arXiv:1802.07684v1 [math.NA] 21 Feb 2018

Multiscale finite elements through advection-induced coordinates for transient advection-diffusion equations

Konrad Simon* and Jörn Behrens*

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

Abstract. Long simulation times in climate sciences typically require coarse grids due to computational constraints. Nonetheless, unresolved subscale information significantly influences the prognostic variables and can not be neglected for reliable long term simulations. This is typically done via parametrizations but their coupling to the coarse grid variables often involves simple heuristics. We explore a novel up-scaling approach inspired by multi-scale finite element methods. These methods are well established in porous media applications, where mostly stationary or quasi stationary situations prevail. In advection-dominated problems arising in climate simulations the approach needs to be adjusted. We do so by performing coordinate transforms that make the effect of transport milder in the vicinity of coarse element boundaries. The idea of our method is quite general and we demonstrate it as a proof-of-concept on a one-dimensional passive advection-diffusion equation with oscillatory background velocity and diffusion.

1 Introduction

Motivation and Overview. Geophysical processes in our atmosphere and in the oceans involve many different spatial and temporal scales. Reliable climate simulations hence need to take into account the interaction of many scales since these are coupled and influence each other. Resolving all relevant scales and their interactions poses immense computational requirements. Even on modern high-performance computers computational constraints force us to make compromises, i.e., to use a limited grid resolution which then often neglects certain scale interactions or to model interactions in a heuristic manner.

For example long-term climate simulations as done in the simulation of paleo climate use grid resolutions of approximately 200km and more. This of course ignores fine scale processes that can not be resolved by this resolution although they significantly influence prognostic variables on the coarse grid. Examples include (but are not limited to) moving ice-shields, land-sea boundaries, flow over rough orography, cloud physics and precipitation. None of these processes is resolved by the grid and current climate simulations cope with this by using so-called parametrizations. These can be seen as replacements or simplifications of subgrid processes or processes that are too complex to be taken into account on the prognostic (coarse) scale. Coupling to the coarse grid is often done heuristically and can even cause convergence issues when refining the computational grid.

It is our aim to improve the process of transferring information from the subgrid scale to the coarse grid in a mathematically consistent way. Numerical multiscale modelling offers promising mathematical frameworks to achieve this goal. Such methods are already quite established in other communities such as the porous media community but rarely made their way to climate simulations. Most mentionable here are methods based on homogenization, the heterogeneous

multiscale method (HMM), multiscale methods based on the Eulerian-Lagrangian localized adjoint method (ELLAM and MsELLAM), variational multiscale methods and multiscale finite element methods (MsFEM).

Homogenization theory [7, 22, 23] was originally developed as an analytical tool to study the behavior of solutions to equations with coefficients that rapidly oscillate on fast scales $\mathcal{O}(\varepsilon)$ in the limit $\varepsilon \to 0$. An equation with effective coefficients is developed that correctly describes the influence of fast scales as the scale separation grows. The process of incorporating fine scale information into coarser scales is usually referred to as upscaling. This does, however, not resolve fine scales. Also analytical results for effective equations are very difficult to obtain and available only in a limited number of cases.

The HMM introduces a framework for a large range of multiscale problems [12, 11, 1] rather than a concrete method. In contrast to many other techniques it constitutes a top-down approach based on blending very different natures of macro and micro models. One simulates a usually incomplete macro-scale model by choosing an appropriate method (e.g., a finite element method when dealing with a variational problem). Missing data is incorporated wherever it is needed by performing constrained micro-scale simulations. The micro-scale data then have to be incorporated into a properly chosen macro solver. The HMM offers a very general approach but needs a careful choice of the macro and micro models and their interaction, i.e., the compression and reconstruction of information.

MsELLAM [10, 25] is based on ELLAM [9, 16, 24] which uses a space time finite element method (FEM). Basis functions that satisfy the adjoint equation are constructed locally in time, which leads to the elimination of non-boundary terms in the weak form. Back-tracking characteristics makes it possible to deal with advection dominated problems but leads to discontinuities in time.

Variational multiscale methods first emerged in [8, 19, 20]. The idea here is to decompose the solution space into a coarse and a fine scale part and to then separate the usual variational form into a part that is tested with coarse scale test functions and a part that is tested with fine scale test functions. This approach, in contrast to the HMM and to homogenization, resolves fine scale features of the numerical solution. Mixed variational methods have later been introduced in [2, 3], see also [15]

The MsFEM is closely related to the variational multiscale method and has originally been introduced in [5, 4] and later been substantially refined in [17, 18]. It relies on the use of non-polynomial basis functions that reflect the fine scale behavior of the solution. Basis functions are usually constructed by requiring them to satisfy the equation to leading order. Many variations have been introduced since then, see [13] for a survey. The method is very attractive since it allows for massive parallelization and is therefore very scalable.

All the above mentioned methods work well for elliptic problems and stand in contrast to classical adaptive mesh refinement techniques (AMR) which essentially constitute a down-scaling approach rather than an upscaling framework by locally resolving "interesting" regions of the grid [21, 6]. They are proven to work well also for hyperbolic problems. Our list of methods is by far not complete. We therefore refer in particular to more comprehensive texts on different parts of multiscale modeling [6, 15, 13, 12, 7] and to references therein.

Contribution. In this work we introduce a generalization of the classical MsFEM introduced in [17] to suggest an idea for dealing with problems that arise in long-term simulations of climate. The partial differential equations used for these simulations are mostly time-dependent and are dominated by advection. We therefore demonstrate our idea in a simple setting on a one-dimensional advection diffusion equation on a periodic domain that is dominated by the advective term with fine scale diffusion. In such a scenario the classical MsFEM which is designed for elliptic problems fails since it relies on a decomposition of the computational domain into a number of coarse blocks. These coarse blocks are the support of the modified basis functions but blocks are essentially decoupled from each other. Hence information travelling through the entire domain is blocked at the coarse block boundaries. This means that boundary conditions of the basis functions need to be chosen with care. Also the basis functions need to be transient themselves.

Choosing appropriate boundary conditions for the local problems (here for the basis functions) is a general problem in multiscale computations. Posing Dirichlet boundary conditions as in the classical MsFEM on the coarse block boundaries basis functions can develop boundary layers that do not represent any features that are apparent in the actual flow. We intend to circumvent this problem not by a different choice of boundary conditions but by posing Dirichlet conditions on curves that account for the advective part of the equation.

Our key idea is based on the fact that for simple background velocity fields one can transform an advection-diffusion equation into a pure diffusion equation by following the characteristics of the advective part, i.e., we move to a Lagrangian setting where we only "see" diffusion across the characteristics. Such a transform can be done without any danger of crossing characteristics if the background velocity does not depend on space (it may depend on time though). If the background velocity depends on space we either average in space and follow the mean flow characteristics or we follow the characteristics at the coarse grid boundaries and interpolate them inside the blocks. Each procedure introduces a transform of the equation that reduces the influence of the advective term.

For a well-behaved background velocity, for example taken from a coarse grid in a climate simulation, characteristics do not come too close. Inside each coarse block interpolation of the characteristics emerging at the coarse grid boundary ensures that (interpolated) characteristics do not cross or come too close. This amounts to a setting that is not fully Lagrangian but still makes the advective term milder. The transformed equation then is very close to a pure parabolic setting inside a coarse block and purely parabolic in the vicinity of the boundary of the block. The latter ensures that no (advective) features cross the boundary and the coarse grid blocks are not only computationally but also physically decoupled. Posing Dirichlet boundary conditions on coarse blocks that move with the flow is therefore useful since information flow is not artificially blocked as with fixed coarse blocks.

From the computational point of view our method is quite attractive. It is composed of two parts: an offline phase that precomputes the basis functions and an online phase that uses these basis functions to compute the coarse solution. The overhead of precomputing the basis functions in each coarse block can further be reduced by parallelization as in the classical elliptic MsFEM. The online phase is approximately as fast as a low resolution standard FEM but still reveals fine scale features of a highly resolved solution and is therefore much more accurate than a standard FEM.

Also note that although we suggest to use the precomputed basis functions in a finite element framework that the idea is much more generic. Such modified basis functions can potentially be used in a different global framework that uses a finite volume or discontinuous Galerkin formulation.

We are of course aware of the fact that methods for a simple model like the advection-diffusion equation do not immediately generalize to more important and harder problems like the primitive equations. Nevertheless, we strive to suggest ideas for dealing with the interaction of different parts of such simulations, i.e., between advective and (parametrized) diffusive features that have a multiscale character. Besides that our method can potentially be useful for simpler problems dealing with passive tracer transport. We also point out problems with this approach and with its generalization. At this point we are unfortunately not able to give a profound analysis of the method but we show test cases that will demonstrate the superiority of our modified MsFEM over standard methods in certain flow regimes. A rigourous analysis is left for future work.

Outline. This work is organized as follows. In Section 2 we introduce the modified MsFEM method. This is done systematically starting from an easy setting to the general one. Section 3 shows tests, each of them revealing a different aspect of the modified MsFEM. We conclude with a discussion of strengths and limitations of our method in Section 4.

