_m} \tfrac{2h^2}{|z_i - z_j|^2}\,, \quad W_{ij} = -\tfrac{2h^2}{|z_i - z_j|^2}\,,$$

and suppose that its Cholesky decomposition is $W = C^\top C$. Then the vector $Z = C^{-1}(Y_1, \ldots, Y_{N_m})^\top$ has uniform angular distribution with respect to the norm $\|\cdot\|_{1/2}$, so its normalization $X = \frac{Z}{\|Z\|_{1/2}}$ is uniformly distributed on the unit sphere $\{X \in \mathbb{R}^{N_m} : \|X\|_{1/2} = 1\}$.

A.2. Sampling for boundary patches. Let

$$\widetilde{\phi}_m = \begin{bmatrix} \widetilde{\phi}_{m,d} \\ \widetilde{\phi}_{m,r} \end{bmatrix} \in \mathbb{R}^{N_m}$$

be a random sample, with $\widetilde{\phi}_{m,d}$ representing the physical boundary part and $\widetilde{\phi}_{m,r}$ representing the random part. When we rearrange the weight matrix W as

$$W = \begin{bmatrix} W_{dd} & W_{dr} \\ W_{rd} & W_{rr} \end{bmatrix}\,,$$

so that $\|\widetilde{\phi}_m\|_{1/2}^2 = \widetilde{\phi}_m^\top W \widetilde{\phi}_m$, then it yields

$$\widetilde{\phi}_{m,r}^\top W_{rr} \widetilde{\phi}_{m,r} = R_m^2 - \widetilde{\phi}_{m,d}^\top (W_{dd} - W_{dr} W_{rr}^{-1} W_{rd}) \widetilde{\phi}_{m,d}\,,$$

indicating that the random part lies in an ellipsoid.

Hence, the random part $\widetilde{\phi}_{m,r}$ can be sampled as follows. We decompose it into independently sampled radial and angular parts $\widetilde{\phi}_r = r_m X_m$, so that r_m^D is uniformly distributed in the interval $\left[0, (R_m^2 - \widetilde{\phi}_{m,d}^\top (W_{dd} - W_{dr}W_{rr}^{-1}W_{rd})\widetilde{\phi}_{m,d})^{D/2}\right]$, and X_m is uniformly distributed on the set $\{X_m : X_m^\top W_{rr} X_m = 1\}$.

Appendix B. Sampling method for the nonlinear radiative transfer equations. Here we describe the sampling method for the nonlinear radiative transfer equations discussed in subsection 4.4.

To generate samples for the interior patches $\widetilde{\mathcal{K}}_m$, $m = 2, \ldots, M-1$, each sample is decomposed into radial and angular parts $\widetilde{\phi} = rX$, which are sampled independently. We take $(\frac{r}{R_m})^2$ to be uniformly distributed in $[0,1]$, while X is an (N_v+2)-dimensional vector uniformly distributed in the set $\{X \in \mathbb{R}^{N_v+2} : \|X\| = 1, X \geq 0\}$, where the norm $\|\cdot\|$ is defined by

$$\|\widetilde{\phi}\|^2 = \sum_{j=1}^{\frac{N_v}{2}} w_j |\widetilde{g}^{(2)}(s, v_j)|^2 + \sum_{j=\frac{N_v}{2}+1}^{N_v} w_j |\widetilde{g}^{(1)}(t, v_j)|^2 + |\widetilde{\theta}^{(1)}|^2 + |\widetilde{\theta}^{(2)}|^2,$$

given any discrete boundary condition

$$\widetilde{\phi} = \left(\{\widetilde{g}^{(2)}(s, v_j)\}_{j=1}^{\frac{N_v}{2}}, \{\widetilde{g}^{(1)}(t, v_j)\}_{j=\frac{N_v}{2}+1}^{N_v}, \widetilde{\theta}^{(1)}, \widetilde{\theta}^{(2)} \right).$$

Here N_v is the number of grid points in the velocity direction, and the w_j are the Gaussian–Legendre weights. (We choose $R_m = R = 25$ in our tests.)

To generate X, let $Y_1, \ldots, Y_{N_v+2} \sim \mathcal{N}(0,1)$ be i.i.d. standard Gaussian random variables. Denote the vector

$$Z = \left(\frac{Y_1}{\sqrt{w_1}}, \cdots, \frac{Y_{N_v}}{\sqrt{w_{N_v}}}, Y_{N_v+1}, Y_{N_v+2} \right).$$

Then the normalized vector $X = \frac{Z}{\|Z\|}$ is uniformly distributed on the unit sphere $\{X \in \mathbb{R}^{N_v+2} : \|X\| = 1\}$. Note that (4.16) is invariant under x-translation, so we need only learn one interior dictionary on one interior patch, and then we reuse in for the other interior patches.

