Check for
updates

Semi-Lagrangian Subgrid Reconstruction for Advection-Dominant Multiscale Problems with Rough Data

Konrad Simon[1] · **Jörn Behrens**[1]

© The Author(s) 2021

Abstract

We introduce a new framework of numerical multiscale methods for advection-dominated problems motivated by climate sciences. Current numerical multiscale methods (MsFEM) work well on stationary elliptic problems but have difficulties when the model involves dominant lower order terms. Our idea to overcome the associated difficulties is a semi-Lagrangian based reconstruction of subgrid variability into a multiscale basis by solving many local inverse problems. Globally the method looks like a Eulerian method with multiscale stabilized basis. We show example runs in one and two dimensions and a comparison to standard methods to support our ideas and discuss possible extensions to other types of Galerkin methods, higher dimensions and nonlinear problems.

Keywords Multiscale simulation · Semi-Lagrangian · Inverse problems · Advection–diffusion · Multiscale finite elements

1 Introduction

1.1 Motivation and Overview

Simulating complex physical processes at macroscopic coarse scales poses many problems to engineers and scientists. Such simulations strive to reflect the effective behavior of observables involved at large scales even if the processes are partly driven by highly heterogeneous micro scale behavior. On the one hand resolving the microscopic processes would be the safest choice but such a strategy is prohibitive since it would be computationally expensive. On the other hand microscopic processes significantly influence the macroscopic behavior and cannot be neglected.

Incorporating micro scale effects into macro simulations in a mathematically consistent way is a challenging task. There exist many scenarios in different disciplines of science that are faced with such challenges. In fully coupled paleo climate simulations, i.e., climate simulations over more than hundred thousand years a typical grid cell has edge lengths of around

✉ Konrad Simon
konrad.simon@uni-hamburg.de

1 Department of Mathematics, University of Hamburg, Grindelberg 5, 20144 Hamburg, Germany

200 km and more. Consequently, subgrid processes such as heterogeneously distributed and moving ice shields are not or just insufficiently resolved [30]. These subgrid processes are usually taken care of by so-called parameterizations. One can imagine this as small micro scale simulations that are then coupled to the prognostic variables such as wind speed, temperature and pressure on the coarse grid (scale of the dynamical core). This coupling from fine to coarse scales is being referred to as upscaling and is unfortunately often done in rather heuristic ways. This leads to wrong macroscopic quantities like wrong pressures and eventually even to wrong wind directions and more undesired effects such as phase errors.

The increasing complexity of earth system models (ESMs), in general, demands for mathematically consistent upscaling of parametrized processes that occur or are modelled on very different scales relative to the dynamical core. A first example was already mentioned: In global climate simulations a sea ice model computes the average ice cover for each coarse cell. This information then enters heat fluxes between ocean and atmosphere since sea ice forms an interface between them and is hence modeled as an averaged diffusive process. It is known from homogenization theory that simple averaging of heterogeneous diffusive quantities does not reflect the correct coarse scale diffusion [8].

Another, quite contrary, approach to perform multiscale simulations is adaptive mesh refinement (AMR), see [7]. The idea of AMR is to initially run a coarse scale simulation and then assess the quality of the solution locally by means of mathematically or physically based a posteriori estimators. The coarse mesh is then refined in regions where the quality of the solution is not sufficient or coarsened where the solution is smooth. This approach is different from the above upscaling since micro scales are locally resolved—it is therefore sometimes referred to as downscaling. Such an approach is attractive and is being used in practice but it is not feasible in situations in which heterogeneities in the solution are expected to occur globally.

One early attempt to use multiscale methods are Eulerian Lagrangian localized adjoint methods (ELLAM), which constitute a space-time finite element framework, see [9,18], and form the basis of the multiscale version MsELLAM [10,37]. Such methods rely on operator splitting and basis functions are required to satisfy an adjoint equation. For a review see [32].

There exist many other multiscale methods. Homogenization is an originally analytical tool to find effective models of otherwise heterogeneous models in the limit of a large scale separation [5,8,25]. Unfortunately it is often too difficult to find such an effective equation but there exist numerical techniques that aim at designing numerical algorithms to effectively capture the behavior of the solution to the unknown homogenized problem.

The heterogeneous multiscale method (HMM) was pioneered by Weinan and Enquist [39, 40] and refers to a rich population of variants [3,15,16]. This method emphasizes important principles in the design of multiscale methods such as the choice of macro- and micro-models and their communication. For reviews and further references see [1,2,38] for a discussion of the HMM with special focus on PDEs.

Variational multiscale methods (VMM) have been developed in the 1990s by Hughes and collaborators, see [22,23]. The spirit of the method lies in a decomposition of the solution space and in the design of variational forms that reflect the relevant scale interactions between the spaces. Many versions of it exist and we point the reader to recent reviews [4,31] and the vast literature therein.

The present work is inspired by multiscale finite element methods (MsFEM) which can be seen as a special variant of the VMM. The idea of this method is to introduce subgrid variations into basis functions and can be dated back to works by Babuška et al. [6] and shares ideas with the partition of unity method [29]. The MsFEM in its current form was introduced in [12,20,21]. The essential idea of the method is to capture the local asymptotic

structure of the solution through adding problem dependent bubble correctors to a standard basis and use these as ansatz and trial functions. Many variations of this method exist and we refer the reader to [11,14] for a review.

Many of the afore-mentioned methods have the advantage that they work well for elliptic or parabolic problems and that they are accessible to an analysis. The difficulty in many applications on the other hand is their advection or reaction dominated character, i.e., the dynamics is often driven by low order terms. This poses major difficulties to numerical multiscale methods. Multiscale finite element methods naively applied will not converge to any reasonable solution since basis functions will exhibit artificial boundary layers [36] that are not present in the actual physical flow. Ideas to tackle this problem are based on combining transient multiscale methods with Lagrangian frameworks [36] or with stabilization methods for stationary problems, see [26] for an overview.

When it comes to comparing numerical multiscale methods with dominant advection it is important to distinguish between stationary problems and transient problems since the former usually describe balanced states of (possibly more complicated) transient problems. They therefore contain no information on initial states and one cannot take advantage of knowledge on how "information" is transported by the dynamics inherent to the model. Multiscale methods for stationary problems [3,20,21,26,28] therefore are conceptually different. To the best of our knowledge existing methods for transient problems usually aim at different regimes, e.g., see [16,27] and make some (legitimate) assumptions of the flow field specific to the application that motivated the authors. This regime dependency in turn influences the design of the method. Such regimes include periodic or quasi-periodic subscale coefficients, zero-mean flows, existence of scale separations, incompressible background flows or different limiting regimes. It is our goal to avoid such assumptions keeping in mind that processes that are advectively driven by flows in climate and weather applications have divergence, do not have mean zero and do not exhibit a scale separation. Yet, they dominate many parameterized processes that can manifest in precomputed diffusion coefficients coming from other sources, e.g., see [33,34] for a simulation environment of urban areas and canopies.

1.2 Contribution

Our main contribution is a framework of numerical methods for advection-dominated flows which by reconstructing subgrid variations on local basis functions aims at reflecting the local asymptotic structure of solutions correctly. The idea combines multiscale Galerkin methods with semi-Lagrangian methods by locally solving an inverse problem for the basis representation of solutions that is adapted to the actual flow scenario. We demonstrate the idea on one and two-dimensional advection–diffusion equations with heterogeneous background velocities and diffusivities in both non-conservative and conservative form.

Applying standard MsFEMs directly leads to failure since boundary conditions (BCs) for the modified basis functions cannot be prescribed arbitrarily. Suitable BCs on the subgrid scale are an essential ingredient for many multiscale methods. A wrong placement of Dirichlet BCs, for example, leads to boundary layers in basis functions that are not there in the real large scale flow, i.e., the way information propagates in advection-dominant flow needs to be respected and not artificially blocked. Other choices of BCs usually complicate the enforcement of conformity or obfuscate additional assumptions on the problem structure (such as local periodicity).

Conformal MsFEM techniques for advection-dominated tracer transport were already explored in our previous work in one spatial dimension on a transient advection–diffusion

equation [36]. The finding is that one has to follow a Lagrangian point of view on coarse scales such that flow is "invisible". On fine scales one can then simplify Lagrangian transforms in order to make advective effects milder near coarse grid boundaries without going to a fully Lagrangian setting. This amounts to prescribing Dirichlet boundary conditions on coarse flow characteristics and has the effect that basis functions do not develop spurious boundary layers that are not present in the actual flow. While this work gave some useful insights it is unfortunately not feasible for practical applications since it suffers from several weaknesses. First, it is not directly generalizable to higher dimensions. Secondly, it needs assumptions on the background velocity that are not necessarily fulfilled in practical applications to ensure that coarse scale characteristics do not merge.

In order to circumvent these problems we suggest here a new idea based on a semi-Lagrangian framework that locally in time constructs a multiscale basis on a fixed Eulerian grid. The construction is done in a semi-Lagrangian fashion on the subgrid scale whereas the macroscopic scale is conveniently treated fully Eulerian. This is in contrast to our previous work but still respects that information in advection-dominant flows is "mostly" propagated along flow characteristics. We do not make any special assumptions on the flow field with respect to scale separation, divergence or mean except that the flow-field be not "too badly" behaved. From the point of view of climate and weather applications this is reasonable since velocities are usually given on relatively coarse meshes.

The construction of the basis in each cell at time t^{n+1} is done by tracing each Eulerian cell back to time t^n. This yields a distorted cell. A basis on this distorted cell is then reconstructed by solving an inverse problem for the representation of the global solution at the previous time step. Then the local representation of the solution on this cell is propagated forward in time to get a basis on the Eulerian grid at t^{n+1} instead of propagating the solution itself.

All reconstructions of the basis in coarse cells are independent from each other and can be performed in parallel. The global time step with the modified basis is not critical since the coarse problem is small and since we can use algebraic constructs to make the assembly of the global problem efficient. Note that although we formulate the algorithm globally in an implicit form it can well be formulated explicitly which will allow for computational efficiency for the global step.

Our new approach has several advantages. First, the idea of our method is quite simple and allows to effectively incorporate subgrid behavior of the solution into the multiscale basis via the solution of local inverse problems. It is important to note that the statement that the idea of the method is simple rather refers to the idea of semi-Lagrangian reconstruction than to the actual computational effort that one needs to invest to locally reconstruct a basis.

Secondly, the method can be parallelized similarly to semi-Lagrangian methods and thus the actual basis reconstructions can be done on individual compute nodes and are independent of each other. This eases the computational overhead (which increases with the space dimension) the more compute nodes are involved (ideally, the method scales best when each coarse cell is owned on a separate compute node). We do not take advantage of parallelism here since our first implementation is a proto-type but in a forthcoming paper we will present an MPI parallel and locally threaded implementation (for a climate specific application) to investigate scaling on large clusters. However, this is ongoing work and we cannot report on it yet. Furthermore, the idea works in any dimension.

Numerical tests show that it is accurate in both L^2 and H^1 since it represents subgrid variability correctly. The method can handle problems that involve an additional reaction term. This consequently includes conservation problems since then $\nabla \cdot (\mathbf{c}_\delta u) = \mathbf{c}_\delta \cdot \nabla u + (\nabla \cdot \mathbf{c}_\delta) u$ in our model (2). A preliminary version of the idea was discussed in [35]. The preliminary version, however, does not contain algorithmic details and does not allow to reproduce the

method. Here we aim at a self contained clear presentation with details how inverse problems for basis reconstruction can be designed and with an extensive computational study on model problems that are chosen to be close to our climate and weather application scenarios, i.e., the flow does not exhibit a scale separation and is dominant in both compressible and incompressible regimes while (given) parameterized processes manifest in fine scale diffusion coefficients that often have high contrast. We also provide the reader with algorithmic details on the 2D method and the global assembly procedure in Sect. 2.1.

We would like to stress that the idea is generic and can be used for vector valued problems and problems that involve subgrid information coming from actual data which renders the technique suitable for data assimilation on both the coarse scale and the subgrid scale. Applying it to finite element frameworks with different elements such as conformal elements in $H(curl)$ or $H(div)$ is possible. Thus this manuscript should rather be seen as an instance of how the idea can be realized on a specific model problem whereas we picked the simulation of passive tracer transport.

2 The Semi-Lagrangian Basis Reconstruction in One and Two Spatial Dimensions

In this section we will outline our ideas on an advection–diffusion equation (ADE) with periodic boundaries as a model problem. We will outline all ideas for didactic purposes first in $d = 1$ and in $d = 2$ dimensions on

$$\partial_t u + \mathbf{c}_\delta \cdot \nabla u = \nabla \cdot (\mathbf{A}_\varepsilon \nabla u) + f \quad \text{in } \mathbb{T}^d \times [0, T] \sim [0, 1]^d \times [0, T]$$
$$u(\mathbf{x}, 0) = u_0(\mathbf{x})$$

(1)

and

$$\partial_t u + \nabla \cdot (\mathbf{c}_\delta u) = \nabla \cdot (\mathbf{A}_\varepsilon \nabla u) + f \quad \text{in } \mathbb{T}^d \times [0, T]$$
$$u(\mathbf{x}, 0) = u_0(\mathbf{x})$$

(2)

where $\mathbf{c}_\delta(\mathbf{x}, t)$ is the background velocity, $\mathbf{A}_\varepsilon(\mathbf{x}, t)$ is a matrix-valued diffusivity, and f and u_0 are some smooth external forcing and initial condition. $\mathbb{T}^d$ denotes the d-dimensional torus which is topologically equivalent to $[0, 1]^d$ with periodic boundary conditions. We will use bold letters for the vectors and tensors independent of the dimension. From the modelling point of view the difference between (1) and (2) is that in case of $\nabla \cdot c_\delta \neq 0$ and $f = 0$ that (2) is a conservation law. One can then see that since $\nabla \cdot (\mathbf{c}_\delta u) = c_\delta \cdot \nabla u + (\nabla \cdot c_\delta)u$ this means that (2) is the same as (1) apart from an additional reaction term compensating for "loss" of u.