2 Description of the Method

In this section we will give an overview of our method which is based on multiscale finite element methods. We first introduce the general idea of time-dependent basis functions and formulate the discrete equations. Since this idea does not work in general we then propose transforms of the original equation and reformulate the discretized equation. We develop these transforms hierarchically to provide the reader with a better understanding for the method.

Multiscale finite element methods for elliptic problems have been intensively investigated by the porous media community [13, 15]. In contrast to the HMM [12] they are designed not only to represent the coarse scale features of the solution to the problem correctly but also reveal the fine structure of the solution since the fine scale behavior is resolved by the basis functions. We strive to adopt this approach to one-dimensional transient advection-diffusion equations of the form

$$\begin{aligned} \partial_t u(x,t) + c(x,t)\partial_x u(x,t) &= \partial_x(\mu(x,t)\partial_x u(x,t)) + g(x)\,, \quad && (x,t) \in I \times (0,T] \\ u(0,t) &= u(1,t)\,, && t \in (0,T] \\ u(x,0) &= f(x)\,, && x \in I \end{aligned} \tag{1}$$

on the unit interval $I = [0,1]$ in a periodic setting with rapidly oscillating coefficients $c, \mu \in L^\infty([0,T], L^\infty_{\text{per}}(I))$, and smooth periodic $f, g \in L^2_{\text{per}}(I)$. In order for (1) to be well-posed we assume that $\mu > 0$. Then, by the theory of parabolic equations [14] the weak form of problem (1), i.e., find u with

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}t}\int_I u(x,t)\varphi(x)\,\mathrm{d}x + \int_I c(x,t)\partial_x u(x,t)\varphi(x)\,\mathrm{d}x \\ &\qquad = -\int_I \mu(x,t)\partial_x u(x,t)\partial_x\varphi(x)\,\mathrm{d}x + \int_I g(x)\varphi(x)\,\mathrm{d}x \quad \forall \varphi \in H^1_{\text{per}}(I)\,, t \in (0,T]\,, \end{aligned} \tag{2}$$

has a unique solution u that is in $L^2([0,T], H^1_{\text{per}}(I))$ and its derivative $\partial_t u$ is in $L^2([0,T], H^{-1}_{\text{per}}(I))$. By compactness we additionally have $u \in C([0,T], L^2_{\text{per}}(I))$ and hence the initial condition $u(x,0) = f(x)$ makes sense. We will impose further assumptions on the data in the sequel.

Regime of the Data. Motivated by typical problems that arise in the parametrization of subgrid processes in climate simulations we assume that $\mu = \mu(x/\varepsilon, t)$ and $c = c(x, x/\delta, t)$ with $\varepsilon \ll H$ and $\delta \ll H$ or $\delta \gtrsim H$, where H denotes a the magnitude of a resolved numerical scale. By this we mean that oscillations in the data can not be resolved by H, i.e., one can even think of ϵ, δ as vectors that represent regimes of scales in the data. However, we suppress the ϵ, δ notation unless necessary otherwise. The reader should note that although we keep the idea of possibly fine oscillatory scales $O(\delta)$ in the background velocity the standard case is that $\delta \gtrsim H$. This is because background velocity data usually comes from the coarse simulation scale whereas μ comes from parametrizations and therefore lives on the subgrid scale. However, to be as general as possible we would like to keep both options, i.e., δ to denote resolved or unresolved scales.

Furthermore, we are interested in a regime that is dominated by the advective term. Since the coefficients are assumed to be oscillatory it is less enlightening to use a single Péclet number $\mathrm{Pe} = \|c\|_{L^\infty} L/\mu$, where L is a characteristic length scale, since in our case c can be is a very local quantity as well. Instead we are interested in the regime where $\mathrm{Pe}(x) = c(x,t)L/\mu(x,t)$ is large in average, i.e., we suggest to look at Pe with a large (normalized) L^1-norm. This would mean that the tracer distribution over time is on average dominated by advection, i.e., for the background velocity we assume that it is dominated by its mean.

The Basic Idea of the Method. Multiscale finite elements consist of two main components. A global formulation and modified basis functions, i.e., a localization. In our approach we decompose the domain I into a number coarse cells K_i, $i = 1, \ldots, N-1$ of mesh width H. With each Eulerian node x_m, $m = 1, \ldots, N$, of the coarse mesh we associate a basis function φ^{ms}_m that

satisfies $\varphi_m^{\text{ms}}(x_n) = \delta_{mn}$, i.e., globally we use a conformal finite element formulation. Standard finite elements interpolate the basis functions between the nodes in a prescribed functional way, often polynomial. For the multiscale finite element basis functions used in our global formulation, i.e., the second constituent, we interpolate in such a way that the basis contains information about the fine scale structure of the problem at hand. In the porous media community the common way is to require the basis to satisfy the homogeneous equation locally, i.e., in each coarse grid block, to the highest order. We aim to adopt this idea to transient equations that contain advective terms which is the major difficulty.

In each coarse block K_i we then seek basis functions φ_i^l, $l = 1, 2$, that satisfy

$$\begin{aligned} \partial_t \varphi_i^l(x,t) + c(x,t)\partial_x \varphi_i^l(x,t) &= \partial_x(\mu(x,t)\partial_x \varphi_i^l(x,t)) \,, \quad && (x,t) \in K_i \times (0,T] \\ \varphi_i^l(0,t) &= \varphi_{i,0}^l(0) \,, && t \in (0,T] \\ \varphi_i^l(1,t) &= \varphi_{i,0}^l(1) \,, && t \in (0,T] \\ \varphi_i^l(x,0) &= \varphi_{i,0}^l(x) \,, && x \in K_i \end{aligned} \tag{3}$$

in the weak sense where $\varphi_{i,0}^l$ is the l-th P_1-finite element basis function on the coarse scale. Note that these problems have unique weak solutions $\varphi_i^l \in L^2([0,T], \varphi_{i,0}^l + H_0^1(I))$ with $\partial_t \varphi_i^l \in L^2([0,T], H^{-1}(I))$ and hence $\varphi_i^l \in C([0,T], L^2(I))$. To numerically solve these problems we employ the method of lines, i.e., we discretize in space and then solve the resulting ODE. Using a conformal standard FEM in space the problem reads: Find $\varphi_i^l \in C^1([0,T], V^h(K_i))$ such that

$$\begin{aligned} \int_I \partial_t \varphi_i^{l,h}(x,t)\psi(x)\,\mathrm{d}x + \int_I c(x,t)\partial_x \varphi_i^{l,h}(x,t)\psi(x)\,\mathrm{d}x \\ = -\int_I \mu(x,t)\partial_x \varphi_i^{l,h}(x,t)\partial_x \psi(x)\,\mathrm{d}x \quad \forall \psi \in V^h(K_i)\,, t \in (0,T] \end{aligned} \tag{4}$$

with the initial condition $\phi_i^{l,h}(x,0) = P_{V^h(K_i)}\varphi_{i,0}^l(x)$ where P_{V^h} is the projection onto $V^h(K_i)$. The space $V^h(K_i)$ is given by

$$V^h(K_i) = \left\{ \psi \in H^1(K_i) \mid \psi \text{ is piece-wise polynomial and continuous} \right\} . \tag{5}$$

The nodal basis functions φ_m are then constructed by "gluing" the complementing parts of two basis functions of two adjacent cells, i.e., the basis functions have the same boundary conditions at the intersection node like in a standard FEM. For an illustration see Figure 1.

The basis functions need to be pre-computed in an offline-phase. This can be done in parallel for each cell since the local problems do not depend on each other. In each of the small local problems actually only one basis function φ_i^l has to be computed since the other (unique) solution can be

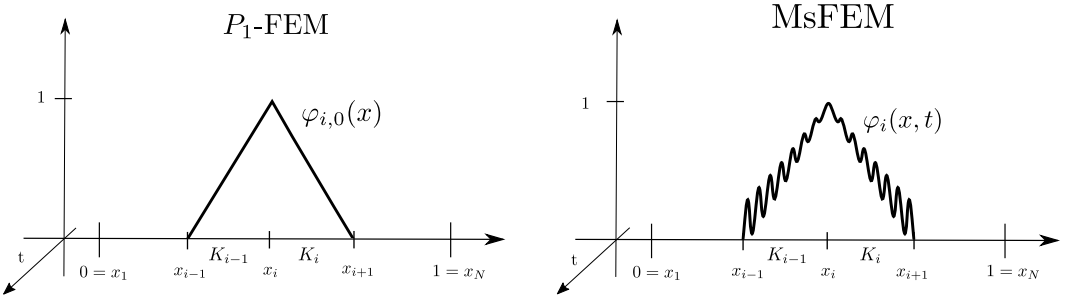


Figure 1: Illustration of basis functions. Standard FEM basis functions $\varphi_{i,0}(x,t) = \varphi_{i,0}(x)$ (left) have constant shape in time and serve as initial conditions for multiscale FEM basis functions $\varphi_i(x,t)$ (right) which does depend on time. The latter is glued together from parts that evolve in each adjacent cell according to the homogeneous equation. It carries the local structure of the flow in space and time.

computed as $\varphi_i^k(x,t) = 1 - \varphi_i^l(x,t)$, $k \neq l$, due to linearity. Consequently the basis functions fulfill $\sum_{j=1}^{N} \varphi_j(x,t) = 1$, i.e., they constitute a decomposition of unity.