Sampling the boundary conditions on the boundary patches can be done in the same way. However, we do adjust the radius r. In particular, $(\frac{r}{R_{1/M}})^2$ is chosen uniformly in $[0,1]$, where $R_{1/M}$ has the fixed boundary condition deducted from R.

REFERENCES

[1] N. B. Abdallah, M. Puel, and M. S. Vogelius, *Diffusion and homogenization limits with separate scales*, Multiscale Model. Simul., 10 (2012), pp. 1148–1179, https://doi.org/10.1137/110828964.

[2] A. Abdulle and Y. Bai, *Reduced basis finite element heterogeneous multiscale method for high-order discretizations of elliptic homogenization problems*, J. Comput. Phys., 231 (2012), pp. 7014–7036.

[3] A. Abdulle, Y. Bai, and G. Vilmart, *Reduced basis finite element heterogeneous multiscale method for quasilinear elliptic homogenization problems*, Discrete Contin. Dyn. Syst. Ser. S, 8 (2015), pp. 91–118.

[4] A. Abdulle and C. Schwab, *Heterogeneous multiscale FEM for diffusion problems on rough surfaces*, Multiscale Model. Simul., 3 (2005), pp. 195–220, https://doi.org/10.1137/030600771.

[5] A. Abdulle and G. Vilmart, *Analysis of the finite element heterogeneous multiscale method for quasilinear elliptic homogenization problems*, Math. Comp., 83 (2014), pp. 513–536.
[6] V. Agoshkov, Boundary Value Problems for Transport Equations, Model. Simul. Sci. Eng. Technol., Birkhäuser Boston, Inc., 1998.
[7] G. Allaire, *Homogenization and two-scale convergence*, SIAM J. Math. Anal., 23 (1992), pp. 1482–1518, https://doi.org/10.1137/0523084.
[8] W. K. Allard, G. Chen, and M. Maggioni, *Multi-scale geometric methods for data sets* II*: Geometric multi-resolution analysis*, Appl. Comput. Harmon. Anal., 32 (2012), pp. 435–462.
[9] H. Amann and J. Moser, *On the existence of positive solutions of nonlinear elliptic boundary value problems*, Indiana Univ. Math. J., 21 (1971), pp. 125–146.
[10] D. G. Anderson, *Iterative procedures for nonlinear integral equations*, J. ACM, 12 (1965), pp. 547–560.
[11] A. Andoni, P. Indyk, and I. Razenshteyn, *Approximate Nearest Neighbor Search in High Dimensions*. preprint, https://arxiv.org/abs/1806.09823, 2018.
[12] I. Babuška and R. Lipton, *Optimal local approximation spaces for generalized finite element methods with application to multiscale problems*, Multiscale Model. Simul., 9 (2011), pp. 373–406, https://doi.org/10.1137/100791051.
[13] I. Babuška and J. M. Melenk, *The partition of unity method*, Int. J. Numer. Methods Eng., 40 (1997), pp. 727–758.
[14] C. Bardos, F. Golse, and D. Levermore, *Fluid dynamic limits of kinetic equations.* I. *Formal derivations*, J. Stat. Phys., 63 (1991), pp. 323–344.
[15] C. Bardos, F. Golse, and B. Perthame, *The Rosseland approximation for the radiative transfer equations*, Commun. Pure Appl. Math., 40 (1987), pp. 691–721.