The indices $\delta > 0$ and $\varepsilon > 0$ indicate that both quantities may have large variations on small scales that are not resolved on coarse scales $H > 0$ of our multiscale method. We will also work locally on a scale $h \ll H$ that can resolve the variations in the coefficients. Furthermore, we assume that $\mathbf{c}_\delta \gg \mathbf{A}_\varepsilon$ (see remark below) and that $\mathbf{c}_\delta$ is well-behaved, for example Lipschitz in space and continuous in time. This assumption is often satisfied in practice for example in climate simulations where $\mathbf{c}_\delta$ is given at the nodes of a coarse grid. Depending on the application variations in $\mathbf{c}_\delta$ may be resolved on scale H or not. The diffusivity tensor is assumed to be positive definite (uniformly in ε and point-wise in $\mathbf{x}$) with $\mathbf{A}_\varepsilon \in L_t^\infty(L_x^\infty)^{d \times d}$, and is often derived from parameterized processes such as a varying sea ice distribution, convection, topographical features, or land use patterns. Note, that (1) does not conserve the tracer u in contrast to Eq. (2) provided $f = 0$.
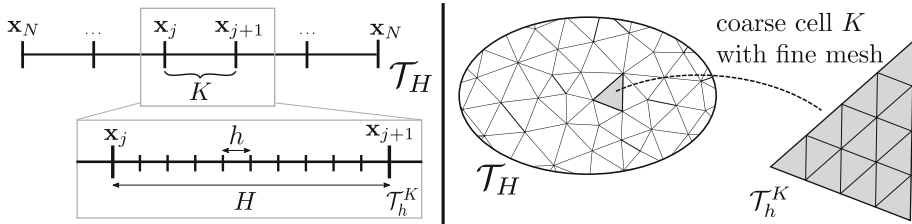
Fig. 1 Global coarse mesh and local fine mesh in 1D (left) and 2D (right)

Remark 1 Advection-dominance of a flow (what we sloppily expressed by $\mathbf{c}_\delta \gg \mathbf{A}_\varepsilon$) is usually expressed by a dimensionless number—the Péclet number Pe which is essentially the ratio between advective and diffusive time scales. There exist several versions of this number [24]. Since for large variations of the coefficients on the subgrid scale advection-dominance is a very local property we need be more precise with what me mean by that. Here we assume that Pe is high on average, i.e., $\text{Pe} = \frac{\|\mathbf{c}_\delta\|_{L^2} L}{\|\mathbf{A}_\varepsilon\|_{L^2}}$ where L is a characteristic length. We take L to be the length of the computational domain.

Although the general idea of reconstructing subgrid variability in basis functions is not substantially different for (1) and (2) there are some differences in the propagation of local boundary information described below that strongly influence the accuracy. We will explicitly describe these differences.

We start with outlining our method in one dimension since the idea is very simple and avoids complications that arise in higher dimensions. The idea is to represent non-resolved fine scale variations of the solution locally on a set of non-polynomial basis functions in each coarse cell. That is we fix two (Eulerian) meshes: A coarse mesh $\mathcal{T}_H$ of width H and on each cell $K \in \mathcal{T}_H$ of the coarse mesh we have a fine mesh $\mathcal{T}_h^K$ of width $h \ll H$ (we assume that h does not depend on K). On the coarse mesh we have a multiscale basis $\varphi_i^{H,\text{ms}} \colon \mathcal{T}_H \times [0, T] \to \mathbb{R}$, $i = 1, \ldots, N_H$, where N_H is the number of nodes of $\mathcal{T}_H$. This basis depends on space and time and will be constructed so that we will obtain a spatially H^1-conformal (multiscale) finite element space. Note that this is a suitable space for problems (1) and (2) since they have unique solutions $u \in L^2([0, T], H^1(\mathbb{T}^d))$ with $\partial_t u \in L^2([0, T], H^{-1}(\mathbb{T}^d))$ and hence in $u \in C([0, T], L^2(\mathbb{T}^d))$ [13]. The initial condition $\mathbf{u}^0$ can therefore be assumed to be in $L^2(\mathbb{T}^d)$ but in later experiments we will choose it to be smooth. The fine mesh on each cell $K \in \mathcal{T}_H$ is used to represent the basis locally, see Fig. 1.

Standard MsFEM methods used for porous media flows are designed in such a way that the coarse scale basis solves the PDE model locally with the same boundary conditions as standard FEM basis functions. Note that the global coarse scale MsFEM solution is a linear combination of modified local basis functions that do resolve the fine scale structure induced by the problem's heterogeneities and therefore do represent the asymptotics correctly. This works for stationary elliptic problems and for parabolic problems as long as there is no advective term involved or even dominant. The reason is that advective terms as they appear in our model problems prevent a basis constructed by a standard MsFEM technique to represent the correct asymptotics. This is since flow of information is artificially blocked at coarse cell boundaries. Due to the incorrect boundary conditions for the multiscale basis artificial steep boundary layers form in basis functions that do not exist in the actual global flow. For transient problems another difficulty is that the local asymptotics around a point depends on the entire domain of dependence of this point and hence subscale information

represented by a base function must contain memory of the entire history of the local domain of that base function.

A first attempt to bypass these difficulties was to pose boundary conditions for the basis on suitable space-time curves, across which naturally no information is propagated on the coarse scales. Such curves are Lagrangian paths. Therefore, a Lagrangian framework on the coarse scale and an "almost Lagrangian" setting on fine scales was proposed by us earlier [36]. Unfortunately, this method is not feasible since it can not be generalized to higher dimensions and needs restrictive assumptions on the flow field.

Nonetheless, the results of [36] suggest that a coarse numerical splitting of the domain should correspond to a reasonable physical splitting of the problem. Instead of a fully Lagrangian method on coarse scales semi-Lagrangian techniques for building the basis can circumvent the difficulties of Lagrangian techniques. But these are only local in time and therefore they do not take into account the entire domain of dependence of a point. We show how to deal with this in the following. First, we start with the global problem.

2.1 The Global Time Step in 1D/2D

Suppose we know a set of multiscale basis functions in a conformal finite element setting, i.e., we approximate the global solution at each time step in a spatially coarse subspace $V^H(t) \subset H^1(\mathbb{T}^d)$ in which the solution u is sought (almost everywhere). We denote this finite-dimensional subspace as

$$V^H(t) = \text{span}\left\{ \varphi_j^{H,\text{ms}}(\cdot, t) \middle| i = 1, \ldots, N_H \right\}. \tag{3}$$

First, we expand the solution $u^H(\mathbf{x}, t)$ in terms of the basis at time $t \in [0, T]$, i.e.,

$$u^H(\mathbf{x}, t) = \sum_{j=0}^{N_H} u_j^H(t) \varphi_j^{H,\text{ms}}(\mathbf{x}, t). \tag{4}$$

Then we test with the modified basis and integrate by parts. Therefore, the spatially discrete version of both problem (1) and (2) becomes the ODE
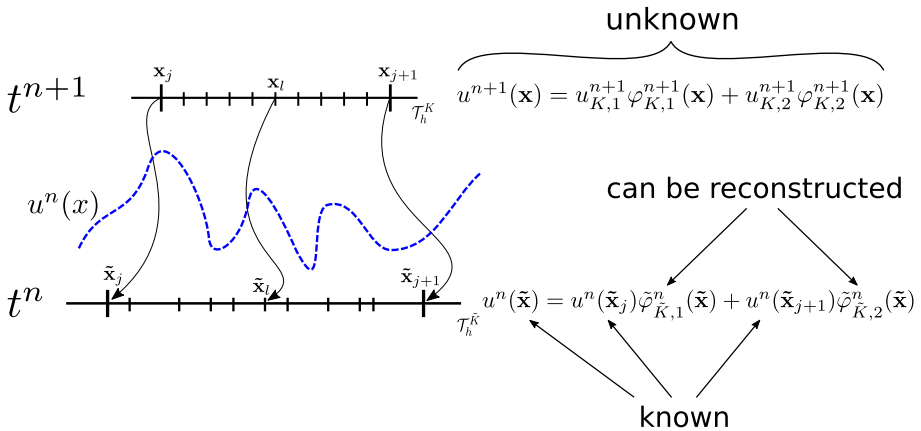
$$\mathbf{M}(t) \frac{\mathrm{d}}{\mathrm{d}t} \mathbf{u}^H(t) + \mathbf{N}(t) \mathbf{u}^H(t) = \mathbf{A}(t) \mathbf{u}^H(t) + \mathbf{f}^H(t)$$
$$\mathbf{u}^H(0) = \mathbf{u}^{H,0} \tag{5}$$

where

$$\mathbf{A}_{ij}(t) = \int_{\mathbb{T}^d} \varphi_i^{H,\text{ms}}(\mathbf{x}, t) \cdot \mathbf{A}_\varepsilon(\mathbf{x}, t) \nabla \varphi_j^{H,\text{ms}}(\mathbf{x}, t) \, \mathrm{d}\mathbf{x}$$
$$- \int_{\mathbb{T}^d} \varphi_i^{H,\text{ms}}(\mathbf{x}, t) \left(\mathbf{c}_\delta(\mathbf{x}, t) \cdot \nabla \right) \varphi_j^{H,\text{ms}}(\mathbf{x}, t) \, \mathrm{d}\mathbf{x} \tag{6}$$

for (1) and

$$\mathbf{A}_{ij}(t) = \int_{\mathbb{T}^d} \varphi_i^{H,\text{ms}}(\mathbf{x}, t) \cdot \mathbf{A}_\varepsilon(\mathbf{x}, t) \nabla \varphi_j^{H,\text{ms}}(\mathbf{x}, t) \, \mathrm{d}\mathbf{x}$$
$$+ \int_{\mathbb{T}^d} \left(\nabla \varphi_i^{H,\text{ms}}(\mathbf{x}, t) \cdot \mathbf{c}_\delta(\mathbf{x}, t) \right) \varphi_j^{H,\text{ms}}(\mathbf{x}, t) \, \mathrm{d}\mathbf{x} \tag{7}$$

Fig. 2 The fine mesh in each cell $K \in \mathcal{T}_H$ is traced back one time step where the known solution can be used to reconstruct a basis representation of the solution

for (2). The mass matrix is given by

$$\mathbf{M}_{ij}(t) = \int_{\mathbb{T}^d} \varphi_i^{H,\mathrm{ms}}(\mathbf{x}, t) \varphi_j^{H,\mathrm{ms}}(\mathbf{x}, t) \, d\mathbf{x}, \tag{8}$$

$\mathbf{f}^H(t)$ contains forcing and boundary conditions and the initial condition $\mathbf{u}^{H,0}$ is the projection of $\mathbf{u}^0 \in L^2(\mathbb{T}^d)$ onto $V^H(0)$. Note that (5) contains a derivative of the mass matrix:

$$\mathbf{N}_{ij}(t) = \int_{\mathbb{T}^d} \varphi_i^{H,\mathrm{ms}}(\mathbf{x}, t) \partial_t \varphi_j^{H,\mathrm{ms}}(\mathbf{x}, t) \, d\mathbf{x}. \tag{9}$$

This is necessary since the basis functions depend on time and since we discretized in space first. The reader will notice that using Rothe's method of lines, i.e., discretizing in time first, it is not clear what basis function to use for testing and this a priori leads to a different linear system.

For the time discretization we simply use the implicit Euler method. The discrete ODE then reads

$$\mathbf{M}(t^n)\mathbf{u}^{n+1} = \mathbf{M}(t^n)\mathbf{u}^n + \delta t \left[\mathbf{A}(t^{n+1})\mathbf{u}^{n+1} - \mathbf{N}(t^{n+1})\mathbf{u}^{n+1} + \mathbf{f}^H(t^n) \right]. \tag{10}$$

Other time discretization schemes, in particular, explicit schemes are of course possible. For didactic reasons we choose to present the algorithm in an implicit version. We will come back to that later. The next step is to show how to construct the multiscale basis.

Convention In both 1D and 2D we use bold letters like $\mathbf{x}$, $\mathbf{A}$ to express potentially vector valued quantities although they would be scalar in one dimension in all formulas and figures. We do so since we would like to avoid confusing the reader with different notations that are merely due to different dimensionality although the basic ideas are the same. Difficulties that arise in more than one dimension will be pointed out explicitly. For the sake of consistency we also use ∇ instead of ∂_x in one dimension. Quantities marked with a tilde like $\tilde{\mathbf{x}}$ signalize (semi-)Lagrangian quantities.

Remark 2 The assembly of the global system matrices or a cell-wise decomposition of their action as done in matrix-free FEM frameworks in high-performance computing software can efficiently be represented as matrix-vector multiplications. To see this let $K \in \mathcal{T}_H$ and let

Fig. 3 2D illustration similar to the one shown in Fig. 2. Each coarse cell is together with its fine mesh being traced back one time step at which the known global solution can be used to reconstruct a basis
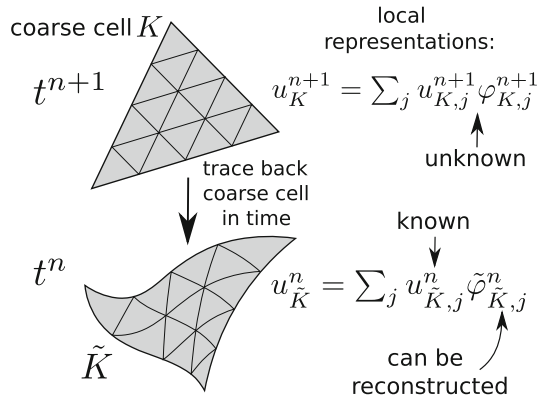
$\varphi_i^{H,\text{ms}}$ and $\varphi_j^{H,\text{ms}}$ be two reconstructed multiscale basis functions that are supported on K at nodes with global degrees of freedom i, j and with local numbering $\alpha(i), \alpha(j)$. In the subsequent presentation such basis functions will be given on the fine meshes in terms of their (fine scale) degrees of freedom so let further φ_i^k be the kth nodal value on the fine mesh $\mathscr{T}_h^K$ of basis function i. If we were, for example, to assemble the element mass matrix contributed to $\mathbf{M}_{ij}$ in (8) by the element K a simple calculation shows that

$$\left(\mathbf{M}_{ij}^K\right)_{\alpha(i),\alpha(j)} = \int_K \varphi_i^{H,\text{ms}}(\mathbf{x}, t)\varphi_j^{H,\text{ms}}(\mathbf{x}, t)\,\mathrm{d}\mathbf{x} \tag{11}$$
$$= \varphi_i^k \mathbf{M}_{kl}^K \varphi_j^l$$

where $\mathbf{M}_{kl}^K$ is the (fine scale) mass matrix on $\mathscr{T}_h^K$. This matrix will be computed anyway (see the subsequent discussion) so one can store it and use it for the assembly in a more efficient way instead of looping first over all fine cells in each coarse. On a single machine storing local matrices can potentially become a bottleneck but this is less of a problem on large clusters. Even the use of matrix-free methods is possible. Since this work is focused on the method we pass on these algorithmic details.