For the global formulation, in contrast to standard finite elements where basis functions do not depend on time, we seek solutions of the form

$$u^H(x,t) = u_j^H(t)\varphi_j(x,t)\,. \tag{6}$$

Here we used the Einstein's sum convention. With this ansatz we design our global weak form as: Find $u^H \in V_{\text{per}}^H$ such that

$$\int_I \partial_t u^H(x,t)\varphi(x,t)\,\mathrm{d}x + \int_I c(x,t)\partial_x u^H(x,t)\varphi(x,t)\,\mathrm{d}x \\ = -\int_I \mu(x,t)\partial_x u^H(x,t)\partial_x\varphi(x,t)\,\mathrm{d}x \quad \forall\varphi \in V_{\text{per}}^H(t)\,, t \in (0,T] \tag{7}$$

with the initial condition $u^H(x,0) = P_{V_{\text{per}}^H(0)} f$ where $P_{V_{\text{per}}^H(0)}$ denotes the projection onto the subspace $V_{\text{per}}^H(0) \subset L_{\text{per}}^2(I)$. Using the method of lines the global finite element space V_{per}^H can be formulated as

$$V_{\text{per}}^H = \left\{u^H \in C^1([0,T], H_{\text{per}}^1(I)) \mid u^H(t) \in V_{\text{per}}^H(t)\right\} \tag{8}$$

with

$$V_{\text{per}}^H(t) = \text{span}_m\left\{\varphi_m(\cdot,t) \mid \varphi_m \text{ satisfies the discretized equation (3) at time } t \in [0,T]\right\} \tag{9}$$

Note that a basis function φ_m is glued together from two parts that are element of the space $V^h(K_i)$. This renders the method conformal in space. We use the same constructed basis as basis of the test function space. However, a Petrov-Galerkin formulation using, e.g., standard P_1-basis functions is possible.

Using expression (6) as ansatz for the solution in (7) we get

$$\int_I \varphi_i(x,t)\partial_t\left(u_j^H(t)\varphi_j(x,t)\right)\,\mathrm{d}x + \int_I \varphi_i(x,t)c(x,t)u_j^H(t)\partial_x\varphi_j(x,t)\,\mathrm{d}x \tag{10}$$

$$= -\int_I \partial_x\varphi_i(x,t)\mu(x,t)u_j^H(t)\partial_x\varphi_j(x,t)\,\mathrm{d}x + \int_I \varphi_i(x,t)g(x)\,\mathrm{d}x \quad \forall i = 1,\dots,N\,, t \in (0,T]\,. \tag{11}$$

which can be expressed as an ODE:

$$M(t)\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{u}^H(t) + N(t)\mathbf{u}^H(t) + A(t)\mathbf{u}^H(t) = D(t)\mathbf{u}^H(t) + \mathbf{G}(t)\,, \tag{12}$$

where $\mathbf{u}^H(t) = (u_1^H(t),\dots,u_N^H(t))^T$, $\mathbf{G}(t)_i = \int_I \varphi_i(x,t)g(x)\,\mathrm{d}x$ and

$$\begin{aligned} M(t)_{ij} &= \int_I \varphi_i(x,t)\varphi_j(x,t)\,\mathrm{d}x\,, \quad N(t)_{ij} = \int_I \varphi_i(x,t)\partial_t\varphi_j(x,t)\,\mathrm{d}x\,, \\ A(t)_{ij} &= \int_I \varphi_i(x,t)c(x,t)\partial_x\varphi_j(x,t)\,\mathrm{d}x\,, \quad \text{and} \quad D(t)_{ij} = -\int_I \partial_x\varphi_i(x,t)\mu(x,t)\partial_x\varphi_j(x,t)\,\mathrm{d}x \end{aligned} \tag{13}$$

and with the initial condition $\mathbf{u}(0)_i = \int_I \varphi(x,0)f(x)\,\mathrm{d}x$. Note that the second term $N(t)$ in (12) appears since the basis depends on time. Also note that this is a purely Eulerian setting.

Now, one could simply solve this ODE using a suitable time integrator. Unfortunately, this method fails due to the dominance of the advective term. Simply taking small time steps does not remedy the problem since the problem lies in the way we compute the solutions to the local problems. The basis functions are supposed to carry information about the fine scale structure of the solution but note that the local problems are posed with Dirichlet boundary conditions. Such type of boundary condition essentially blocks advected information at coarse cell boundaries since

all coarse cells are decoupled from each other. Hence, for flow dominated by advection multiscale basis functions constructed according to (3) would develop a boundary layer that is not present in the global flow. The "correct" choice of type of boundary conditions is indeed a general problem that one is faced with when solving such local problems [13].

A Modified Idea. We try to circumvent the problem above by moving to a different set of coordinates that make the advective term milder, e.g., we could follow a suitable set of curves emerging at our coarse grid points as done in a full Lagrangian framework. Then Dirichlet boundary conditions could be posed on these curves since these naturally decouple flow regions. Intuitively this would give better results since the transform will bring us "closer" to a parabolic setting. However, a full Lagrangian framework is, in general, difficult to handle since it is not easy to implement and since characteristics could potentially come very close (or even cross) which requires a special treatment in such regions. We therefore propose a simpler method.

We start by explaining our idea in a simple setting and then generalize. For this let the background velocity $c(x,t) = c(t)$, i.e., we consider a flow without shear. This special assumption actually allows us to move to a full Lagrangian setting since we can simply follow characteristics and know they will not intersect. The new coordinates (ξ,τ) are then (implicitly) given by

$$\begin{aligned} x(\xi,\tau) &= \xi + \int_0^\tau c(s)\,\mathrm{d}s \\ t(\xi,\tau) &= \tau\,. \end{aligned} \tag{14}$$

A little bit more general is the choice

$$\begin{aligned} x(\xi,\tau) &= \xi + \int_0^\tau \langle c\rangle(s)\,\mathrm{d}s\,, \quad \text{where} \quad \langle c\rangle(t) = \frac{1}{|I|}\int_I c(x,t)\,\mathrm{d}x \\ t(\xi,\tau) &= \tau \end{aligned} \tag{15}$$

if c explicitly depends on space. In the new coordinates (15) equation (1) reduces to

$$\begin{aligned} \partial_\tau \hat{u}(\xi,\tau) + \tilde{c}(\xi,\tau)\partial_\xi \hat{u}(\xi,\tau) &= \partial_\xi(\hat{\mu}(\xi,\tau)\partial_\xi \hat{u}(\xi,\tau)) + \hat{g}(\xi)\,, \quad (\xi,\tau) \in I \times (0,T] \\ \hat{u}(0,\tau) &= \hat{u}(1,\tau)\,, \qquad \tau \in (0,T] \\ \hat{u}(\xi,0) &= \hat{f}(\xi)\,, \qquad \xi \in I \end{aligned} \tag{16}$$

where $\hat{u}(\xi,\tau) = u(x(\xi,\tau),t(\tau))$, respectively $\hat{c},\hat{\mu},\hat{f},\hat{g}$, and

$$\tilde{c}(\xi,\tau) = \hat{c}(\xi,\tau) - \langle c\rangle(\tau)\,. \tag{17}$$

Note that if c is constant or only depends on time, i.e., we employ (14), then $\tilde{c}(\xi,\tau) = 0$ and (16) reduces to a simple diffusion equation. An illustration of the two transforms is done in Figure 2(a) and (b).

The subscale problem for the basis functions (3) needs to be transformed as well: Find a (weak) solution $\hat{\varphi}_i^l(\xi,\tau)$, $l = 1,2$, such that

$$\begin{aligned} \partial_\tau \hat{\varphi}_i^l(\xi,\tau) + \tilde{c}(\xi,\tau)\partial_\xi \hat{\varphi}_i^l(\xi,\tau) &= \partial_\xi(\hat{\mu}(\xi,\tau)\partial_\xi \hat{\varphi}_i^l(\xi,\tau))\,, \quad (\xi,\tau) \in \hat{K}_i \times (0,T] \\ \hat{\varphi}_i^l(0,\tau) &= \varphi_{i,0}^l(0)\,, \qquad \tau \in (0,T] \\ \hat{\varphi}_i^l(1,\tau) &= \varphi_{i,0}^l(1)\,, \qquad \tau \in (0,T] \\ \hat{\varphi}_i^l(\xi,0) &= \varphi_{i,0}^l(\xi)\,, \qquad \xi \in \hat{K}_i\,. \end{aligned} \tag{18}$$

A cell $\hat{K}_i$ then essentially moves with the flow described by the transform without being deformed.