[16] C. Bardos, F. Golse, B. Perthame, and R. Sentis, *The nonaccretive radiative transfer equations: Existence of solutions and Rosseland approximation*, J. Funct. Anal., 77 (1988), pp. 434–460.
[17] C. Bardos, R. Santos, and R. Sentis, *Diffusion approximation and computation of the critical size*, Trans. Amer. Math. Soc., 284 (1984), pp. 617–649.
[18] M. Bebendorf, *Why finite element discretizations can be factored by triangular hierarchical matrices*, SIAM J. Numer. Anal., 45 (2007), pp. 1472–1494, https://doi.org/10.1137/060669747.
[19] M. Belkin and P. Niyogi, *Laplacian eigenmaps and spectral techniques for embedding and clustering*, Adv. Neural Inform. Process. Syst. (2002), pp. 585–591.
[20] A. Bensoussan, L. Boccardo and F. Murat, *H convergence for quasi-linear elliptic equations with quadratic growth*, Appl. Math. Optim., 26 (1992), pp. 253–272.
[21] A. Bensoussan, J.-L. Lions and G. Papanicolaou, *Boundary layers and homogenization of transport processes*, Publ. Res. Inst. Math. Sci., 15 (1979), pp. 53–157.
[22] A. Bensoussan, J.-L. Lions and G. Papanicolaou, Asymptotic Analysis for Periodic Structures, vol. 374. American Mathematical Soc., 2011.
[23] A. Buhr and K. Smetana, *Randomized local model order reduction*, SIAM J. Sci. Comput., 40 (2018), pp. A2120–2151, https://doi.org/10.1137/17M1138480.
[24] S. Chandrasekhar, An Introduction to the Study of Stellar Structure, Vol. 2. Courier Corporation, 1957.
[25] K. Chen, Q. Li and J. G. Liu, *Online learning in optical tomography: A stochastic approach*, Inverse Probl., 34 (2018), 075010.
[26] K. Chen, Q. Li, J. Lu, and S. J. Wright, *Random sampling and efficient algorithms for multiscale PDEs*, SIAM J. Sci. Comput., 42 (2020), pp. A2974–A3005, https://doi.org/10.1137/18M1207430.
[27] K. Chen, Q. Li, J. Lu, and S. J. Wright, *Randomized sampling for basis function construction in generalized finite element methods*, Multiscale Model. Simul., 18 (2020), pp. 1153–1177, https://doi.org/10.1137/18M1166432.
[28] K. Chen, Q. Li, J. Lu, and S. J. Wright, *A low-rank Schwarz method for radiative transfer equation with heterogeneous scattering coefficient*, Multiscale Model. Simul., 19 (2021), pp. 775–801, https://doi.org/10.1137/19M1276327.
[29] Z. Chen and T. Y. Savchuk, *Analysis of the multiscale finite element method for nonlinear and random homogenization problems*, SIAM J. Numer. Anal., 46 (2008), pp. 260–279, https://doi.org/10.1137/060654207.
[30] M. Chipot, *Elliptic Equations: An Introductory Course*, Springer, 2009.
[31] E. T. Chung, Y. Efendiev, W. T. Leung, and M. Wheeler, *Nonlinear nonlocal multicontinua upscaling framework and its applications*, Int. J. Multiscale Comput. Eng., 16 (2018), pp. 487–507.