2.2 The Reconstruction Mesh in 1D/2D

Our idea combines the advantage of both semi-Lagrangian and multiscale methods to account for dominant advection. The reconstruction method is based on the simple observation that local information of the entire domain of dependence is still contained in the global solution at the previous time step. This can be used to construct an Eulerian multiscale basis: we trace back an Eulerian cell $K \in \mathscr{T}_H$ at time t^{n+1} on which the solution and the basis are unknown to the previous time step t^n. This gives a distorted cell $\tilde{K}$ over which the solution u^n is known but not the multiscale basis $\tilde{\varphi}_i, i = 1, 2$.

In order to find the points where transported information originates we trace back all nodes in $\mathscr{T}_h^K$ from time t^{n+1} to t^n. For this one simply needs to solve an ODE with the time-reversed velocity field that reads

$$\frac{\mathrm{d}}{\mathrm{d}t}\tilde{\mathbf{x}}_l(t) = -\mathbf{c}_\delta(\tilde{\mathbf{x}}_l(t), -t), \quad t \in [-t^{n+1}, -t^n] \tag{12}$$
$$\tilde{\mathbf{x}}_l(-t^{n+1}) = \mathbf{x}_l$$
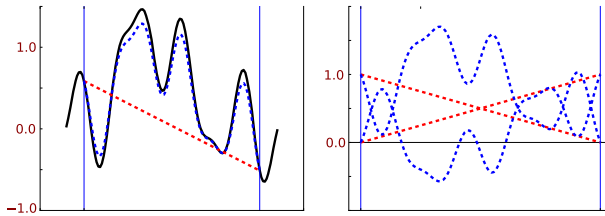
Fig. 4 Left: Illustration of an oscillatory function (black) being approximated by a standard linear basis (red) on a single cell compared to a modified basis (blue) that solves (14). The regularization parameters were taken as $\alpha_i = 0.1$. Right: Comparison of the standard basis to the modified basis. The modified basis does not constitute a partition of unity (colour figure online)

for each $\mathbf{x}_l$ and then take $\tilde{\mathbf{x}}_l = \tilde{\mathbf{x}}_l(-t^n)$, see Figs. 2 and 3 for an illustration. This procedure is standard in semi-Lagrangian schemes and can be parallelized.

Remark 3 In semi-Lagrangian schemes it is possible that (if Courant numbers are large) cells may nearly collapse or flip when tracing them back in time. In practice this can be prevented by decreasing the time step $\delta t = t^{n+1} - t^n$ according to the (cell-wise) Jacobian of the Lagrangian transform induced by the flow. In our numerical experiments, however, we did not choose our time step too large so that we did not run into this problem.

2.3 Basis Reconstruction in 1D

After tracing back each point $\mathbf{x}_l$ of $K \in \mathscr{T}_H$ to its origin $\tilde{\mathbf{x}}_l$ in a distorted coarse cell $\tilde{K} \in \mathscr{T}_H$ we need to reconstruct a local representation of the (known) solution u^n on $\tilde{K}$:

$$u^n(\mathbf{x})\big|_{\tilde{K}} = u^n(\tilde{\mathbf{x}}_j)\tilde{\varphi}_{K,1}(\mathbf{x}) + u^n(\tilde{\mathbf{x}}_{j+1})\tilde{\varphi}_{K,2}(\mathbf{x}) \tag{13}$$

where $\tilde{\mathbf{x}}_j$ and $\tilde{\mathbf{x}}_{j+1}$ are the boundary points of $\tilde{K}$. In one dimension one can of course choose a representation using the standard basis of hat functions but this would not incorporate subgrid information at step t_n at all. Even in the forward advection step (explained below) that is being carried out in the next step subgrid information on the basis will not contain the correct subgrid variability because the information contained in the basis does not take into account the entire domain of dependence of K. We choose to bypass this problem by solving an inverse problem for the basis to adjusts the representation. The idea is to fit a linear combination of the basis locally such that u^n is optimally represented, i.e., we solve

$$\underset{\tilde{\varphi}_{K,i} \in V(\tilde{K})}{\text{minimize}} \quad \left\| u^n - \left(u^n_j \tilde{\varphi}_{\tilde{K},1} + u^n_{j+1} \tilde{\varphi}_{\tilde{K},2} \right) \right\|^2_{L^2(\tilde{K})} + \sum_i \alpha_i \mathscr{R}_i(\tilde{\varphi}_{\tilde{K},i})$$

$$\text{s.t.} \quad u^n_j = u^n(\tilde{\mathbf{x}}_j), \quad u^n_{j+1} = u^n(\tilde{\mathbf{x}}_{j+1}) \tag{14}$$

$$\varphi_{\tilde{K},1}(\tilde{\mathbf{x}}_j) = \varphi_{\tilde{K},2}(\tilde{\mathbf{x}}_{j+1}) = 1$$

$$\varphi_{\tilde{K},1}(\tilde{\mathbf{x}}_{j+1}) = \varphi_{\tilde{K},2}(\tilde{\mathbf{x}}_j) = 0.$$

The functions $\mathscr{R}_i : V(\tilde{K}) \to \mathbb{R}$ are regularizers weighted by positive numbers $\alpha_i \in \mathbb{R}$ and V is a suitable function space. In practice we choose V to be a finite-dimensional subspace (which is then closed) of H^1. A simple regularizer that we found useful in one spatial dimension is a penalization of the deviation of the modified basis function from the standard linear basis

function with the same boundary values in the quadratic mean, i.e., we use

$$\mathscr{R}_i(\tilde{\varphi}_{\tilde{K},i}) = \left\| \tilde{\varphi}_{\tilde{K},i} - \tilde{\varphi}^0_{\tilde{K},i} \right\|^2_{H^1} \tag{15}$$

where $\tilde{\varphi}^0_{\tilde{K},i}$ denotes the i-th standard (linear) basis on $\tilde{K}$. Note that the problem is convex and hence there exists a solution. In finite dimensions this amounts to a system of linear equations that can be computed explicitly. In a spatially discrete version this system will be small and cheap to solve. A suitable choice of a regularizer depends on the problem at hand and prevents instabilities in the global time step. For problems in two dimensions we found a different regularizer to be useful (see below). Figure 4 illustrates the effect of a local reconstruction of a basis compared to a representation with a standard basis.

2.4 Basis Propagation in 1D

After having reconstructed a suitable basis on each coarse cell $\tilde{K}$ we have an H^1-conformal basis. This basis, however is a basis at time step t^n and does not live on the coarse Eulerian grid $\mathscr{T}_H$ that we initially fixed. The step to take now is to construct a basis at t^{n+1} on $\mathscr{T}_H$. This is done similarly to [36], i.e., we evolve the basis according to the model at hand with a vanishing external forcing. Note, however, that we compute the basis at t^{n+1} along Lagrangian trajectories starting from t^n, i.e., we need to transform the original model. Equation (1) becomes

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\varphi_{K,i} &= \tilde{\nabla} \cdot \left(\tilde{\mathbf{A}}_\varepsilon \tilde{\nabla} \varphi_{K,i} \right) \quad \text{in } \tilde{K} \times [t^n, t^{n+1}] \\
\varphi_{K,i}(\tilde{\mathbf{x}}_j, t) &= \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}}_j), \quad t \in [t^n, t^{n+1}] \\
\varphi_{K,i}(\tilde{\mathbf{x}}_{j+1}, t) &= \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}}_{j+1}), \quad t \in [t^n, t^{n+1}] \\
\varphi_{K,i}(\tilde{\mathbf{x}}, t^n) &= \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}})
\end{aligned} \tag{16}$$

and Eq. (2) transforms into

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\varphi_{K,i} + \left(\tilde{\nabla} \cdot \tilde{\mathbf{c}}_\delta \right) \varphi_{K,i} &= \tilde{\nabla} \cdot \left(\tilde{\mathbf{A}}_\varepsilon \tilde{\nabla} \varphi_{K,i} \right) \quad \text{in } \tilde{K} \times [t^n, t^{n+1}] \\
\varphi_{K,i}(\tilde{\mathbf{x}}_j, t) &= \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}}_j), \quad t \in [t^n, t^{n+1}] \\
\varphi_{K,i}(\tilde{\mathbf{x}}_{j+1}, t) &= \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}}_{j+1}), \quad t \in [t^n, t^{n+1}] \\
\varphi_{K,i}(\tilde{\mathbf{x}}, t^n) &= \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}}).
\end{aligned} \tag{17}$$

where $\tilde{\nabla}$ denotes the gradient in Lagrangian coordinates. Note that these evolution equations are solved on $\tilde{K}$, i.e., on the element $K \in \mathscr{T}_H$ traced back in time. Advection is "invisible" in these coordinates. The end state $\varphi_{K,i}(\tilde{\mathbf{x}}, t^{n+1})$ on $\tilde{K}$ can then be transformed onto the Eulerian element $K \in \mathscr{T}_H$ to obtain the desired basis function $\varphi_{K,i}(\mathbf{x}, t^{n+1}) \sim \varphi^{n+1}_{K,i}(\mathbf{x})$ at the next time step. Corresponding basis functions in neighboring cells can then be glued together to obtain a modified global basis $\varphi^{H,\mathrm{ms}}_i$, $i = 1, \ldots, N_H$. This way we get a basis of a subspace of H^1 that is neither a partition of unity nor is it necessarily positive. Nonetheless, it is adjusted to the problem and the data at hand. The propagation step is illustrated in Fig. 5.

Using our method we reconstruct and advect the representation of the global solution first and then the solution itself using the modified representation. The global step is completely Eulerian while the local reconstruction step is semi-Lagrangian in contrast to [36] where the global step is Lagrangian and the local step is "almost"-Lagrangian. The essential

Fig. 5 The basis reconstructed according to (14) at time t^n is propagated forward to time t^{n+1} according to (16) or (17)
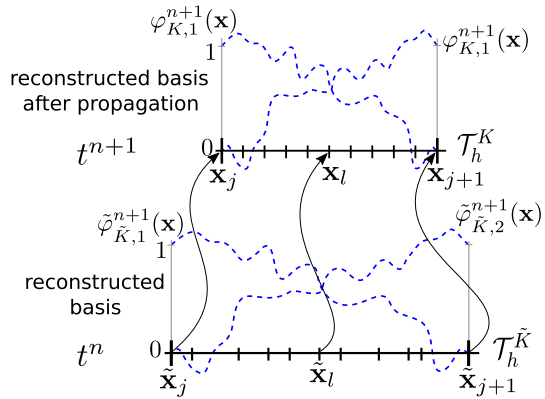
Algorithm 1 Sketch of reconstruction algorithm for a 1D problem. Implicit version using backward Euler. Note that all loops over cells $K \in \mathcal{T}_H$ can be parallelized.

---

input   : Problem parameters for (1) or (2), H, h, δt
output: Weights $\mathbf{u}^H(t_j)$ of multiscale solution and set of multiscale basis functions $\varphi_i^{H,\mathrm{ms}}$

1 **begin**
2 | Initialize a coarse mesh $\mathcal{T}_H$
3 | On each cell $K \in \mathcal{T}_H$ initialize a fine mesh $\mathcal{T}_h^K$
   | /* We need to reconstruct a basis at time t^0. Note that no
   | trace back and no propagation are needed.   */
4 | **for** $K \in \mathcal{T}_H$ **do in parallel**
5 | | Reconstruct the optimal basis representation of $u_0(x)|_K$ according to (14)
6 | **end**
   | /* Now the time stepping starts. We compute the solution
   | at t^{n+1}.   */
7 | **for** $n = 0$ **to** $n \leq N_{\mathrm{steps}}$ **do**
8 | | **for** $K \in \mathcal{T}_H$ **do in parallel**
9 | | | Trace back each node in K one time step from t^{n+1} to t^n according to (12)
10 | | | Reconstruct the optimal basis representation of $u^n(x)|_{\tilde{K}}$ according to (14)
11 | | | Propagate the optimal basis forward onto K according to (16) or (17)
12 | | **end**
13 | | Assemble the global (coarse) system matrices for (10)
14 | | Make a global backward Euler time step using (10)
15 | **end**
16 | Postprocess the solution
17 | Return
18 **end**

---

steps described above are summarized in Algorithm 1. Note that the steps to reconstruct the multiscale basis are embarrassingly parallel and consist of small local problems.

2.5 Basis Reconstruction in 2D

The basis reconstruction step in two dimensions is slightly different. We separate the description of the procedure from the reconstruction in one dimension to make the reader aware of difficulties that arise when increasing the space dimension. Keeping the difficulties in mind one can then easily generalize the procedure even to 3D. We will comment on that again later. It is worth mentioning that these difficulties do not need to arise in non-conformal settings but since we agreed on a conformal method we will be consistent here.