A Generalized Modified Idea. In case of a large scale separation between the mean flow and the oscillatory parts of the background velocity the method using transform (15) delivers good results. However, as mentioned earlier, in many practically relevant scenarios in climate simulations the background velocity does not exhibit such a scale separation, i.e., there are mostly (globally resolved) scales $\delta \gtrsim H$. We would nevertheless like to keep possible small scale variations in c to be as general as possible. Variations at a scale $O(H)$ and larger would have the consequence that the average velocity $\tilde{c}$ in each cell $\hat{K}_i$ can potentially be large in contrast to the case where one has a scale separation. Consequently basis functions following the evolution given by (18) can still exhibit steep boundary layers that are not apparent in the actual flow. As a remedy one can make an effort to stay "close" to the full Lagrangian setting locally, i.e., we can track the characteristics (particle trajectories) of the flow starting at the coarse nodes. Boundary conditions for the multiscale basis functions can then be posed on these characteristics. In scenarios in which characteristics do not come too close or diverge too strongly this is, as we will show in numerical examples, a reasonable strategy. Fortunately, in many situations in climate simulations there is no situation in which true shocks occur, e.g., wind field data that is taken from a coarse grid does not behave too badly.

To formalize this strategy let $x(\xi, \tau)$ denote the characteristics emerging from $\xi \in I$. The evolution is governed by the set of ODEs that reads

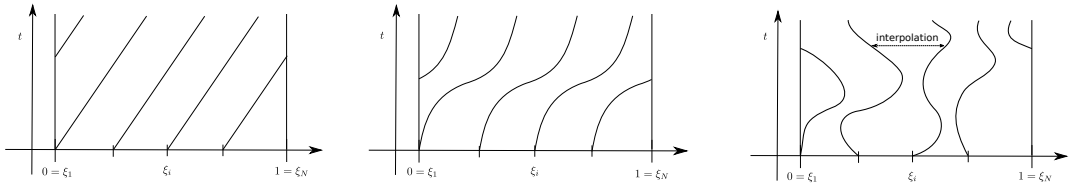
$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}\tau} x(\xi,\tau) &= c(x(\xi,\tau),\tau)\,, \quad x(\xi,0) = \xi \\ \frac{\mathrm{d}}{\mathrm{d}\tau} t(\xi,\tau) &= 1\,, \quad t(\xi,0) = 0\,. \end{aligned} \tag{19}$$

This evolution induces a differentiable transform between (x,t)-coordinates and (ξ,τ)-coordinates as long as characteristics do not intersect. The latter is the case if $c(x,t)$ is locally Lipschitz-continuous in x and continuous in t. If we were to use transform (19) everywhere in our domain (what we do not pursue) this would simply mean a change from Eulerian to Lagrangian variables. The following computations are formal and only hold as long as trajectories do not cross. As a first step we will rewrite the equations in characteristic coordinates. For the gradients we have the relation

$$\nabla_{(x,t)}(\cdot) = \nabla_{(\xi,\tau)}(\cdot) \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial t} \\ \frac{\partial \tau}{\partial x} & \frac{\partial \tau}{\partial t} \end{pmatrix} \tag{20}$$

and

$$\nabla_{(\xi,\tau)}(\cdot) = \nabla_{(x,t)}(\cdot) \begin{pmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \tau} \\ \frac{\partial t}{\partial \xi} & \frac{\partial t}{\partial \tau} \end{pmatrix} \tag{21}$$



$c(x,t) \equiv \text{const}$
$x(\xi,\tau) = \xi + c\tau$
$t(\xi,\tau) = \tau$

(a)

$c = c(t)$ or $c = c(x,t)$
$x(\xi,\tau) = \xi + \int_0^\tau \langle c \rangle(s)\mathrm{d}s$
$t(\xi,\tau) = \tau$

(b)

$c = c(x,t)$
$\frac{\mathrm{d}}{\mathrm{d}\tau} x(\xi,\tau) = c(x(\xi,\tau),\tau)$
$\frac{\mathrm{d}}{\mathrm{d}\tau}\, t(\xi,\tau) = 1$

(c)

Figure 2: Illustration of different transforms. **(a)** If the background velocity $c(x,t)$ is constant transform (14) gives a pure diffusion equation. **(b)** A spatially dependent c is averaged over the entire domain and one follows the characteristics of the mean flow by transform (15). In the special case $c(x,t) = c(t)$ then the average equals $c(t)$ and the transformed equation is purely parabolic. **(c)** Transform (19) follows the true characteristics that emerge from each coarse node ξ_i and interpolates them linearly inside the coarse cells.

Hence, we get

$$\begin{aligned} \partial_x u(x,t) &= \partial_\xi \hat{u}(\xi,\tau)\frac{\partial \xi}{\partial x} + \partial_\tau \hat{u}(\xi,\tau)\frac{\partial \tau}{\partial x} \\ \partial_t u(x,t) &= \partial_\xi \hat{u}(\xi,\tau)\frac{\partial \xi}{\partial t} + \partial_\tau \hat{u}(\xi,\tau)\frac{\partial \tau}{\partial t}\,. \end{aligned} \tag{22}$$

By means of the inverse function theorem the Jacobian in (21) is (locally) the inverse of the Jacobian in (20). Using Kramer's rule the inverse is given by

$$\begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial t} \\ \frac{\partial \tau}{\partial x} & \frac{\partial \tau}{\partial t} \end{pmatrix} = \frac{1}{D}\begin{pmatrix} \frac{\partial t}{\partial \tau} & -\frac{\partial x}{\partial \tau} \\ -\frac{\partial t}{\partial \xi} & \frac{\partial x}{\partial \xi} \end{pmatrix} \tag{23}$$

where

$$D = \frac{\partial x}{\partial \xi}\frac{\partial t}{\partial \tau} - \frac{\partial x}{\partial \tau}\frac{\partial t}{\partial \xi} \tag{24}$$

is the determinant of the Jacobian in (21).

Now, using (19) in (22) we see that

$$\partial_t u(x,t) + c(x,t)\partial_x u(x,t) = \partial_\tau \hat{u}(\xi,\tau) + \left[\hat{c}(\xi,\tau) - \frac{\partial x}{\partial \tau}\right]\left(\frac{\partial x}{\partial \xi}\right)^{-1}\hat{u}(\xi,\tau)\,. \tag{25}$$

Looking again at (19) we see that

$$\frac{\partial x}{\partial \tau} = \hat{c}(\xi,\tau) \tag{26}$$

and hence the velocity term in the transformed equation would vanish. The advantage of such a full Lagrangian setting would be that one has then a purely parabolic equation as in the case when the background velocity is constant or solely depends on time. But transform (19) breaks down in the presence of shocks. However, true shocks can not occur since the overall equation we are dealing with is parabolic (although dominated by advection) and strictly speaking there are no characteristics. We would nonetheless still like to reduce the advective effects and retain the "nice" part of the equation. This is where our framework starts to differ from the full Lagrangian setting.

The idea is now, as mentioned earlier, to pose Dirichlet boundary conditions on the characteristics in order to solve subscale problems similar to (18). In order to do this we only need the coarse cell boundaries to be characteristics of the advective term. Inside each coarse cell we just need to reduce the advective effects of the original equation. This will then prevent flow across coarse cell boundaries since near the boundary the transformed equation will be nearly parabolic. Further, the solution to the equation inside the coarse cell will reflect all the subscale features of the solution of the original equation.

The diffusive term transforms to

$$\partial_x(\mu(x,t)\partial_x u(x,t)) = \partial_\xi\left[\hat{\mu}(\xi,\tau)\partial_\xi \hat{u}(\xi,\tau)\left(\frac{\partial x}{\partial \xi}\right)^{-1}\right]\left(\frac{\partial x}{\partial \xi}\right)^{-1}\,. \tag{27}$$

Collecting (25) and (27) together with the boundary conditions we are now ready to introduce the transformed problem for the multiscale basis functions as: By the method of lines find a

(variational) solution $\hat{\varphi}_i^l \in C^1([0,T], V^h(\hat{K}_i))$, $l = 1, 2$, such that

$$\begin{aligned}
\partial_\tau \hat{\varphi}_i^l(\xi,\tau) + \tilde{c}(\xi,\tau)\partial_\xi \hat{\varphi}_i^l(\xi,\tau) &= \partial_\xi \left[\hat{\mu}(\xi,\tau) \left(\frac{\partial x}{\partial \xi}\right)^{-1} \partial_\xi \hat{\varphi}_i^l(\xi,\tau) \right] \left(\frac{\partial x}{\partial \xi}\right)^{-1} \\
&\quad + \hat{g}(\xi,\tau) \qquad , (\xi,\tau) \in \hat{K}_i \times (0,T] \\
\hat{\varphi}_i^l(0,\tau) &= \varphi_{i,0}^l(0) \; , \qquad \tau \in (0,T] \\
\hat{\varphi}_i^l(1,\tau) &= \varphi_{i,0}^l(1) \; , \qquad \tau \in (0,T] \\
\hat{\varphi}_i^l(\xi,0) &= \varphi_{i,0}^l(\xi) \; , \qquad \xi \in \hat{K}_i
\end{aligned} \tag{28}$$

where

$$\tilde{c}(\xi,\tau) = \left[\hat{c}(\xi,\tau) - \frac{\partial x}{\partial \tau} \right] \left(\frac{\partial x}{\partial \xi}\right)^{-1} \tag{29}$$

and

$$V^h(\hat{K}_i) = \left\{ \psi \in H^1(\hat{K}_i) \mid \hat{\varphi} \text{ is piece-wise polynomial and continuous} \right\} . \tag{30}$$

In each of the $N-1$ coarse cells $\hat{K}_i = [\xi_i, \xi_{i+1}]$ we now have to compute two additional quantities, i.e., $\partial x/\partial \xi$ and $\partial x/\partial \tau$. Computation of the former is based on linear interpolation in space between the left and right boundary nodes of $\hat{K}_i$ which follow the characteristics starting at ξ_i and ξ_{i+1}. (more details in the paragraph about the implementation). For the latter, i.e., $\partial x/\partial \tau$, we use the fact that $\partial x/\partial \tau(\xi_i, \tau) = \hat{c}(\xi_i, \tau)$ at each coarse node. Inside the cells $\hat{K}_i$ we can again interpolate linearly the values $\hat{c}(\xi_i, \tau)$ and $\hat{c}(\xi_{i+1}, \tau)$. Note that now $\tilde{c}(\xi_i, \tau) = 0$ at all coarse nodes ξ_i globally in time. This means that characteristics of the transformed system starting at the coarse nodes are separatrices of the dynamical system $\mathrm{d}/\mathrm{d}s\xi(s) = \tilde{c}(\xi(s), s)$ and hence no flow across coarse cell boundaries is possible in these coordinates, i.e., the flow is separated by the cells $\hat{K}_i$.