[32] E. T. Chung, Y. Efendiev, W. T. Leung, and Z. Zhang, *Cluster-based generalized multiscale finite element method for elliptic PDEs with random coefficients*, J. Comput. Phys., 371 (2018), pp. 606–617.
[33] R. R. Coifman, S. Lafon, A. B. Lee, M. Maggioni, B. Nadler, F. Warner, and S. W. Zucker, *Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps*, Proc. Natl. Acad. Sci. USA, 102 (2005), pp. 7426–7431.
[34] P. Degond, *Asymptotic-preserving schemes for fluid models of plasmas,* in Numerical Models for Fusion, Panor. Synthèses, 39/40, Soc. Math. France, 2013, pp. 1–90.
[35] E. Di Nezza, G. Palatucci, and E. Valdinoci, *Hitchhiker's guide to the fractional Sobolev spaces*, Bull. Sci. Math., 136 (2012), pp. 521–573.
[36] G. Dimarco and L. Pareschi, *High order asymptotic-preserving schemes for the Boltzmann equation*, C. R. Math. Acad. Sci. Paris, 350 (2012), pp. 481–486.
[37] G. Dimarco and L. Pareschi, *Numerical methods for kinetic equations*, Acta Numer., 23 (2014), pp. 369–520.
[38] L. Dumas and F. Golse, *Homogenization of transport equations*, SIAM J. Appl. Math., 60 (2000), pp. 1447–1470, https://doi.org/10.1137/S0036139997332087.
[39] E. W. and B. Engquist, *The heterogeneous multiscale methods*, Commun. Math. Sci., 1 (2003), pp. 87–132.
[40] E. W., P. Ming, and P. Zhang, *Analysis of the heterogeneous multiscale method for elliptic homogenization problems*, J. Am. Math. Soc., 18 (2005), pp. 121–156.
[41] Y. Efendiev, J. Galvis, G. Li, and M. Presho, *Generalized multiscale finite element methods. Nonlinear elliptic equations*, Commun. Comput. Phys., 15 (2014), pp. 733–755.
[42] Y. Efendiev and T. Y. Hou, Multiscale Finite Element Methods: Theory and Applications, Surv. Tutor. Appl. Math. Sci. 4, Springer , 2009.
[43] Y. Efendiev, T. Y. Hou, and V. Ginting, *Multiscale finite element methods for nonlinear problems and their applications*, Commun. Math. Sci., 2 (2004), pp. 553–589.
[44] Y. R. Efendiev, T. Y. Hou, and X.-H. Wu, *Convergence of a nonconforming multiscale finite element method*, SIAM J. Numer. Anal., 37 (2000), pp. 888–910, https://doi.org/10.1137/S0036142997330329.
[45] H.-R. Fang and Y. Saad, *Two classes of multisecant methods for nonlinear acceleration*, Numer. Linear Algebra Appl., 16 (2009), pp. 197–221.
[46] F. Filbet and S. Jin, *A class of asymptotic-preserving schemes for kinetic equations and related problems with stiff sources*, J. Comput. Phys., 229 (2010), pp. 7625–7648.
[47] F. Filbet and S. Jin, *An asymptotic preserving scheme for the ES-BGK model of the Boltzmann equation*, J. Comput. Phys., 46 (2011), pp. 204–224.
[48] P. Gérard, P. Markowich, N. Mauser and F. Poupaud, *Homogenization limits and Wigner transforms*, Commun. Pure Appl. Math., 50 (1997), pp. 323–379.
[49] D. Gilbarg and N. Trudinger, Elliptic Partial Differential Equations of Second Order, Springer, 2015.
[50] T. Goudon and A. Mellet, *Diffusion approximation in heterogeneous media*, Asymptot. Anal., 28 (2001), pp. 331–358.
[51] T. Goudon and A. Mellet, *Homogenization and diffusion asymptotics of the linear Boltzmann equation*, ESAIM Control Optim. Calc. Var., 9 (2003), pp. 371–398.
[52] Y. Guo and L. Wu, *Geometric correction in diffusive limit of neutron transport equation in 2D convex domains*, Arch. Ration. Mech. Anal., 226 (2017), pp. 321–403.
[53] W. Hackbusch, Hierarchical Matrices: Algorithms and Analysis, Springer Ser. Comput. Math. 49, Springer, 2015.
[54] N. Halko, P. G. Martinsson and J. A. Tropp, *Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions*, SIAM Rev., 53 (2011), pp. 217–288, https://doi.org/10.1137/090771806.
[55] P. Henning, A. Målqvist, and D. Peterseim, *A localized orthogonal decomposition method for semi-linear elliptic problems*, ESAIM Math. Model. Numer. Anal., 48 (2014), pp. 1331–1349.
[56] T. Y. Hou, F.-N. Hwang, P. Liu, and C.-C. Yao, *An iteratively adaptive multi-scale finite element method for elliptic PDEs with rough coefficients*, J. Comput. Phys., 336 (2017), pp. 375–400.
[57] T. Y. Hou, Q. Li, and P. Zhang, *Exploring the locally low dimensional structure in solving random elliptic PDEs*, Multiscale Model. Simul., 15 (2017), pp. 661–695, https://doi.org/10.1137/16M1077611.
[58] T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, J. Comput. Phys., 134 (1997), pp. 169–189.