To construct a H^1-conformal basis we need to ensure that the reconstructed global basis is continuous across coarse cell boundaries. This can be achieved by first looping over all coarse edges of the traced back mesh and reconstructing the solution at the previous time step t^n with a basis representation on each edge, i.e., we solve first an inverse problem of the kind (14).

$$
\begin{aligned}
\underset{\tilde{\varphi}_{\tilde{\Gamma},i} \in V(\tilde{\Gamma})}{\text{minimize}} \quad & \left\| u^n|_{\tilde{\Gamma}} - \left(u_1^n \tilde{\varphi}_{\tilde{\Gamma},1} + u_2^n \tilde{\varphi}_{\tilde{\Gamma},2} \right) \right\|_{L^2(\tilde{\Gamma})}^2 + \sum_i \alpha_i \mathscr{R}_i(\tilde{\varphi}_{\tilde{\Gamma},i}) \\
\text{s.t.} \quad & u_j^n = u^n(\tilde{\mathbf{x}}_j), \quad \tilde{\mathbf{x}}_j \text{ is end point of (traced back) edge } \tilde{\Gamma} \\
& \varphi_{\tilde{\Gamma},i}(\tilde{\mathbf{x}}_j) = \delta_{ij}.
\end{aligned}
\tag{18}
$$

Note that the regularizer (15) needs to be replaced since the edge can be curved (it is unclear what $\tilde{\varphi}_{K,i}^0$ in this case is). We use a harmonic prior for the reconstructed basis

$$
\mathscr{R}_i(\tilde{\varphi}_{\tilde{\Gamma},i}) = \left\| \Delta_{g(\tilde{\Gamma})} \tilde{\varphi}_{\tilde{\Gamma},i} \right\|_{L^2(\tilde{\Gamma})}^2
\tag{19}
$$

with a low weight α_i in (14). The operator $\Delta_{g(\tilde{\Gamma})}$ denotes the Laplace–Beltrami operator induced by the standard Laplace operator with the trace topology of the respective (traced back) edge $\tilde{\Gamma}$, i.e., $g(\tilde{\Gamma})$ is the metric tensor. We pass on providing details here.

The edge reconstructed basis then serves as boundary value for the cell basis reconstruction. The optimization problem to solve on the traced back cell $\tilde{K}$ then reads

$$
\begin{aligned}
\underset{\tilde{\varphi}_{\tilde{K},i} \in V(\tilde{K})}{\text{minimize}} \quad & \left\| u^n - \sum_{j=1}^3 u_j^n \tilde{\varphi}_{\tilde{K},j} \right\|_{L^2(\tilde{K})}^2 + \sum_i \alpha_i \mathscr{R}_i(\tilde{\varphi}_{\tilde{K},i}) \\
\text{s.t.} \quad & u_j^n = u^n(\tilde{\mathbf{x}}_j), \quad \tilde{\mathbf{x}}_j \text{ is corner point of (traced back) cell } \tilde{K} \\
& \varphi_{\tilde{K},j}|_{\tilde{\Gamma}_l} = \varphi_{\tilde{\Gamma},j_l},
\end{aligned}
\tag{20}
$$

where we also used a Laplacian regularizer. However, the essential task is to ensure conformity of the global basis by first reconstructing representations on all edges and then inside the cells $\tilde{K}$.

In three dimensions it would be necessary to first reconstruct edge then face and only then the interior representations. This sounds potentially expensive but it is embarrassingly parallel since all reconstructions are independent.

2.6 Basis Propagation in 2D

Again, for didactic reasons we make the reader aware of differences to the propagation of the basis in contrast to one spatial dimension. As in Sect. 2.5 we need to preserve conformity

Fig. 6 Illustration of a reconstructed multiscale basis in 2D
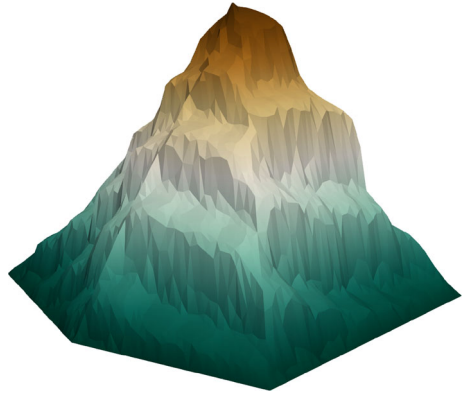
of the global basis. This can easily be done by fixing the boundary values in the propagation step of the basis functions similar to the propagation in 1D, see Sect. 2.4.

However, this strategy can result in numerical errors from two sources. Recall that the goal of our multiscale method is to represent the local asymptotic structure of the solution correctly by putting subgrid variations into the basis. This can be interpreted as adding a non-polynomial corrector function in each coarse cell to a standard FEM basis which changes the boundary conditions. These boundary conditions are themselves subject to evolution and keeping them fixed in time will not reflect any system intrinsic dynamics.

The first source of a numerical error is a resonance error. This error usually becomes dominant if the scale of oscillations in the diffusion term is close to the scales resolved by the coarse grid. This is well-documented in the literature on multiscale FEMs for stationary elliptic problems that are usually not dominated by lower order terms. Note, that in practical problems it is often unclear if a scale separation exists or it can not be quantified which makes it difficult to identify a resonance regime [17,21].

Another source for not capturing the asymptotic structure with fixed boundary values even if there was a scale separation between coarse mesh and diffusion oscillations (which can not be guaranteed) appears in the conservative case (2). This is due to the reactive term

$$\left(\tilde{\nabla} \cdot \tilde{\mathbf{c}}_\delta \right) \varphi_{K,i} \tag{21}$$

that appears also in (17). This term compensates divergence, i.e., it triggers a reaction on the boundary wherever the divergence of the background velocity does not vanish. Essentially that means that this term is very local but can be quite strong, i.e., it is of the order of δ^{-1} if $\mathbf{c}$ is of order 1 but oscillates on a scale of order δ. This term must be taken into account when evolving a reconstructed basis (Fig. 6).

Fortunately, both errors can be dealt with by employing two strategies. The first one is oversampling, i.e., the local reconstruction and basis propagation on a coarse cell K is performed on a slightly larger domain than $K^{\mathrm{os}} \supset K$. After that the basis is being truncated to the relevant domain K. Using this strategy is generally possible but results in a non-conformal technique and hence we will not use it here [17]. Instead we will solve a reduced evolution problem on the edges that predicts the boundary values and then evolve the reconstructed interior basis for which we then know the boundary values in time. The edge evolution strategy is in general only necessary in the case of a missing scale separation for the diffusion as explained above or in the conservative case. Note that both errors do not appear in one dimension. We summarize this discussion below.

Algorithm 2 Sketch of modified reconstruction algorithm for a 2D problem. Implicit version using backward Euler.

---

input : Problem parameters for (1) or (2), H, h, δt

output: Weights $\mathbf{u}^H(t_j)$ of multiscale solution and set of multiscale basis functions $\varphi_i^{H,\text{ms}}$

19 **begin**

20   Initialize a coarse mesh $\mathscr{T}_H$

21   On each cell $K \in \mathscr{T}_H$ initialize a fine mesh $\mathscr{T}_h^K$

      `/* We need to reconstruct a basis at time` t^0`. Note that no`
        `trace back and no propagation are needed.`   `*/`

22   **for** $K \in \mathscr{T}_H$ **do in parallel**

23   Reconstruct the optimal basis representation of $u_0(x)|_K$ according to (14)

24   **end**

      `/* Now the time stepping starts. We compute the solution`
        `at` t^{n+1}`.`   `*/`

25   **for** $n = 0$ **to** $n \leq N_{\text{steps}}$ **do**

26   **for** $K \in \mathscr{T}_H$ **do in parallel**

27   Trace back each node in K one time step from t^{n+1} to t^n according to (12)

28   Reconstruct the boundary condition of optimal basis representation of $u^n(x)|_{\tilde{K}}$ according to (18), see Section 2.5

29   Reconstruct the optimal basis representation of $u^n(x)|_{\tilde{K}}$ according to (20), see Section 2.5

30   **if** *Problem is not in conservation form* **then**

31   Propagate the optimal basis forward onto K according to (22)

32   **else**

33   Propagate the boundary conditions of the optimal basis forward onto K according to (23)

34   Propagate the optimal basis forward onto K according to (24)

35   **end**

36   **end**

37   Assemble the global (coarse) system matrices for (10)

38   Make a global backward Euler time step using (10)

39   **end**

40   Postprocess the solution

41   Return

42 **end**

Strategy Without Edge Evolution In this case we keep the boundary values of the basis that we reconstructed at the previous time step fixed. Similarly to (16) this amounts to solving the evolution problem

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\varphi_{K,i} &= \tilde{\nabla} \cdot \left(\tilde{\mathbf{A}}_\varepsilon \tilde{\nabla}\varphi_{K,i}\right) \quad \text{in } \tilde{K} \times [t^n, t^{n+1}] \\
\varphi_{K,i}(\cdot, t)|_{\tilde{\Gamma}_l} &= \tilde{\varphi}_{\tilde{\Gamma}_l,i}, \quad t \in [t^n, t^{n+1}] \\
\varphi_{K,i}(\tilde{\mathbf{x}}, t^n) &= \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}})
\end{aligned}
\tag{22}
$$

for the ith basis function.

Strategy with Edge Evolution This method requires a prediction of the boundary values. The prediction is done by solving a reduced evolution problem

$$\frac{d}{dt}\varphi_{\Gamma,i} + \left(\tilde{\nabla} \cdot \tilde{\mathbf{c}}_\delta^{\text{red}}\right)\varphi_{\Gamma,i} = \tilde{\nabla} \cdot \left(\tilde{\mathbf{A}}_\varepsilon^{\text{red}}\tilde{\nabla}\varphi_{\Gamma,i}\right) \quad \text{on } \tilde{\Gamma} \times [t^n, t^{n+1}]$$

$$\varphi_{K,i}(x_j, t) = \delta_{ij}, \quad t \in [t^n, t^{n+1}], \ \tilde{\mathbf{x}}_j \text{ is end point of edge } \tilde{\Gamma} \qquad (23)$$

$$\varphi_{\Gamma,i}(\tilde{\mathbf{x}}, t^n) = \tilde{\varphi}_{\tilde{\Gamma},i}(\tilde{\mathbf{x}})$$

on each edge $\tilde{\Gamma}$ of $\tilde{K}$. The reduced quantities $\tilde{\mathbf{c}}_\delta^{\text{red}}$ and $\tilde{\mathbf{A}}_\varepsilon^{\text{red}}$ only contain the tangential parts of their counterparts $\tilde{\mathbf{c}}_\delta$ and $\tilde{\mathbf{A}}_\varepsilon$. Once this evolution problem is solved for each edge $\tilde{\Gamma}$ the boundary values for the full evolution problem on $\tilde{K}$ are known and one can solve

$$\frac{d}{dt}\varphi_{K,i} + \left(\tilde{\nabla} \cdot \tilde{\mathbf{c}}_\delta\right)\varphi_{\Gamma,i} = \tilde{\nabla} \cdot \left(\tilde{\mathbf{A}}_\varepsilon \tilde{\nabla}\varphi_{K,i}\right) \quad \text{in } \tilde{K} \times [t^n, t^{n+1}]$$

$$\varphi_{K,i}(\cdot, t)|_{\tilde{\Gamma}_l} = \tilde{\varphi}_{\tilde{\Gamma}_l,i}(\cdot, t), \quad t \in [t^n, t^{n+1}] \qquad (24)$$

$$\varphi_{K,i}(\tilde{\mathbf{x}}, t^n) = \tilde{\varphi}_{\tilde{K},i}(\tilde{\mathbf{x}})$$

to get the ith basis.

In both cases note again that the evolution Eqs. (22) and (24) are solved on $\tilde{K}$. The end state $\varphi_{K,i}(\tilde{\mathbf{x}}, t^{n+1})$ on $\tilde{K}$ can again be mapped onto the Eulerian element $K \in \mathscr{T}_H$ to obtain the desired basis function $\varphi_{K,i}(\mathbf{x}, t^{n+1}) \sim \varphi_{K,i}^{n+1}(\mathbf{x})$ at the next time step.

The described modifications for 2D problems are summarized in Algorithm 2. Note that the number of local problems can be quite large and one needs to be careful in designing a true parallel algorithm to avoid unnecessary communication between nodes. Ideally, one would like to use a number of processors that is of the order of the number of coarse cells. This number can be very large but super computers with several million cores do already exist so it does not exceed today's capacities.

3 Numerical Examples in 1D

We will show several 1D examples in a non-conservative and conservative setting according to (1) and (2), respectively. For all 1D tests we use a Gaussian

$$u_0(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp{-\frac{(x-\mu)^2}{2\sigma^2}}. \qquad (25)$$

with variance $\sigma = 0.1$ centered in the middle of the domain $\mathbb{T}^1$, i.e., $\mu = 0.5$. The end time is set to $T = 1$ with a time step $\delta t = 1/300$. We show our semi-Lagrangian multiscale reconstruction method (SLMsR) with a coarse resolution $H = 10^{-1}$ in comparison to a standard FEM with the same resolution and high order quadrature. As a reference method we choose a high-resolution standard FEM with $h_{\text{ref}} = 10^{-3}$. For the multiscale method we choose a fine mesh $\mathscr{T}_h^K$ with $h = 10^{-2}$ in each coarse cell $K \in \mathscr{T}_H$.