The global weak form can be formulated as: Find $\hat{u}(\xi,\tau) \in \hat{V}_{\mathrm{per}}^H$ such that

$$\begin{aligned}
&\int_I \partial_\tau \hat{u}^H(\xi,\tau)\hat{\varphi}^H(\xi,\tau)\,\mathrm{d}\xi + \int_I \tilde{c}(\xi,\tau)\partial_\xi u^H(\xi,\tau)\hat{\varphi}^H(\xi,\tau)\,\mathrm{d}\xi \\
&\qquad = -\int_I \left[\hat{\mu}(\xi,\tau)\partial_\xi \hat{u}^H(\xi,\tau) \left(\frac{\partial x}{\partial \xi}\right)^{-1} \right] \partial_\xi \left[\hat{\varphi}^H(\xi,\tau) \left(\frac{\partial x}{\partial \xi}\right)^{-1} \right] \mathrm{d}\xi \\
&\qquad\qquad + \int_I \hat{g}(\xi,\tau)\hat{\varphi}^H(\xi,\tau)\,\mathrm{d}\xi \quad \forall \hat{\varphi}^H \in V_{\mathrm{per}}^H(\tau)\,, \tau \in (0,T]
\end{aligned} \tag{31}$$

where $\tilde{c}$ is given by (29). The initial condition is $\hat{u}^H(\xi,0) = P_{\hat{V}_{\mathrm{per}}^H(0)} f$ where $P_{\hat{V}_{\mathrm{per}}^H(0)}$ denotes the projection onto the subspace $\hat{V}_{\mathrm{per}}^H(0) \subset L_{\mathrm{per}}^2(I)$. The global finite element space $\hat{V}_{\mathrm{per}}^H$ can be formulated as

$$\hat{V}_{\mathrm{per}}^H = \left\{ \hat{u} \in C^1([0,T], H_{\mathrm{per}}^1(I)) \mid \hat{u}(\tau) \in \hat{V}_{\mathrm{per}}^H(\tau) \right\} \tag{32}$$

with

$$\hat{V}_{\mathrm{per}}^H(\tau) = \mathrm{span}_m \left\{ \hat{\varphi}_m \mid \hat{\varphi}_m \text{ satisfies the discretized equation (28) at time } \tau \in [0,T] \right\} . \tag{33}$$

Note that this formulation includes the ones using the transforms (14) or (15) and that it is not fully Lagrangian if one uses (19).

The Algorithm and some Notes on the Implementation. In this paragraph we give a brief overview of steps to be taken for the implementation of our above described method.

Step 1. The first step is to set up a uniform mesh with nodes ξ_m, $m = 1, \ldots, N$, and cells $\hat{K}_i$, $i = 1, \ldots, N-1$ with mesh size $\epsilon, \delta \gtrsim h$. Now one decides for the transform. For each node ξ_m either uses (14) or (15) or one solves the ODE (19) on the time interval $[0,T]$ using a suitable

solver. For our examples we used an adaptive Runge-Kutta-4/5 solver.

Step 2. One now needs to solve for the basis in an offline phase. This can be done in parallel since all local problems are independent of each other. As indicated earlier we solve equation (28) using a conformal standard P_1-finite element formulation. For this we create in each cell $\hat{K}_i$ a uniform mesh with cells $\hat{k}_j^i$, $i = 1, \dots, N_f - 1$ and nodes ξ_n^m, $n = 1, \dots, N_f$, with $\xi_1^m = \xi_m$ and $\xi_{N_f}^m = \xi_{m+1}$. In cell $\hat{K}_i$ the discretized ODE for the vector $\hat{\boldsymbol{\varphi}}_i^{l,H}(\tau)$ of degrees of freedom of the basis function $\hat{\varphi}_i^{l,H}(\tau)$ reads

$$\hat{M}^{\hat{K}_i} \frac{\mathrm{d}}{\mathrm{d}\tau} \hat{\boldsymbol{\varphi}}_i^{l,H}(\tau) + \hat{A}(\tau)^{\hat{K}_i} \hat{\boldsymbol{\varphi}}_i^{l,H}(\tau) = \hat{D}(\tau)^{\hat{K}_i} \hat{\boldsymbol{\varphi}}_i^{l,H}(\tau) \,,\ \hat{\boldsymbol{\varphi}}_i^{l,H}(0) = \boldsymbol{\varphi}_{i,0}^{l,H} \tag{34}$$

where $\boldsymbol{\varphi}_{i,0}^{l,h}$ denotes the nodal values of the standard basis. The system matrices are given by

$$\begin{aligned}
\hat{M}_{kl}^{\hat{K}_i} &= \int_{\hat{K}_i} \psi_k^h(\xi) \psi_l^h(\xi) \,\mathrm{d}\xi \,, \\
\hat{A}(\tau)_{kl}^{\hat{K}_i} &= \int_{\hat{K}_i} \psi_k^h(\xi) \tilde{c}(\xi, \tau) \partial_\xi \psi_l^h(\xi) \,\mathrm{d}\xi \,, \\
\hat{D}(\tau)_{kl}^{\hat{K}_i} &= -\int_{\hat{K}_i} \left[\hat{\mu}(\xi, \tau) \partial_\xi \psi_k^h(\xi) \left(\frac{\partial x}{\partial \xi} \right)^{-1} \right] \left[\psi_l^h(\xi) \left(\frac{\partial x}{\partial \xi} \right)^{-1} \right] \mathrm{d}\xi
\end{aligned} \tag{35}$$

where ψ_k^h denote the basis of standard that consists of the common hat functions. Note again that the computation of $A(\tau)$ and $D(\tau)$ involves terms of the form

$$\frac{\partial x}{\partial \tau} \quad \text{and} \quad \left(\frac{\partial x}{\partial \xi} \right)^{-1} . \tag{36}$$

Time stepping is done using a second-order explicit scheme for the advective term and an (implicit) Crank-Nicolson scheme for the diffusion. Note that the advection matrix and the diffusion matrix need to be assembled at each time step. All system matrices in (35) are stored for each time step (for a reason see Step 3).

In each cell actually only one basis function $\hat{\varphi}_i^l$ needs to be computed. By linearity the other (unique) solution is then given by $\varphi_i^k(x,t) = 1 - \varphi_i^l(x,t)$, $k \neq l$. Hence the constructed basis functions globally constitute a decomposition of unity.

Step 3. The global system (31) is solved in an online phase. The discretized equations form a system of ODEs of size $(N-1)$-by-$(N-1)$ (note the periodic boundary conditions):

$$\hat{M}(\tau) \frac{\mathrm{d}}{\mathrm{d}\tau} \hat{\mathbf{u}}^H(\tau) + \hat{N}(\tau) \hat{\mathbf{u}}^H(\tau) + \hat{A}(\tau) \hat{\mathbf{u}}^H(\tau) = \hat{D}(\tau) \hat{\mathbf{u}}^H(\tau) + \hat{G}(\tau) \tag{37}$$

where $\hat{G}(\tau) = \int_I \hat{\varphi}_i^H(\xi, \tau) \hat{g}(\xi, \tau) \,\mathrm{d}\xi$,

$$\begin{aligned}
\hat{M}(\tau)_{ij} &= \int_I \hat{\varphi}_i^H(\xi,\tau) \hat{\varphi}_j^H(\xi,\tau) \,\mathrm{d}\xi \,, \quad \hat{N}(\tau)_{ij} = \int_I \hat{\varphi}_i^H(\xi,\tau) \partial_t \hat{\varphi}_j^H(\xi,\tau) \,\mathrm{d}\xi \,, \\
\hat{A}(\tau)_{ij} &= \int_I \hat{\varphi}_i^H(\xi,\tau) \tilde{c}(\xi,\tau) \partial_\xi \hat{\varphi}_j^H(\xi,\tau) \,\mathrm{d}\xi \,, \quad \text{and} \quad \hat{D}(\tau)_{ij} = -\int_I \partial_\xi \hat{\varphi}_i^H(\xi,\tau) \mu(\xi,\tau) \partial_\xi \hat{\varphi}_j^H(\xi,\tau) \,\mathrm{d}\xi \,.
\end{aligned} \tag{38}$$