[59] T. Y. Hou, X.-H. Wu, and Z. Cai, *Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients*, Math. Comp., 68 (1999), pp. 913–943.

[60] J. Hu, S. Jin, and Q. Li *Asymptotic-preserving schemes for multiscale hyperbolic and kinetic equations*, in Handbook of Numerical Methods for Hyperbolic Problems, Handb. Numer. Anal. 18, Elsevier/North-Holland, 2017, pp. 103–129.

[61] A. M. Il'in, *Differencing scheme for a differential equation with a small parameter affecting the highest derivative*, Math. Notes Acad. Sci. USSR, 6 (1969), pp. 596–602.

[62] P. Indyk and R. Motwani, *Approximate nearest neighbors: towards removing the curse of dimensionality*, in Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing, 1998, pp. 604–613.

[63] S. Jin, *Efficient asymptotic-preserving (AP) schemes for some multiscale kinetic equations*, SIAM J. Sci. Comput., 21 (1999), pp. 441–454, https://doi.org/10.1137/S1064827598334599.

[64] S. Jin and Q. Li, *A BGK-penalization-based asymptotic-preserving scheme for the multispecies Boltzmann equation*, Numer. Methods Partial Differ. Equations, 29 (2013), pp. 1056–1080.

[65] W. B. Johnson and J. Lindenstrauss, *Extensions of Lipschitz mappings into a Hilbert space*, in Proceedings of the Conference in Modern Analysis and Probability (New Haven, Conn., 1982), Contemp. Math. 26, Amer. Math. Soc., pp. 189–206.

[66] D. D. Joseph and T. S. Lundgren, *Quasilinear Dirichlet problems driven by positive sources*, Arch. Ration. Mech. Anal., 49 (1973), pp. 241–269.

[67] S. Klainerman and A. Majda, *Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids*, Commun. Pure Appl. Math., 34 (1981), pp. 481–524.

[68] S. Klainerman and A. Majda, *Compressible and incompressible fluids*, Commun. Pure Appl. Math., 35 (1982), pp. 629–651.

[69] A. Klar and C. Schmeiser, *Numerical passage from radiative heat transfer to nonlinear diffusion models*, Math. Models Methods Appl. Sci., 11 (2001), pp. 749–767.

[70] A. Klar and N. Siedow, *Boundary layers and domain decomposition for radiative heat transfer and diffusion equations: applications to glass manufacturing process*, Eur. J. Appl. Math., 9 (1998), pp. 351–372.

[71] M. Lemou and L. Mieussens, *A new asymptotic preserving scheme based on micro-macro formulation for linear kinetic equations in the diffusion limit*, SIAM J. Sci. Comput., 31 (2008), pp. 334–368, https://doi.org/10.1137/07069479X.

[72] Q. Li and W. Sun, *Applications of kinetic tools to inverse transport problems*, Inverse Probl., 36 (2020), 035011.

[73] Q. Li and L. Wang, *Implicit asymptotic preserving method for linear transport equations*, Commun. Comput. Phys., 22 (2017), pp. 157–181.

[74] S. Li, Z. Zhang, and H. Zhao, *A data-driven approach for multiscale elliptic PDEs with random coefficients based on intrinsic dimension reduction*, Multiscale Model. Simul., 18 (2020), pp. 1242–1271, https://doi.org/10.1137/19M1277485.

[75] P.-L. Lions, *On the existence of positive solutions of semilinear elliptic equations*, SIAM Rev., 24 (1982), pp. 441–467, https://doi.org/10.1137/1024101.

[76] A. V. Little, M. Maggioni, and L. Rosasco, *Multiscale geometric methods for data sets I: Multiscale SVD, noise and curvature*, Appl. Comput. Harmon. Anal., 43 (2017), pp. 504–567.

[77] A. Målqvist and D. Peterseim, *Localization of elliptic multiscale problems*, Math. Comp., 83 (2014), pp. 2583–2603.

[78] M. F. Modest, *Radiative Heat Transfer*, Elsevier Science, 2013.

[79] H. Owhadi and L. Zhang, *Metric-based upscaling*, Commun. Pure Appl. Math., 60 (2007), pp. 675–723.

[80] É. Pardoux, *BSDEs, Weak Convergence and Homogenization of Semilinear PDEs*, Springer Netherlands, 1999, pp. 503–549.

[81] H.-G. Roos, M. Stynes, and L. Tobiska, *Robust Numerical Methods for Singularly Perturbed Differential Equations: Convection-Diffusion-Reaction and Flow Problems*, 2nd ed., Springer Ser. Comput. Math. 24, Springer-Verlag, 2008.

[82] S. T. Roweis and L. K. Saul, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), pp. 2323–2326.

[83] S. Schochet, *The compressible Euler equations in a bounded domain: Existence of solutions and the incompressible limit*, Commun. Math. Phys., 104 (1986), pp. 49–75.

[84] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 2004.

[85] A. Toselli and O. Widlund, Domain Decomposition Methods—Algorithms and Theory, Springer Ser. Comput. Math. 34, Springer-Verlag, 2005.
[86] R. Viskanta and E. E. Anderson, *Heat transfer in semitransparent solids*, in Advances in Heat Transfer, Vol. 11, Elsevier, 1975, pp. 317–441.
[87] H. F. Walker and P. Ni, *Anderson acceleration for fixed-point iterations*, SIAM J. Numer. Anal., 49 (2011), pp. 1715–1735, https://doi.org/10.1137/10078356X.
[88] Z. Zhang and H. Zha, *Principal manifolds and nonlinear dimensionality reduction via tangent space alignment*, SIAM J. Sci. Comput., 26 (2004), pp. 313–338, https://doi.org/10.1137/S1064827502419154.