Test 1 We start with four tests showing the capability of the SLMsR to capture subgrid variations correctly. Note that the coarse standard FEM has as many cells the SLMsR has coarse cells and that it does not capture subgrid variations in the following tests. The resolution for the reference solution resolves all subgrid variations but the reader should keep in mind that practical applications do not allow the application of high-resolution methods. The coefficients

$$c_\delta(x, t) = \frac{1}{4} + \frac{1}{2} \cos(10\pi x) + \frac{1}{4} \cos(74\pi x) + \frac{3}{20} \cos(196\pi x)$$
$$A_\varepsilon(x, t) = 10^{-3} + 9 \cdot 10^{-4} \cos(10\pi t) \cos(86\pi x) \tag{26}$$

and

$$c_\delta(x, t) = \frac{1}{2} \cos(2\pi t) + \frac{1}{4} \cos(6\pi t) \cos(8\pi x) + \frac{1}{8} \cos(4\pi t) \cos(62\pi x)$$
$$+ \frac{1}{8} \cos(150\pi x) \tag{27}$$
$$A_\varepsilon(x, t) = 10^{-3} + 9 \cdot 10^{-4} \cos(10\pi t) \cos(86\pi x)$$

are chosen for the non-conservative equation (1) and the coefficients

$$c_\delta(x, t) = \frac{1}{2} + \frac{1}{8} \cos(8\pi x) + \frac{1}{8} \cos(62\pi x)$$
$$+ \frac{1}{8} \cos(150\pi x) \tag{28}$$
$$A_\varepsilon(x, t) = 10^{-2} + 9 \cdot 10^{-3} \cos(10\pi t) \cos(86\pi x)$$

and

$$c_\delta(x, t) = \frac{3}{4} + \frac{1}{2} \cos(8\pi x) + \frac{1}{4} \cos(62\pi x) + \frac{1}{10} \cos(150\pi x)$$
$$A_\varepsilon(x, t) = 10^{-2} + 9 \cdot 10^{-3} \cos(10\pi t) \cos(86\pi x) \tag{29}$$

conservative equation (2). The latter one is numerically more difficult when it comes to capturing fine-scale variations, i.e., if $c_\delta(x) \sim f(x/\delta)$ then $\frac{d}{dx} c_\delta(x) \sim \delta^{-1} f'(x/\delta)$. So if f and f' are of the same order like in one term of a Fourier expansion of c_δ then one can expect very steep slopes in the solution. The results of the tests are shown in Fig. 7 and the corresponding errors in Fig. 8 and in Table 1.

Test 2a It is an important question how the SLMsR behaves in different regimes of data. For the non-conservative case we show two tests.

The first test demonstrates how the SLMsR behaves when all coefficients of the equation are resolved by the SLMsR, i.e., $h \leq \varepsilon, \delta$ but not by the low resolution FEM that has the same resolution as the SLMsR's coarse resolution, i.e., $H \gg \varepsilon, \delta$. For the coefficients we chose

$$c_\delta(x, t) = \frac{1}{2} + \frac{1}{4} \cos(8\pi x) + \frac{1}{8} \cos(196\pi x) + \frac{1}{16} \cos(210\pi x)$$
$$A_\varepsilon(x, t) = 10^{-3} + 9 \cdot 10^{-4} \cos(174\pi x). \tag{30}$$

The different spatial resolutions were taken as $H = \frac{1}{8}, \frac{1}{32}, \frac{1}{128}, \frac{1}{512}$ while the number of fine cells in each coarse cell was fixed to $n_f = 64$. The reference solution was computed with $h_{\text{ref}} = 2^{-11}$ and the time step for all methods was taken as $\delta t = 10^{-2}$.

Error plots of the results are shown in the first row of Fig. 9. The plots indeed show that the SLMsR captures the reference solution much more accurately than the standard FEM in both L^2 and H^1 even for small H. Only when decreasing H to a resolution that resolves all coefficients the FEM is able to represent the solution correctly and starts converging, i.e., it leaves the pre-asymptotic regime. The reader should keep in mind that in real world applications the standard FEM will be too expensive to compute while the SLMsR allows for massive parallelization. This is the regime that is of practical interest. Increasing the fine

resolution in each coarse cell on the other hand does not improve the accuracy of the SLMsR. It increases the size of the subgrid problems for the basis functions though.

Test 2b The second test is to show that the SLMsR essentially starts behaving like a standard FEM if $H \ll \varepsilon, \delta$. To demonstrate that we took low frequency coefficients

$$c_\delta(x, t) = \frac{1}{2} + \frac{1}{4} \cos(6\pi x) + \frac{1}{8} \cos(10\pi x) + \frac{1}{16} \cos(14\pi x)$$
$$A_\varepsilon(x, t) = 10^{-3} + 9 \cdot 10^{-4} \cos(16\pi x). \tag{31}$$

and refined with the sequence $H = \frac{1}{16}, \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \frac{1}{256}$ while we fixed the number of fine cells in each coarse cell to $n_f = 32$. Note that we start with a coarse resolution regime that almost resolves the data.

The error plots for the SLMsR and the FEM indeed indicate that as H decreases the SLMsR does not dramatically increase its accuracy while the FEM with linear basis functions increases its accuracy according to the standard estimates. This is expected since the standard low-resolution FEM leaves the pre-asymptotic regime with increasing resolution. At the same time that means that the solution gets locally smoother, i.e., it is easier to approximate with piece-wise linear functions. Hence the reconstructed basis does not deviate much from a (linear) standard basis. Note that the SLMsR is still slightly more accurate in the resolved regime but the additional effort that is spent on reconstructing the basis may not be justified any more. This can be observed in the second row of Fig. 9. The relative errors at $t = 1/2$ and $t = 1$ for both test problems (30) and (31) are summarized in Tables 2 and 3 in the "Appendix" together with the estimated order of convergence (EOC) in space.

We recall that the EOC can be computed as follows: Let $e_H : \mathbb{R}^n_+ \to \mathbb{R}_+$ be an (error) function that maps a high dimensional vector to a positive real for each $H > 0$. Then the EOC w.r.t. H can then be computed as

$$\text{EOC} = \frac{\log(e_H) - \log(e_{H/q})}{\log(q)}. \tag{32}$$

Remark 4 Interpreting the EOCs for the SLMsR is delicate in both regimes. In particular the decay rate of the error for the multiscale method will depend on a homogenized solution and on the fine scale. In both the unresolved regime of test (2a) and in the resolved regime of test (2b) one can observe that the error of the SLMsR is lower in L^2 and H^1 than the error of the standard FEM [in L^2 even one order of magnitude in test (2a)]. While the convergence rate of the standard FEM follows a known power law in the asymptotic regime the EOC of the SLMsR follows a different pattern. For a much simpler stationary diffusion problem, for example, one obtains in the case $H \gg \epsilon$ estimates of the form

$$\left\| u - u^H \right\|_{H^1} \leq C_1(H + \varepsilon) \| f \|_{L^2} + C_2 \left(\frac{\varepsilon}{H} \right)^{1/2} \| u_{\text{hom}} \|_{W^{1,\infty}}, \tag{33}$$

see [11, Ch. 6]. Determining the exact asymptotic pattern in our case requires a substantial scale analysis and possibly the knowledge of a homogenized problem (the homogenized limiting regime may depend on an a priori known coupling of the small scale parameters). This will need different tools than in the "simple" case in [11] and is likely to be very difficult in our case. This is out of the scope of this work.

Similar results in terms of accuracy can be obtained in case of a conservative equation. We pass on showing this here.

Test 3 This test shows an example where both diffusion and background velocity are generated randomly. We intend to show an example of the SLMsR behaves when data is involved that
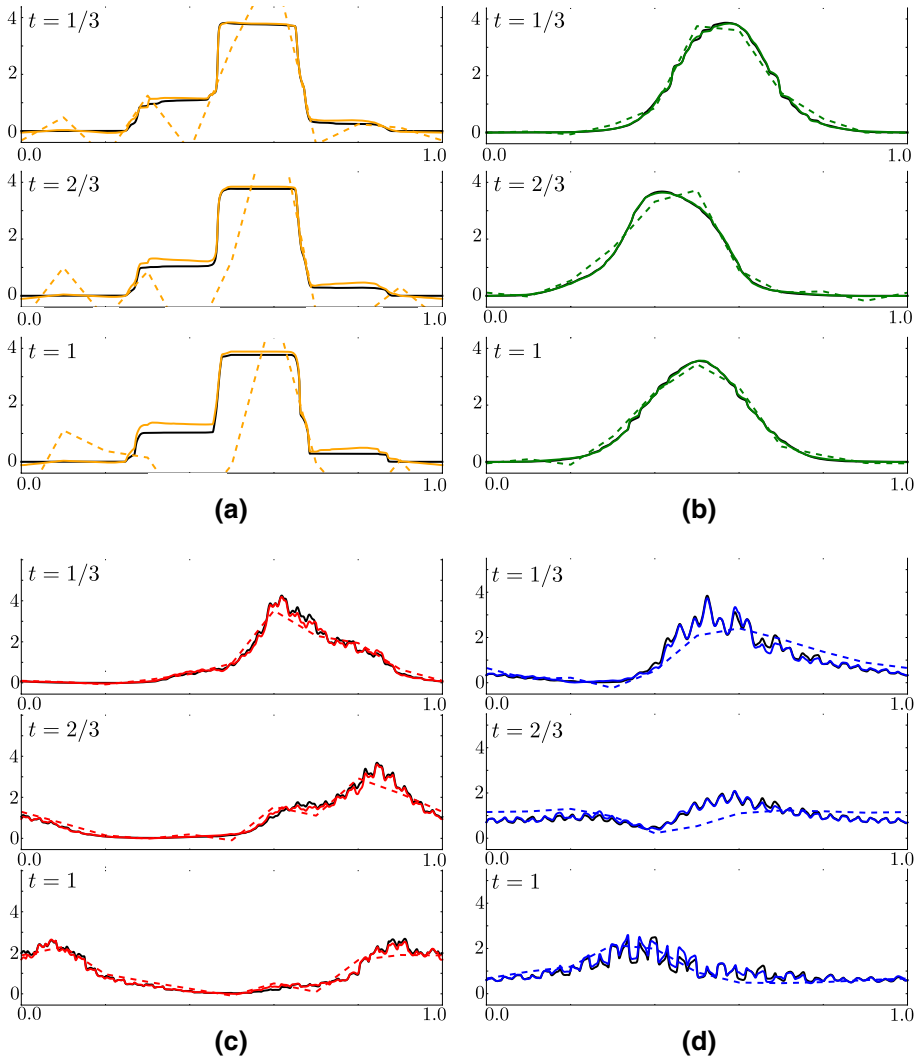
Fig. 7 Snapshots of the solution at $t = 1/3$, $t = 2/3$ and $T = 1$. The colored dashed lines show the solution of the standard FEM, the colored line shows the SLMsR. The reference solution is shown in black. **a** Non-conservative equation (1) Coefficients (26). **b** Non-conservative equation (1) Coefficients (27). **c** Conservative equation (2) Coefficients (28). **d** Conservative equation (2) Coefficients (29) (colour figure online)

does not exhibit a clear scale separation. For this we initially generate fixed mesh based functions with random nodal coefficients. In each mesh cell the functions are interpolated linearly. Note that this is not to simulate a sampled stochastic process. We simply intend not to create any scale or symmetry bias when constructing coefficient functions. The results look appealing and show a clear advantage of the SLMsR, see Fig. 10.
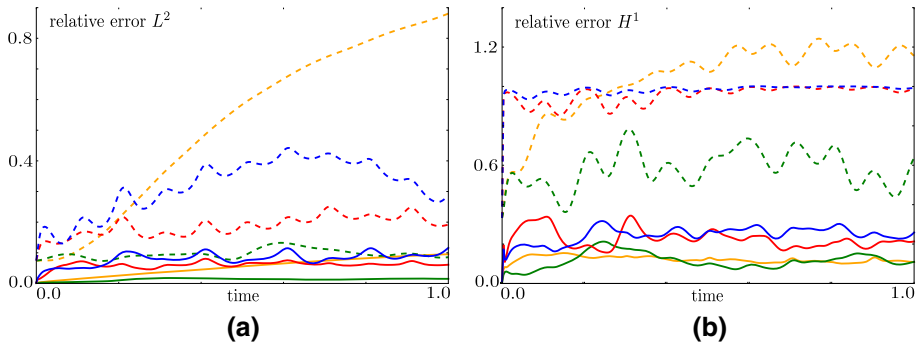
Fig. 8 Relative errors over time in **a** $L^2(\mathbb{T}^1)$ and **b** $H^1(\mathbb{T}^1)$ to the reference solution of the tests shown in Fig. 7. The dashed lines show the error of the standard FEM, the full line shows the error of the SLMsR. Color codes are the same as in Fig. 7
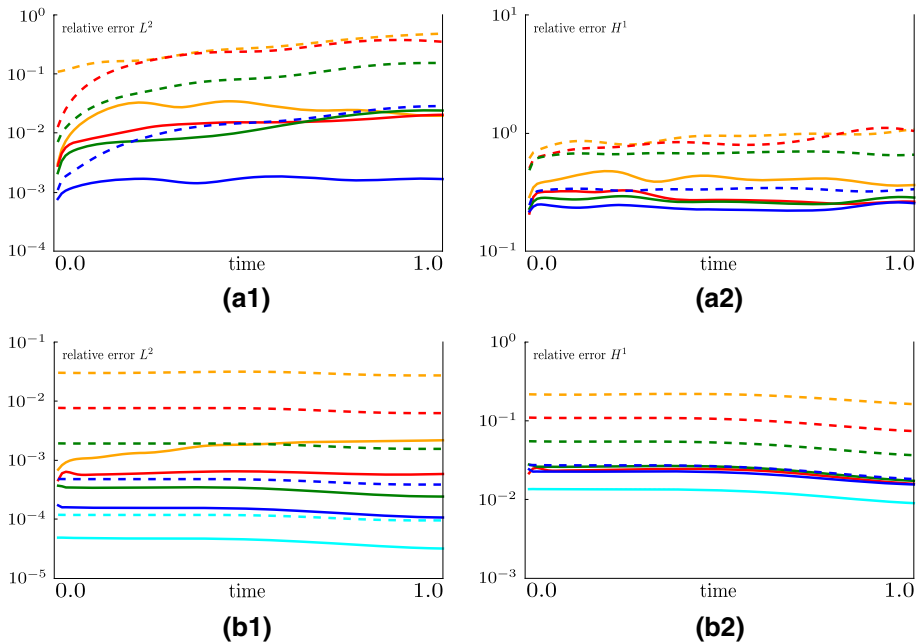
Fig. 9 Relative errors over time for the unresolved regime of test problem (30) in $L^2(\mathbb{T}^1)$ are shown in **(a1)** and errors in $H^1(\mathbb{T}^1)$ are shown in **(a2)**. The coarse resolutions were chosen as $H = 1/8$ (yellow), $H = 1/32$ (red), $H = 1/128$ (green), $H = 1/512$ (blue). The errors of the numerically resolved regime of problem (31) in $L^2(\mathbb{T}^1)$ are shown in **(b1)** and errors in $H^1(\mathbb{T}^1)$ are shown in **(b2)**. The respective reference solutions were computed with $h_{\text{ref}} = 2^{-11}$ and for all experiments we used $\delta t = 10^{-2}$. The dashed lines show the error of the standard FEM, the full line shows the error of the SLMsR. The coarse resolutions were chosen as $H = 1/16$ (yellow), $H = 1/32$ (red), $H = 1/64$ (green), $H = 1/128$ (blue), and $H = 1/256$ (cyan). All errors are shown in a logarithmic scale (colour figure online)