For the time stepping we again use the same second-order explicit scheme for the advective term and the (implicit) Crank-Nicolson scheme for the diffusion that we already used for the basis. Since all the matrices in (38) need to be assembled at each time step solving the global system can potentially become very expensive. By using a trick looping over all coarse cells and then over all fine cells can be circumvented since we stored all system matrices in (35). To see this note that

each global basis function is a linear combination of fine scale (standard) basis functions:

$$\hat{\varphi}_i^H(\xi,\tau) = \alpha_{ij}(\tau)\hat{\psi}_j^h(\xi) \tag{39}$$

As a generic example we show this for the advective term. By virtue of (39) we have for the global element (advection) matrix for the element $\hat{K}_n$

$$\begin{aligned}\hat{A}(\tau)_{kl}^n &= \left(\int_{\hat{K}_n} \varphi_i^H(\xi,\tau)\tilde{c}(\xi,\tau)\partial_\xi\varphi_j^H(\xi,\tau)\,\mathrm{d}\xi\right)_{kl} \\ &= \int_{\hat{K}_n} \alpha_{ik}(\tau)\hat{\psi}_k^h(\xi)\tilde{c}(\xi,\tau)\partial_\xi\alpha_{jl}(\tau)\hat{\psi}_l^h(\xi)\,\mathrm{d}\xi \\ &= \alpha\hat{A}^{\hat{K}_n}(\tau)\alpha^T\end{aligned} \tag{40}$$

where $\hat{A}^{\hat{K}_n}(\tau)$ is the stored (advection) matrix of the coarse cell $\hat{K}_n$ given by (35). Hence, assembling the global system at each time step breaks down in collecting information from each coarse block and a simple matrix multiplication. This renders the online phase very fast. Also, to ensure the unique solvability of the characteristic equation we additionally need to assume that $c(x,\cdot)$ be continuous and that $c(\cdot,t)$ be Lipschitz-continuous.

3 Numerical Tests

In this section we demonstrate our method on a number of examples. For all examples presented here we use the same initial condition. In our case this is a normalized Gaussian wave package, see Figure 3 which is given by

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{(x-\nu)^2}{2\sigma^2}\right) \tag{41}$$

All data and all results will be set and shown from an Eulerian point of view. Since no analytical reference solutions are available we will compute a reference solution by a high-resolution standard FEM (750 elements, first order in space) in mean flow coordinates (15), i.e., the advection dominance is already reduced. For our multiscale FEM we will use $N = 10$ coarse elements and $N_f = 75$ fine elements in each coarse cell. Basis functions are computed with a standard FEM. To show the advantage of our multiscale FEM we will compare it to a low-resolution standard FEM that uses mean flow coordinates like the reference solution and $N = 10$ elements, i.e., as many elements as the multiscale method uses on the coarse scale. All simulations will be carried out on the time intervall $[0,T]$ with $T = 1$. We fix these parameters unless we explicitly say something else.

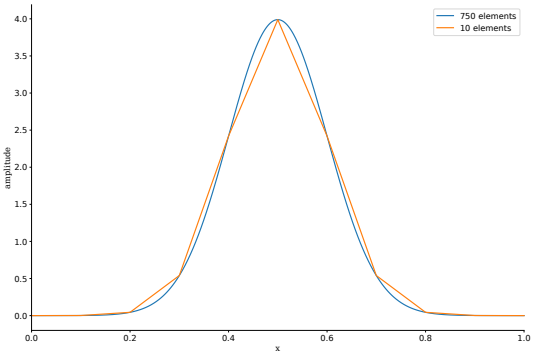


Figure 3: Initial condition for all tests is given by (41).

Case 1. We start with a simple setting. This test is a direct generalization of existing methods for elliptic equations. We have time dependent coefficients given by

$$\begin{aligned}c(t) &= 5\cos(10\pi t)\,, \\ \mu(x,t) &= 5(t+1)\left(0.01 + 0.0099\cos(2\pi k x)\right)\,.\end{aligned} \tag{42}$$

Note that the background velocity only depends on time. Therefore, characteristics can never cross and hence transforms (14), (15) and (19) are identical. Snapshots of the solution and the corresponding errors to the reference solution are shown at four different time stamps in Figure 5 and error graphs are shown in Figure 6. The characteristics induced by the background velocity are shown in Figure 4 and Table 1 summarizes the errors in $L^2(I)$ and $L^\infty(I)$ for a number of different k. Since the multiscale solution resolves the local structure of the solution within coarse elements we also show the error of the derivative, i.e., the $H^1(I)$-error. This can also indirectly be seen in Figure 5 since the error of the multiscale solution is smoother than the one of the standard solution. The time step was taken to be $\delta t = 10^{-3}$.

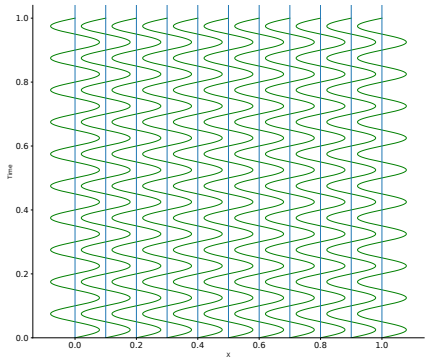


Figure 4: Characteristics emerging at each coarse node induced by the velocity coefficient in (42) with Characteristics originating at $\xi = 0$ and $\xi = 1$ are identical due to periodic boundary conditions.

	$L^2(I)$		$L^\infty(I)$		$H^1(I)$	
	FEM	MsFEM	FEM	MsFEM	FEM	MsFEM
$k = 15$	$1.036 \cdot 10^{-4}$	$1.848 \cdot 10^{-5}$	$1.141 \cdot 10^{-4}$	$2.258 \cdot 10^{-5}$	$5.622 \cdot 10^{-4}$	$2.160 \cdot 10^{-4}$
$k = 30$	$1.770 \cdot 10^{-4}$	$1.437 \cdot 10^{-5}$	$1.809 \cdot 10^{-4}$	$1.785 \cdot 10^{-5}$	$6.329 \cdot 10^{-4}$	$1.706 \cdot 10^{-4}$
$k = 45$	$2.054 \cdot 10^{-4}$	$1.160 \cdot 10^{-5}$	$2.031 \cdot 10^{-4}$	$1.460 \cdot 10^{-5}$	$6.299 \cdot 10^{-4}$	$1.556 \cdot 10^{-4}$
$k = 60$	$2.370 \cdot 10^{-4}$	$1.143 \cdot 10^{-5}$	$2.293 \cdot 10^{-4}$	$1.405 \cdot 10^{-5}$	$6.523 \cdot 10^{-4}$	$1.887 \cdot 10^{-4}$

Table 1: Relative error of case 1 at $T = 1$. Coefficients are given by (42).

Case 2. This test is to compare the two MsFEM versions using either transform (15), i.e., the mean flow transform (MF-MsFEM), or transform (19) which we will refer to as the characteristic MsFEM (Char-MsFEM). We test the influence of oscillations in the background velocity that are either in the resolved part of the spectrum or in the unresolved part. By this we mean that we compare scenarios in which oscillations can or can not be resolved by the coarse grid. The coefficients are given by

$$\begin{aligned} c(x,t) &= 10 + \cos(2k\pi x)\,, \\ \mu(x,t) &= 5(t+1)\,(0.01 + 0.0099\cos(60\pi x))\,. \end{aligned} \tag{43}$$

Here the advective term is on average stronger than the diffusive term by a factor of approximately $5 * 10^2$ (comparing the mean values) and by a factor of 10^5 in regions where he diffusivity has a minimum. Oscillations in the background velocity with $k < 5$ can be resolved on a grid with only $N = 10$ elements whereas oscillations of the diffusivity can not be resolved. Figure 7 shows snapshots of the solution for the case $k = 3$ and the case $k = 60$. Errors are shown in Figure 8. In the first case the Char-MsFEM shows superior performance over both the standard method and the MF-MsFEM. In the case of a larger frequency separation between the mean velocity and the oscillatory part one can see that the MF-MsFEM performs slightly better than the Char-MsFEM. Errors are summarized in Table 2. Note that the interesting scenario for us is the one where $k < 5$, i.e., the background velocity can be resolved by the coarse grid since typically in climate simulations velocities are taken from a coarse mesh and the diffusion parameter includes parametrizations of processes that are not resolved. In such a scenario our Char-MsFEM outperforms the MF-MsFEM whereas if $k > 5$ we observe a similar to slightly better performance of the MF-MsFEM.