(a)

(b)   **(c)**
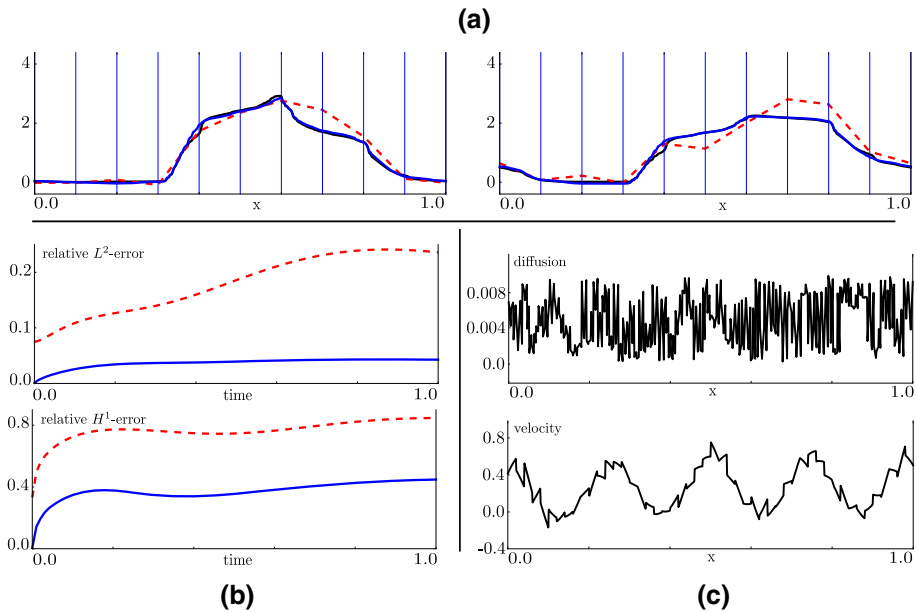
Fig. 10 Comparison of SLMsR and standard FEM for randomly generated (but fixed) coefficient functions. **a** Snapshots of the solution at $t = 1/2$ and $t = 1$. Solid black lines show the reference solution, dashed red lines show the standard solution and solid blue lines show the SLMsR. **b** Relative error plots for H^1 and L^2. **c** Coefficient functions: the diffusion coefficient (upper plot) was generated from a uniform distribution and scale to have minimum 10^{-5} and maximum 10^{-2}. The velocity coefficient (lower plot) we chose to be a smooth function disturbed by Gaussian noise with mean zero and variance 0.1 (colour figure online)

4 Numerical Examples in 2D

This section is to experimentally demonstrate that the SLMsR can handle conservative and non-conservative advection–diffusion equations with dominant advection term in higher dimensions. All tests are being carried out on the torus $\mathbb{T}^2$ (periodic unit square) in the time interval $t \in [0, 1]$ and use the same initial condition. As initial value we chose a normalized super-position of two non-isotropic Gaussians

$$u_0(\mathbf{x}) = \frac{1}{2\sqrt{(2\pi)^2 \det(\mathbf{M})}} \sum_{i=1}^{2} \exp\left\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)^T \mathbf{M}^{-1}(\mathbf{x} - \boldsymbol{\mu}_i)\right\} \tag{34}$$

where

$$\mathbf{M} = \begin{bmatrix} \frac{3}{100} & 0 \\ 0 & \frac{3}{100} \end{bmatrix} \quad \text{and} \quad \boldsymbol{\mu}_i = \left[\frac{i}{3}, \quad \frac{1}{2}\right]^T. \tag{35}$$

All tests of the SLMsR are performed on a coarse unstructured uniform triangular Delaunay mesh with $n_c = 62$ coarse cells, i.e., for our triangulation $H \sim 0.3$ (mean diameter of circumcircle of a cell). We compare the SLMsR to a standard low resolution FEM with the same resolution and to a standard high resolution FEM with approximately $n_f = 63K$ cells. To get a fine mesh on each coarse cell of the SLMsR we created a triangulation such that the sum of all fine cells over all coarse cells is approximately n_f to get a fair comparison of
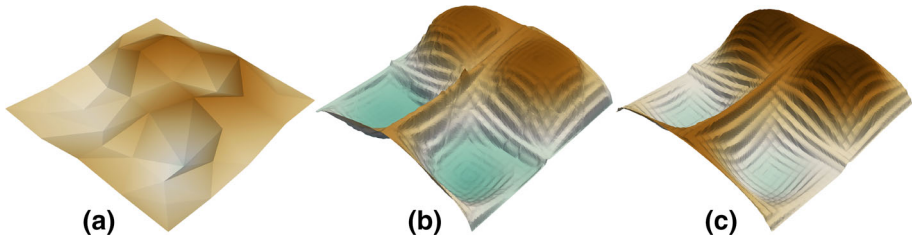
Fig. 11 Snapshots of the solution for *Test 1* at time $T = 1$ for **a** the low resolution standard FEM, **b** the SLMsR and **c** the reference solution

the SLMsR to the low resolution standard FEM with respect to the reference solution that resolves all coefficients involved.

Test 1 Here we test our multiscale reconstruction method with a solenoidal field $\mathbf{c}_\delta$ described by the stream function

$$\psi(\mathbf{x}, t) = \sin(2\pi(x_1 - t)) \sin(2\pi x_2) \tag{36}$$

so that

$$\mathbf{c}_\delta(\mathbf{x}, t) = \nabla^\perp \psi = 2\pi \begin{bmatrix} \sin(2\pi(x_1 - t)) \cos(2\pi x_2) \\ -\cos(2\pi(x_1 - t)) \sin(2\pi x_2) \end{bmatrix}. \tag{37}$$

This background velocity describes four vortices moving in time through the (periodic) domain from left to right and get back to their starting point at $T = 1$. Note that this velocity field involves scales that are resolved by the coarse mesh as well as scales that are not resolved, see Fig. 12. Note that since $\nabla \cdot \mathbf{c}_\delta = 0$ Eqs. (1) and (2) are (analytically) identical and hence we only solve (1).

The diffusion tensor is chosen to be

$$\mathbf{A}_\varepsilon(\mathbf{x}, t) = \frac{1}{100} \mathbb{I}_2 - 0.9999 \begin{bmatrix} \sin(60\pi x_1) & 0 \\ 0 & \sin(60\pi x_2) \end{bmatrix}. \tag{38}$$

Note that in this case advection dominance is a local property and Péclet numbers are ranging from Pe $= 0$ to Pe $\sim 6 \cdot 10^6$. Snapshots of the solutions are shown in Fig. 11. It can be observed that the low resolution FEM does not capture the effective solution well since it diffuses too strongly while the SLMsR reasonably captures the effective behavior of the solution and even the fine scale structure. This can be seen also in the error plots (Fig. 18).

Test 2a For this test we use two divergent background velocities and hence we split it into two parts to distinguish between the diffusive transport problem (1) and the conservation law (2). We start with showing how the SLMsR behaves on equation (1). The velocity field is a modification of (37) and is given by

$$\mathbf{c}_\delta(\mathbf{x}, t) = \begin{bmatrix} \cos(2\pi t) & \sin(2\pi t) \\ -\sin(2\pi t) & \cos(2\pi t) \end{bmatrix} \cdot \begin{bmatrix} 2\pi \sin(2\pi(x_1 - t)) \cos(2\pi x_2) \\ -\cos(2\pi(x_1 - t)) \sin(2\pi x_2) \sin(2\pi x_1) \end{bmatrix}.$$

while the diffusion tensor is also given by (38). Here the standard solution does not converge to any reasonable approximation of the effective solution and no valuable understanding of the dynamics can be drawn from it. The SLMsR shows a surprisingly good approximation of the reference solution, see Fig. 14.

Test 2b Here we solve Eq. (2), i.e., a conservation problem. The divergent velocity field for this test describes regions of fast and slow flow with two separatrices across which there is

Fig. 12 Background velocity for *Test 1* and *Test 3*. Four vortices moving through the domain from left to right and come back to their starting points at $T = 1$. The color indicates the magnitude (red is high, blue is low) (colour figure online)
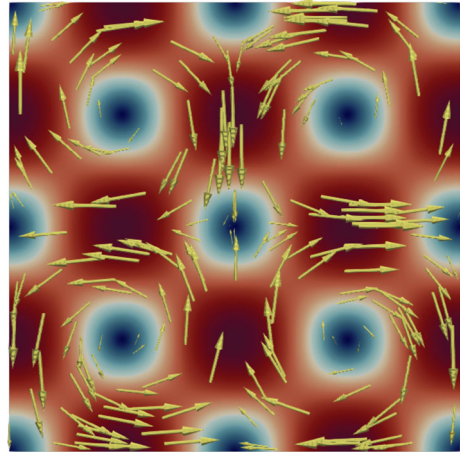
no flow moving in time from left to right once through the domain during the time interval. The field is given by

$$\mathbf{c}_\delta(\mathbf{x}, t) = \frac{2\pi}{5} \begin{bmatrix} \sin(2\pi(x_1 - t))^2 \cos(\pi(x_2 - 0.5))^2 \\ 2\sin(2\pi(x_1 - t)) \cos(2\pi(x_1 - t)) \cos(\pi(x_2 - 0.5))^2. \end{bmatrix} \tag{40}$$

Both velocity fields for *Test 2a* and *Test 2b* are shown in Fig. 13. The diffusion tensor is again given by (38). The difference to the previous test is that we have quite a large additional reaction term to handle. This will be taken care of in the basis representation as described in Sects. 2.5 and 2.6 as well as in the global weak form, see Eq. (7).

Note that the difficulty we focus on here is not the actual conservation (although this is an important issue) but the fact that we do have another lower order (reaction) term in the equation with a multiscale character. Fortunately this term is very local and is handled nicely by the SLMsR since it scales with its magnitude. The results are shown in Fig. 15 where we also show how the quality of the SLMsR solution is negatively affected when the reconstructed basis boundary conditions are not propagated. As the reader can see the SLMsR without edge evolution shows a slight grid imprinting of the coarse mesh. This effect can be more pronounced depending on the strength of the reactive term since it is due to keeping the boundary conditions fixed in time in the local basis evolution step. This in turn increases the error in the evolution of the asymptotic structure near coarse grid boundaries due to the additional reaction term.

We pass on showing a comparison of all snapshots to the low resolution FEM here (which does not behave well) for the sake of brevity. Error plots including the low resolution FEM are again summarized in Fig. 18.

Test 3 This test has a similar intention as *Test 3* in Sect. 3. We intend to show a reasonable behavior of the SLMsR if the coefficients do not show a clear scale separation on Eq. (1). For this we again used (37) as background velocity and generated a diagonal diffusion tensor that is constant in each cell of a mesh generated to represent solely the diffusion. The cell based constants are random and are fixed at the beginning of the simulation. The diffusion tensor was then scaled to contain values in the range of 10^{-5} to 10^{-1}, see Fig. 16. Snapshots of the solution can be found in Fig. 17 and error plots in Fig. 18.