	$L^2(I)$		
	FEM	MF-MsFEM	Char-MsFEM
$k = 3$	$4.052 \cdot 10^{-5}$	$4.284 \cdot 10^{-5}$	$1.038 \cdot 10^{-5}$
$k = 15$	$4.080 \cdot 10^{-5}$	$3.205 \cdot 10^{-6}$	$6.787 \cdot 10^{-6}$
$k = 30$	$1.171 \cdot 10^{-4}$	$1.634 \cdot 10^{-5}$	$7.543 \cdot 10^{-6}$
$k = 60$	$4.765 \cdot 10^{-5}$	$7.046 \cdot 10^{-6}$	$2.868 \cdot 10^{-5}$
	$L^\infty(I)$		
	FEM	MF-MsFEM	Char-MsFEM
$k = 3$	$5.258 \cdot 10^{-5}$	$5.404 \cdot 10^{-5}$	$1.743 \cdot 10^{-5}$
$k = 15$	$5.071 \cdot 10^{-5}$	$6.818 \cdot 10^{-6}$	$1.542 \cdot 10^{-5}$
$k = 30$	$1.385 \cdot 10^{-4}$	$2.332 \cdot 10^{-5}$	$1.087 \cdot 10^{-5}$
$k = 60$	$5.969 \cdot 10^{-5}$	$9.408 \cdot 10^{-6}$	$3.863 \cdot 10^{-5}$
	$H^1(I)$		
	FEM	MF-MsFEM	Char-MsFEM
$k = 3$	$5.201 \cdot 10^{-4}$	$4.108 \cdot 10^{-4}$	$2.987 \cdot 10^{-4}$
$k = 15$	$5.185 \cdot 10^{-4}$	$2.539 \cdot 10^{-4}$	$2.609 \cdot 10^{-4}$
$k = 30$	$7.323 \cdot 10^{-4}$	$3.467 \cdot 10^{-4}$	$2.665 \cdot 10^{-4}$
$k = 60$	$5.558 \cdot 10^{-4}$	$2.459 \cdot 10^{-4}$	$5.210 \cdot 10^{-4}$

Table 2: Relative error of case 2 at $T = 1$. Coefficients are given by (43).

	$L^2(I)$		$L^\infty(I)$	
	FEM	Char-MsFEM	FEM	Char-MsFEM
$v = 4$	$2.688 \cdot 10^{-4}$	$5.635 \cdot 10^{-5}$	$2.692 \cdot 10^{-4}$	$6.779 \cdot 10^{-5}$
$v = 8$	$1.722 \cdot 10^{-4}$	$4.731 \cdot 10^{-5}$	$1.684 \cdot 10^{-4}$	$6.077 \cdot 10^{-5}$
$v = 12$	$1.286 \cdot 10^{-4}$	$2.213 \cdot 10^{-5}$	$1.281 \cdot 10^{-4}$	$2.848 \cdot 10^{-5}$
$v = 16$	$9.850 \cdot 10^{-4}$	$2.736 \cdot 10^{-5}$	$9.924 \cdot 10^{-4}$	$3.874 \cdot 10^{-5}$

Table 3: Relative error of case 3 at $T = 1$. Coefficients are given by (44).

Case 3. Here we demonstrate the advantage of the Char-MsFEM in regimes dominated by advection where we have frequencies in the background velocity that are resolved and that are unresolved by the coarse grid. We do not show a comparison here to the MF-MsFEM since this method fails if resolved frequencies are apparent. Furthermore, we use an oscillatory diffusion coefficient as we did in previous other tests. Coefficients for this test are given by

$$\begin{aligned} c(x,t) &= v + 1.5\cos(2\pi x) + 0.5\cos(60\pi x)\,, \\ \mu(x,t) &= 5(t+1)\left(0.01 + 0.0099\cos(50\pi x)\right)\,. \end{aligned} \tag{44}$$

Snapshots of the solution and errors for $v = 4$ are shown in Figure 9 as well as a summary of errors in Table 3. From the snapshots it is clearly visible that the standard FEM does not capture the correct large scale behavior, i.e., it is too fast.

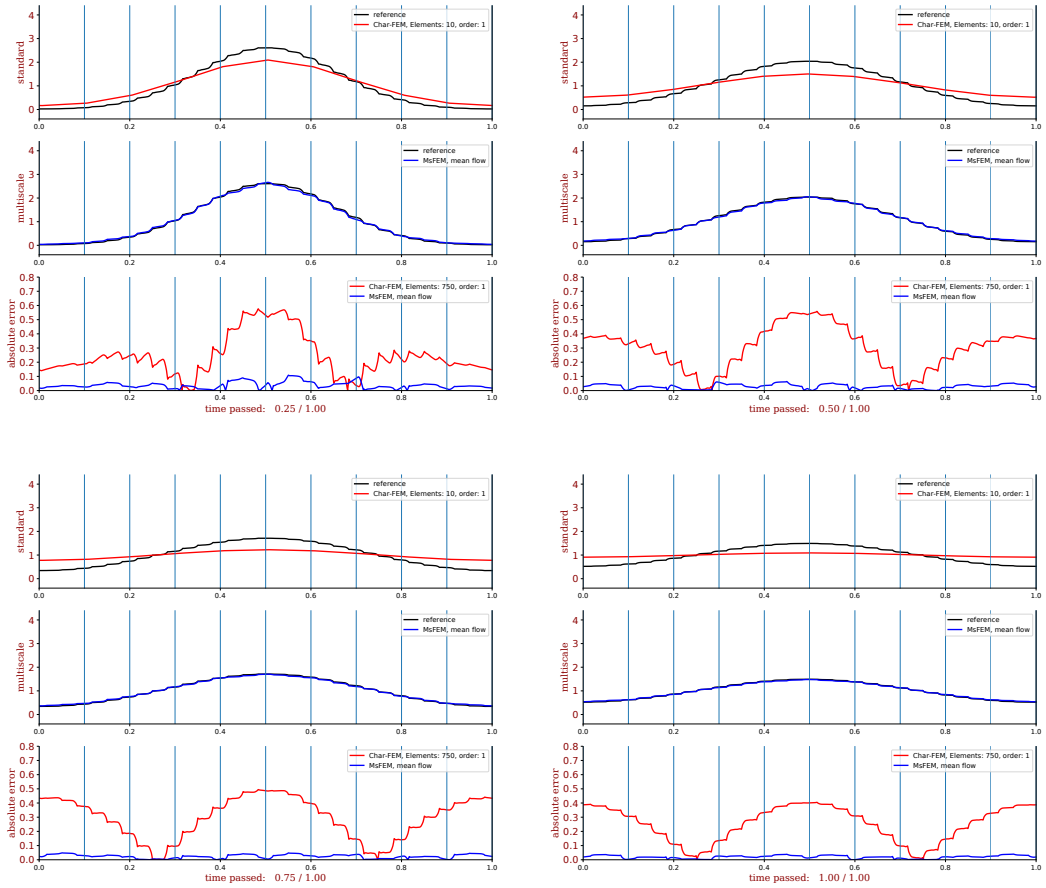


Figure 5: Snapshots of the solution of case 1 with data given by (42) at time stamps $t = 0.25$, $t = 0.5$, $= 0.75$ and $T = 1$ for $k = 30$. The upper graphs show a low-resolution standard FEM (red) compared to a high-resolution standard FEM reference solution (black). The second row of each snapshot shows our MsFEM (blue) compared to the reference solution. The third row of each snapshot shows the absolute difference of the low-resolution FEM solution (red) and of the MsFEM solution (blue) to the reference solution. The absolute error of the MsFEM is clearly lower and smoother that the error of the standard FEM.