Remark 5 According to the choice of boundary conditions of the basis functions during the reconstruction the basis are linearly independent. Theoretically it could nonetheless happen

Fig. 13 Background velocities for *Test 2a* and *Test 2b* given by (39) and (40), respectively. Both fields are divergent. The color indicates the magnitude (red is high, blue is low) (colour figure online)
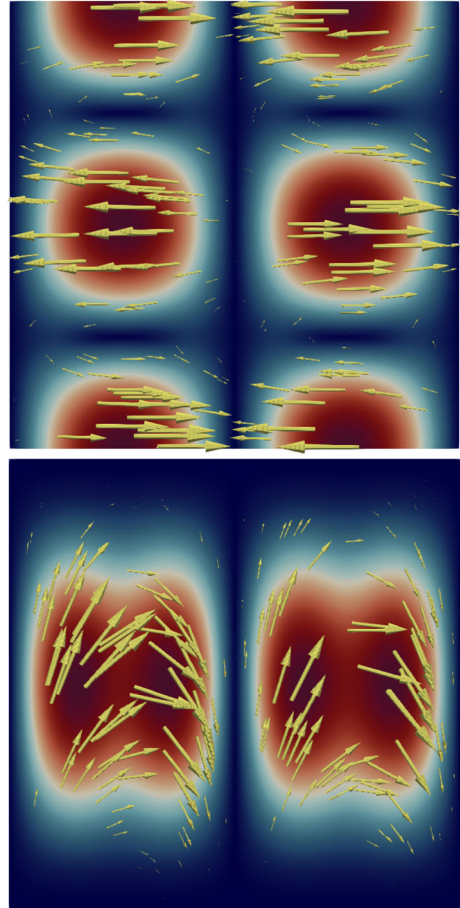

that the basis functions become "nearly dependent" which would result in "nearly" ill-posed (ill-conditioned) discrete problems. This case is for example given if the multiscale corrector (i.e., the difference between a standard basis and the computed multiscale basis) becomes too dominant over the standard basis. We did not observe this in our experiments and it can generally be avoided through putting more weight on the regularizing term in the inverse problem for the basis reconstruction, e.g. to penalize a deviation from the standard basis that is too strong as done in Eq. (15). After the application of boundary conditions the condition numbers of the discrete systems to solve that we found were around 3–4 for the coarse standard FEM and about 4–5 times higher compared to the coarse standard problem for our semi-Lagrangian multiscale method which is below the condition number of the high-resolution reference FEM.
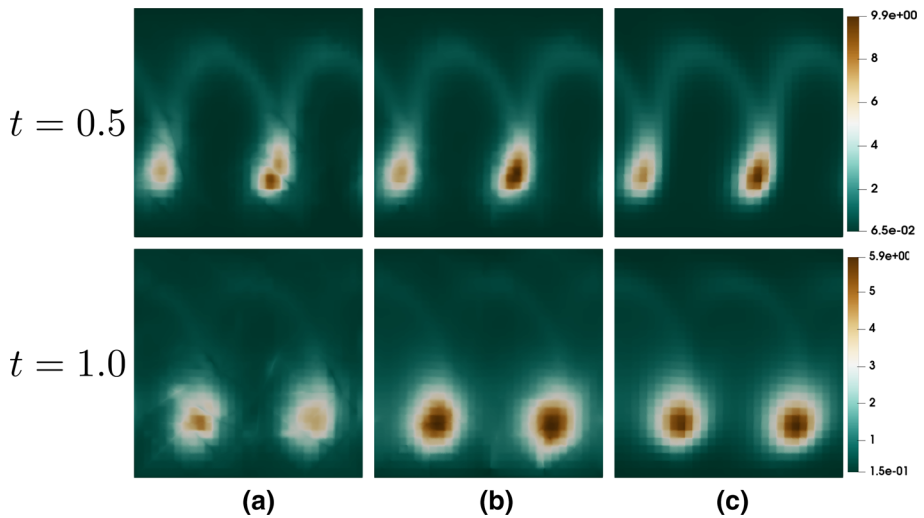
5 Summary and Discussion

In this work we introduced a multiscale method for advection–diffusion equations that are dominated by the advective term. Such methods are of importance, for example, in reservoir
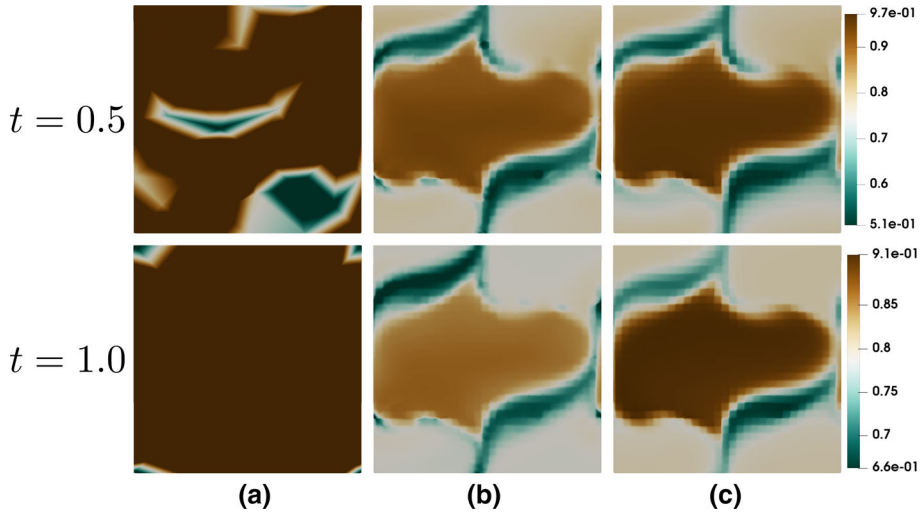
Fig. 14 Comparison of **a** low resolution FEM and **b** SLMsR to **c** the reference solution of *Test 2a* at two time instances


Fig. 15 Comparison of **a** SLMsR without edge evolution and **b** SLMsR with edge evolution to **c** the reference solution in case of conservative problem *Test 2b* shown at two time instances. The SLMsR without edge evolution shows stronger grid imprinting of the coarse mesh

modeling and tracer transport in earth system models. The main obstacles in these applications are, first, the advection-dominance and, secondly, the multiscale character of the background velocity and the diffusion tensor.

The latter makes it impossible to simulate with standard methods due to computational constraints while simulating using standard methods with lower resolution that does not resolve variations in the coefficients leads to wrong effective solutions.

Our idea to cope with these difficulties is inspired by ideas for semi-Lagrangian methods, ideas based on "convected fluid microstructure" as described in [19], inverse problems and
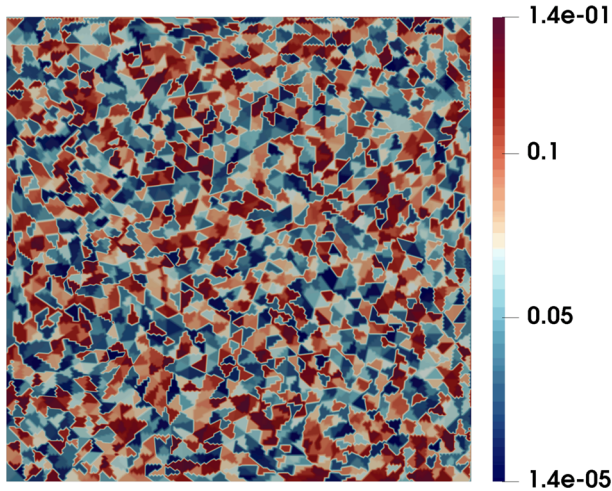
Fig. 16 Magnitude of randomly generated (but fixed) diffusion tensor for *Test 3*

(a)   **(b)**   **(c)**
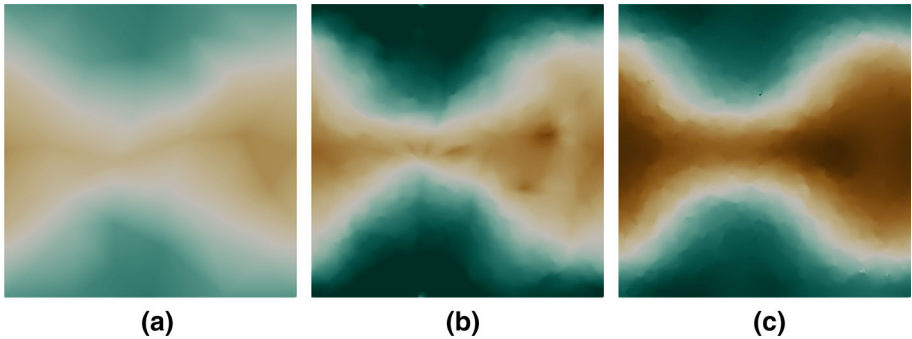
Fig. 17 Snapshot of solutions of *Test 3* at $T = 1$. We show **a** the low resolution FEM, **b** the SLMsR and **c** the reference solution
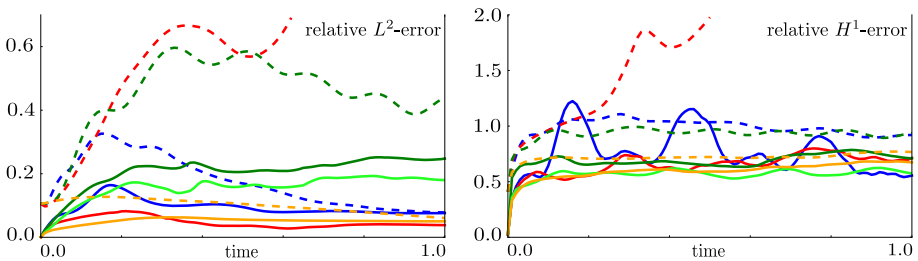
Fig. 18 Relative errors in $L^2(\mathbb{T})$ and $H^1(\mathbb{T})$ over time for all test cases of Sect. 4. The SLMsR (solid lines) in general shows lower errors than the low resolution standard FEM (dashed lines). The color codes are as follows: *Test 1* is shown in blue, *Test 2a* is shown in red, *Test 2b* is shown in green where the dark green solid line shows the error without edge evolution and the light green solid line shows the error with edge evolution according to (23). *Test 3* is shown in yellow (colour figure online)

multiscale finite elements [11]. At each time step we reconstruct fine scale information from the solution at the previous time step. This fine scale information enters the local representation of the solution in each coarse cell, i.e., it is added as a corrector to the local basis such that the basis representation is optimal in some sense. Due to advection-dominance the optimal basis needs to be reconstructed in its departure area, i.e., we locally trace back information like in a semi-Lagrangian method for the reconstruction. The reconstruction is performed by solving an inverse problem with a suitable regularizer. It constructs a basis that does not constitute a partition of unity (PoU) and is tailored for the actual problem at hand. The idea of adding prior knowledge about the solution to a local representation in PoU methods, however, is similar, see for example [15,29]. After reconstructing the basis at the previous time step it is evolved with suitable boundary conditions to the time level, where it is needed; i.e., we evolve the local representation of the solution rather that the solution itself. Note that the global framework of the SLMsR is completely Eulerian while only the local reconstruction step in each coarse cell is semi-Lagrangian.

The SLMsR also shares shallow similarities with stabilized methods such as streamline upwind Petrov–Galerkin (SUPG) methods in which "advected" basis functions are used as correctors for test functions to add a small diffusion in the direction of the flow. This is however not the same since our basis also corrects for the fine scale asymptotic structure induced by diffusion and other low order terms.

Numerical experiments that we carried out in one and two dimensions for conservative and non-conservative advection diffusion equations show a clear advantage in terms of accuracy for the SLMsR at low (coarse) resolutions. We formulated the method globally as a conformal FEM to show how the idea works in a quite restrictive setting. Other frameworks are possible. A non-conformal FEM, for example, would put less restrictions on the basis reconstruction at the price of having to compute a boundary integral in the global assembly for each cell.

One of the main features of the SLMsR is its scalability: Although it sounds expensive to trace back each coarse cell, then solve an inverse problem and then solve a PDE at each time step (the so-called offline phase) we would like to point out that these local problems are independent and usually small and therefore the offline phase is embarrassingly parallel. The global time step (online phase) also consists of a small problem and matrix assembly procedures can be made very efficient by using algebraic expressions, see [11,21].

Also, note that although the global time step maybe needs to be decreased to prevent coarse cells from flipping when tracing the cell back but this problem appears in all semi-Lagrangian methods. Small scale variations in the background flow of order δ though are not too dangerous it is rather the large scale dominant features that are problematic. Yet, as is known from standard semi-Lagrangian methods large Courant numbers can be achieved in the reconstruction step (how large depends on the coarse mesh size and on the data). The fine scale time steps may need to be smaller since scales of order ϵ, δ must be resolved locally. But, again, this is only a local constraint. This does not pose a severe bottleneck either due to the option to parallelize and since local problems are independent.

We would like to further emphasize the flexibility of the SLMsR. Here we presented an implicit version but explicit time stepping is possible. Also, the method can be transferred to higher dimensions as well as it can be extended to deal with advection–diffusion-reaction problems. Furthermore, the use of inverse problems in the local steps to adjust the basis makes it generally possible to incorporate knowledge coming from measurement data. For this a thorough understanding of the data is necessary (as for any other assimilation method). We would like to explore that opportunity in the future.

We believe that this method has potential to improve solutions to real-life multi-scale problems since it is flexible and rises many possibilities to transfer ideas to different types of

equations and methods. Some of the future issues that we aim at include its transfer to other Galerkin methods such as discontinuous Galerkin methods, attacking nonlinear and vector valued problems and at the use of actual data for subscale assimilation.

Declarations

Code Availability A prototypical implementation of the method in Julia v1.1 in 2D is made a available on Github.

A Tables

See Tables 1, 2 and 3

Table 1 Relative errors of standard FEM and SLMsR for the non-conservative test problem (27) (upper table) and for the conservative problem (28) (lower table)

H	t	L^2_{rel}		H^1_{rel}	
		FEM	SLMsR	FEM	SLMsR
$H = 1/8$	1/3	0.109517	0.0170849	0.738011	0.223987
	2/3	0.110731	0.0121468	0.521164	0.104679
	1	0.100034	0.0183614	0.652919	0.191475
$H = 1/32$	1/3	0.0429804	0.0141442	0.681634	0.305783
	2/3	0.0336967	0.00553025	0.27492	0.139942
	1	0.0264641	0.00911659	0.559342	0.144821
$H = 1/128$	1/3	0.0148088	0.00477275	0.431396	0.17528
	2/3	0.0163083	0.00225746	0.173665	0.0740701
	1	0.0113428	0.00258289	0.426786	0.160761
$H = 1/512$	1/3	0.00088987	0.00097043	0.126697	0.0947134
	2/3	0.000589497	0.000504804	0.0446262	0.0435349
	1	0.000578547	0.000491731	0.159977	0.173261
$H = 1/8$	1/3	0.214453	0.0702007	0.921891	0.321898
	2/3	0.171949	0.0662604	0.940582	0.261431
	1	0.188325	0.0743565	0.975117	0.336754
$H = 1/32$	1/3	0.139444	0.0283479	0.975416	0.256681
	2/3	0.108827	0.0294402	0.974576	0.250716
	1	0.140284	0.0398691	1.02447	0.252878
$H = 1/128$	1/3	0.0373321	0.00634348	0.62717	0.109299
	2/3	0.0322432	0.0119677	0.628683	0.086647
	1	0.0446856	0.0233804	0.78208	0.183882
$H = 1/512$	1/3	0.00274732	0.00200674	0.182691	0.0595046
	2/3	0.00238125	0.00412889	0.181782	0.0554508
	1	0.0045627	0.00622009	0.290947	0.140458

The errors are shown at three different time steps

Table 2 Upper table: relative errors at time $t = 1/2$ and $t = 1$ for the unresolved regime of test problem (30) in $L^2(\mathbb{T}^1)$ and $H^1(\mathbb{T}^1)$ at different coarse resolutions

H	t	L^2_{rel}		H^1_{rel}	
		FEM	SLMsR	FEM	SLMsR
1/8	1/2	$2.71124 \cdot 10^{-1}$	$3.29375 \cdot 10^{-2}$	$9.47533 \cdot 10^{-1}$	$4.26613 \cdot 10^{-1}$
	1	$4.80502 \cdot 10^{-1}$	$1.9577 \cdot 10^{-2}$	$1.07721 \cdot 10^{0}$	$3.62467 \cdot 10^{-1}$
1/32	1/2	$2.38448 \cdot 10^{-1}$	$1.51868 \cdot 10^{-2}$	$8.06106 \cdot 10^{-1}$	$2.71302 \cdot 10^{-1}$
	1	$3.4961 \cdot 10^{-1}$	$2.03215 \cdot 10^{-2}$	$1.038 \cdot 10^{0}$	$2.62539 \cdot 10^{-1}$
1/128	1/2	$8.23856 \cdot 10^{-2}$	$1.07117 \cdot 10^{-2}$	$6.75349 \cdot 10^{-1}$	$2.62084 \cdot 10^{-1}$
	1	$1.52502 \cdot 10^{-1}$	$2.39788 \cdot 10^{-2}$	$6.54747 \cdot 10^{-1}$	$2.84093 \cdot 10^{-1}$
1/512	1/2	$1.48402 \cdot 10^{-2}$	$1.74951 \cdot 10^{-3}$	$3.35263 \cdot 10^{-1}$	$2.24681 \cdot 10^{-1}$
	1	$2.85318 \cdot 10^{-2}$	$1.65701 \cdot 10^{-3}$	$3.33875 \cdot 10^{-1}$	$2.54352 \cdot 10^{-1}$

$H \to H/4$	$h \to h/4$	EOC L^2_{rel}		EOC H^1_{rel}	
		FEM	SLMsR	FEM	SLMsR
$2^{-3} \to 2^{-5}$	$2^{-9} \to 2^{-11}$	0.22939	-0.02692	0.02674	0.23265
$2^{-5} \to 2^{-7}$	$2^{-11} \to 2^{-13}$	0.59845	-0.11937	0.33239	-0.05691
$2^{-7} \to 2^{-9}$	$2^{-13} \to 2^{-15}$	1.20909	1.92755	0.48581	0.07977

The fine resolution was fixed to $n_f = 64$ cells per coarse cell. Lower table: estimated order of convergence at $t = 1$

Table 3 Upper table: relative errors at time $t = 1/2$ and $t = 1$ for the resolved regime of test problem (31) in $L^2(\mathbb{T}^1)$ and $H^1(\mathbb{T}^1)$ at different coarse resolutions

H	t	L^2_{rel}		H^1_{rel}	
		FEM	SLMsR	FEM	SLMsR
1/16	1/2	$3.14117 \cdot 10^{-2}$	$1.83944 \cdot 10^{-3}$	$2.16986 \cdot 10^{-1}$	$2.17566 \cdot 10^{-2}$
	1	$2.71271 \cdot 10^{-2}$	$2.16896 \cdot 10^{-3}$	$1.62237 \cdot 10^{-1}$	$1.58667 \cdot 10^{-2}$
1/32	1/2	$7.61179 \cdot 10^{-3}$	$6.45978 \cdot 10^{-4}$	$1.064 \cdot 10^{-1}$	$2.4219 \cdot 10^{-2}$
	1	$6.31955 \cdot 10^{-3}$	$5.84086 \cdot 10^{-4}$	$7.45369 \cdot 10^{-2}$	$1.58233 \cdot 10^{-2}$
1/64	1/2	$1.90004 \cdot 10^{-3}$	$3.36293 \cdot 10^{-4}$	$5.32668 \cdot 10^{-2}$	$2.61811 \cdot 10^{-2}$
	1	$1.56285 \cdot 10^{-3}$	$2.41687 \cdot 10^{-4}$	$3.68329 \cdot 10^{-2}$	$1.72855 \cdot 10^{-2}$
1/128	1/2	$4.73831 \cdot 10^{-4}$	$1.51248 \cdot 10^{-4}$	$2.66104 \cdot 10^{-2}$	$2.2229 \cdot 10^{-2}$
	1	$3.88823 \cdot 10^{-4}$	$1.07755 \cdot 10^{-4}$	$1.83422 \cdot 10^{-2}$	$1.55475 \cdot 10^{-2}$
1/256	1/2	$1.17218 \cdot 10^{-4}$	$4.54382 \cdot 10^{-5}$	$1.32288 \cdot 10^{-2}$	$1.31489 \cdot 10^{-2}$
	1	$9.60583 \cdot 10^{-5}$	$3.19474 \cdot 10^{-5}$	$9.11126 \cdot 10^{-3}$	$9.05413 \cdot 10^{-3}$

$H \to H/2$	$h \to h/2$	EOC L^2_{rel}		EOC H^1_{rel}	
		FEM	SLMsR	FEM	SLMsR
$2^{-4} \to 2^{-5}$	$2^{-9} \to 2^{-10}$	2.10184	1.89275	1.12208	0.003951
$2^{-5} \to 2^{-6}$	$2^{-10} \to 2^{-11}$	2.01564	1.27304	1.01696	-0.12751
$2^{-6} \to 2^{-7}$	$2^{-11} \to 2^{-12}$	2.00699	1.16538	1.00583	0.152887
$2^{-7} \to 2^{-8}$	$2^{-12} \to 2^{-13}$	2.01713	1.75399	1.00944	0.780031

The fine resolution was fixed to $n_f = 32$ cells per coarse cell. lower table: estimated order of convergence at $t = 1$

References

1. Abdulle, A.: The finite element heterogeneous multiscale method: a computational strategy for multiscale PDES. GAKUTO Int. Ser. Math. Sci. Appl. **31**(EPFL–ARTICLE–182121), 135–184 (2009)
2. Abdulle, A., Weinan, E., Engquist, B., Vanden-Eijnden, E.: The heterogeneous multiscale method. Acta Numer. **21**, 1–87 (2012)
3. Abdulle, A., Engquist, B.: Finite element heterogeneous multiscale methods with near optimal computational complexity. Multiscale Model. Simul. **6**(4), 1059–1084 (2007)
4. Ahmed, N., Rebollo, T.C., John, V., Rubino, S.: A review of variational multiscale methods for the simulation of turbulent incompressible flows. Arch. Comput. Methods Eng. **24**(1), 115–164 (2017)
5. Allaire, G.: Homogenization and two-scale convergence. SIAM J. Math. Anal. **23**(6), 1482–1518 (1992)
6. Babuška, I., Caloz, G., Osborn, J.E.: Special finite element methods for a class of second order elliptic problems with rough coefficients. SIAM J. Numer. Anal. **31**(4), 945–981 (1994)
7. Behrens, J.: Adaptive Atmospheric Modeling-Key Techniques in Grid Generation, Data Structures, and Numerical Operations with Applications, Lecture Notes in Computational Science and Engineering, vol. 54. Springer, Berlin (2006)
8. Bensoussan, A., Lions, J.L., Papanicolaou, G.: Asymptotic Analysis for Periodic Structures, vol. 374. American Mathematical Society, Providence (2011)
9. Celia, M.A., Russell, T.F., Herrera, I., Ewing, R.E.: An Eulerian–Lagrangian localized adjoint method for the advection–diffusion equation. Adv. Water Resour. **13**(4), 187–206 (1990)
10. Cheng, A., Wang, K., Wang, H.: A preliminary study on multiscale ELLAM schemes for transient advection–diffusion equations. Numer. Methods Partial Differ. Equ. **26**(6), 1405–1419 (2010)
11. Efendiev, Y., Hou, T.Y.: Multiscale Finite Element Methods: Theory and Applications, vol. 4. Springer, Berlin (2009)
12. Efendiev, Y.R., Hou, T.Y., Wu, X.H.: Convergence of a nonconforming multiscale finite element method. SIAM J. Numer. Anal. **37**(3), 888–910 (2000)
13. Evans, L.C.: Partial Differential Equations. American Mathematical Society, Providence (2010)
14. Graham, I.G., Hou, T.Y., Lakkis, O., Scheichl, R.: Numerical Analysis of Multiscale Problems, vol. 83. Springer, Berlin (2012)
15. Henning, P., Morgenstern, P., Peterseim, D.: Multiscale partition of unity. In: Meshfree Methods for Partial Differential Equations VII, pp. 185–204. Springer, Berlin (2015)
16. Henning, P., Ohlberger, M.: The heterogeneous multiscale finite element method for advection–diffusion problems with rapidly oscillating coefficients and large expected drift. Netw. Heterog. Med. **5**(4), 711–744 (2010)
17. Henning, P., Peterseim, D.: Oversampling for the multiscale finite element method. Multiscale Model. Simul. **11**(4), 1149–1175 (2013)
18. Herrera, I., Ewing, R.E., Celia, M.A., Russell, T.F.: Eulerian–Lagrangian localized adjoint method: the theoretical framework. Numer. Methods Partial Differ. Equ. **9**(4), 431–457 (1993)
19. Holm, D.D., Tronci, C.: Multiscale turbulence models based on convected fluid microstructure. J. Math. Phys. **53**(11), 115614 (2012)
20. Hou, T., Wu, X.H., Cai, Z.: Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. Math. Comput. Am. Math. Soc. **68**(227), 913–943 (1999)
21. Hou, T.Y., Wu, X.H.: A multiscale finite element method for elliptic problems in composite materials and porous media. J. Comput. Phys. **134**(1), 169–189 (1997)
22. Hughes, T.J.: Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. Comput. Methods Appl. Mech. Eng. **127**(1–4), 387–401 (1995)
23. Hughes, T.J., Feijóo, G.R., Mazzei, L., Quincy, J.B.: The variational multiscale method—a paradigm for computational mechanics. Comput. Methods Appl. Mech. Eng. **166**(1–2), 3–24 (1998)
24. Huysmans, M., Dassargues, A.: Review of the use of Péclet numbers to determine the relative importance of advection and diffusion in low permeability environments. Hydrogeol. J. **13**(5–6), 895–904 (2005)
25. Jikov, V.V., Kozlov, S.M., Oleinik, O.A.: Homogenization of Differential Operators and Integral Functionals. Springer, Berlin (2012)
26. Le Bris, C., Legoll, F., Madiot, F.: A numerical comparison of some multiscale finite element approaches for advection-dominated problems in heterogeneous media. ESAIM Math. Model. Numer. Anal. **51**(3), 851–888 (2017)
27. Lee, Y., Engquist, B.: Multiscale numerical methods for passive advection–diffusion in incompressible turbulent flow fields. J. Comput. Phys. **317**(317), 33–46 (2016)
28. Li, G., Peterseim, D., Schedensack, M.: Error analysis of a variational multiscale stabilization for convection-dominated diffusion equations in 2d. J. Numer. Anal. **38**(3), 1229–1253 (2018)

29. Melenk, J.M., Babuška, I.: The partition of unity finite element method: basic theory and applications. Comput. Methods Appl. Mech. Eng. **139**(1–4), 289–314 (1996)
30. Notz, D., Bitz, C.M.: Sea ice in earth system models, chap. 12. In: Thomas, D.N. (ed.) Sea Ice, pp. 304–325. Wiley, New York (2017)
31. Rasthofer, U., Gravemeier, V.: Recent developments in variational multiscale methods for large-Eddy simulation of turbulent flow. Arch. Comput. Methods Eng. **25**(3), 647–690 (2018)
32. Russell, T.F., Celia, M.A.: An overview of research on Eulerian–Lagrangian localized adjoint methods (ELLAM). Adv. Water Resour. **25**(8), 1215–1231 (2002)
33. Salim, M.H., Schlünzen, H., Grawe, D., Boettcher, M., Gierisch, A.M., Fock, B.: The microscale obstacle-resolving meteorological model MITRAS v2.0: model theory. Geosci. Model Dev. **11**(8), 3427–3445 (2018)
34. Schlünzen, K., Hinneburg, D., Knoth, O., Lambrecht, M., Leitl, B., Lopez, S., Lüpkes, C., Panskus, H., Renner, E., Schatzmann, M., et al.: Flow and transport in the obstacle layer: first results of the micro-scale model MITRAS. J. Atmos. Chem. **44**(2), 113–130 (2003)
35. Simon, K., Behrens, J.: A semi-Lagrangian multiscale framework for advection-dominant problems. In: International Conference on Computational Science, pp. 393–409. Springer, Berlin (2019)
36. Simon, K., Behrens, J.: Multiscale finite elements for transient advection–diffusion equations through advection-induced coordinates. Multiscale Model. Simul. **18**(2), 543–571 (2020)
37. Wang, H., Ding, Y., Wang, K., Ewing, R.E., Efendiev, Y.R.: A multiscale Eulerian–Lagrangian localized adjoint method for transient advection–diffusion equations with oscillatory coefficients. Comput. Vis. Sci. **12**(2), 63–70 (2009)
38. Weinan, E.: Principles of Multiscale Modeling. Cambridge University Press, Cambridge (2011)
39. Weinan, E., Engquist, B.: The heterogenous multiscale methods. Commun. Math. Sci. **1**(1), 87–132 (2003)
40. Weinan, E., Engquist, B., Huang, Z.: Heterogeneous multiscale method: a general methodology for multiscale modeling. Phys. Rev. B **67**(9), 092101 (2003)