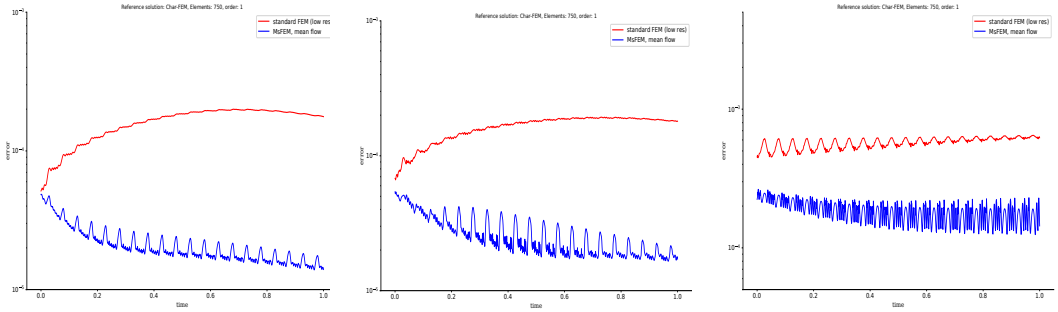
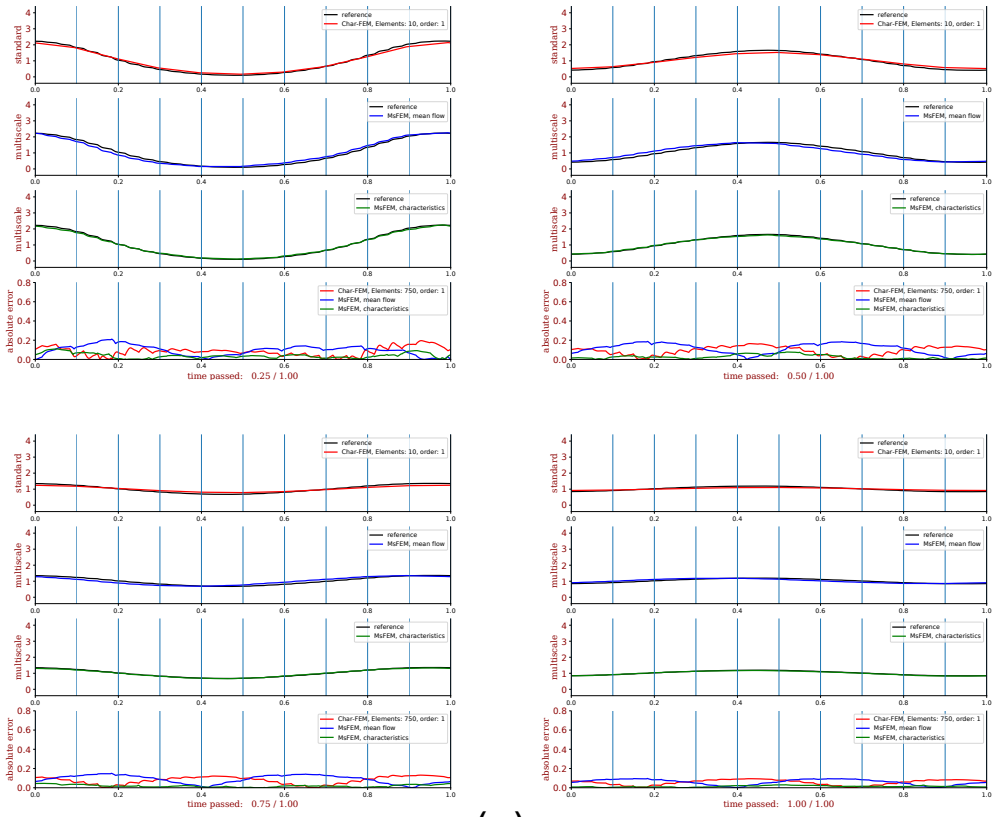
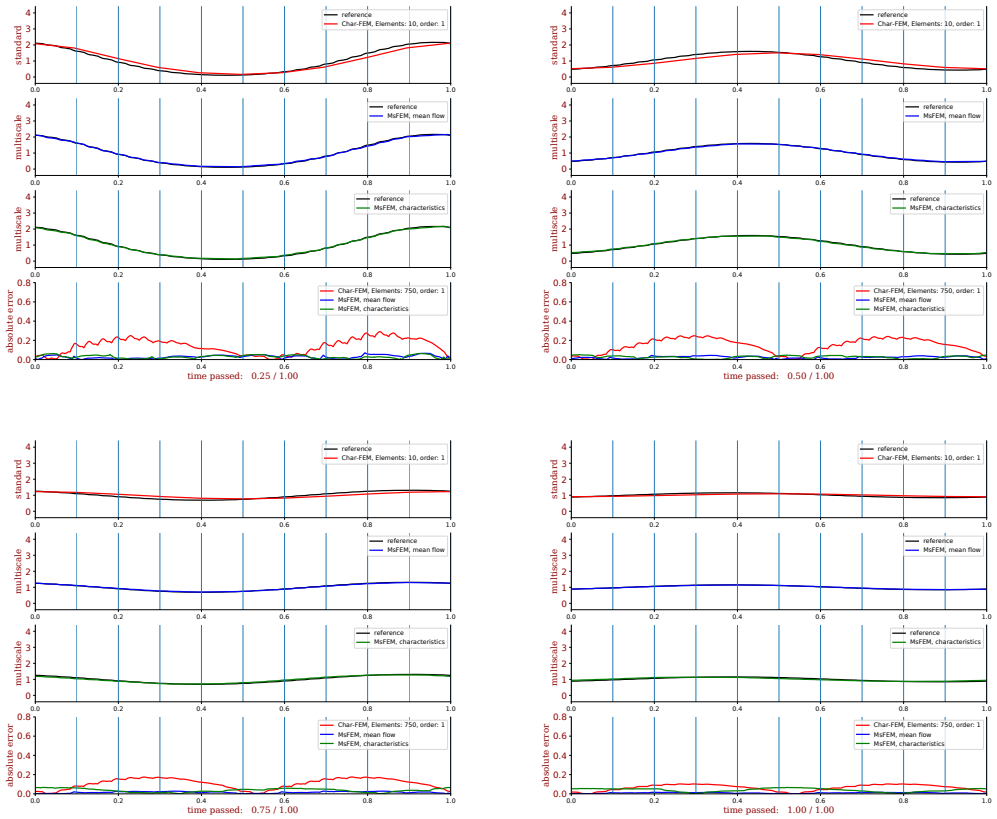


Figure 6: Errors of the solution of case 1 with data given by (42) for $k = 30$. Errors of a low-resolution standard FEM (red) are compared to errors of our MsFEM (blue) with respect to a high-resolution reference solution. The L^2-error and the L^∞-error of the MsFEM are approximately one order of magnitude better than the errors of the standard FEM. The error of the derivative of the MsFEM is approximately half an order of magnitude better than the error of the standard FEM.



(a)



(b)

Figure 7: Comparison of snapshots of the solution of case 2 with data given by (43) at time stamps $t = 0.25$, $t = 0.5$, $= 0.75$ and $T = 1$ for **(a)** $k = 3$ and **(b)** $k = 60$. In each image the first row represents a comparison of either a standard FEM (red), the MF-MsFEM (blue) or a Char-MsFEM (green) to a reference solution (black). The respective absolute errors are shown in the last row of each image.

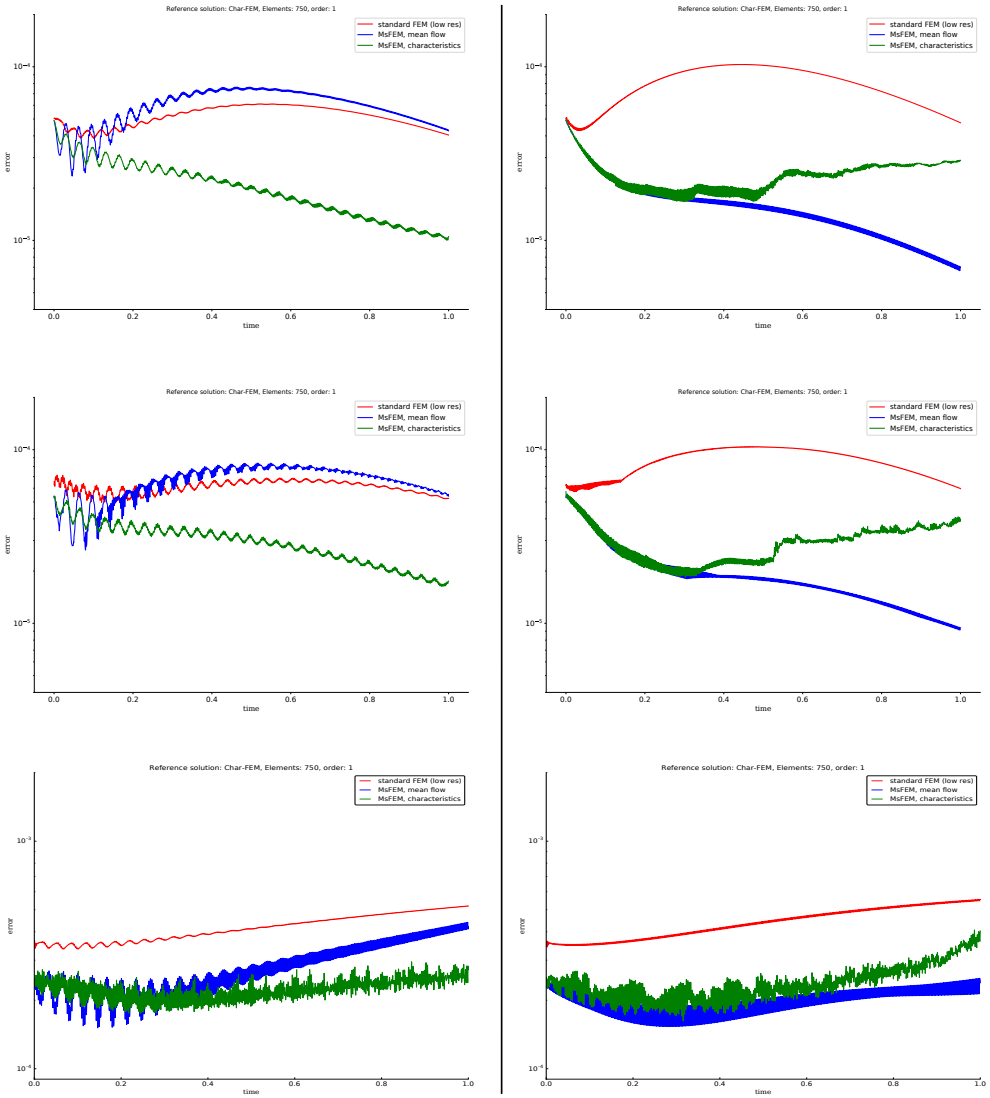
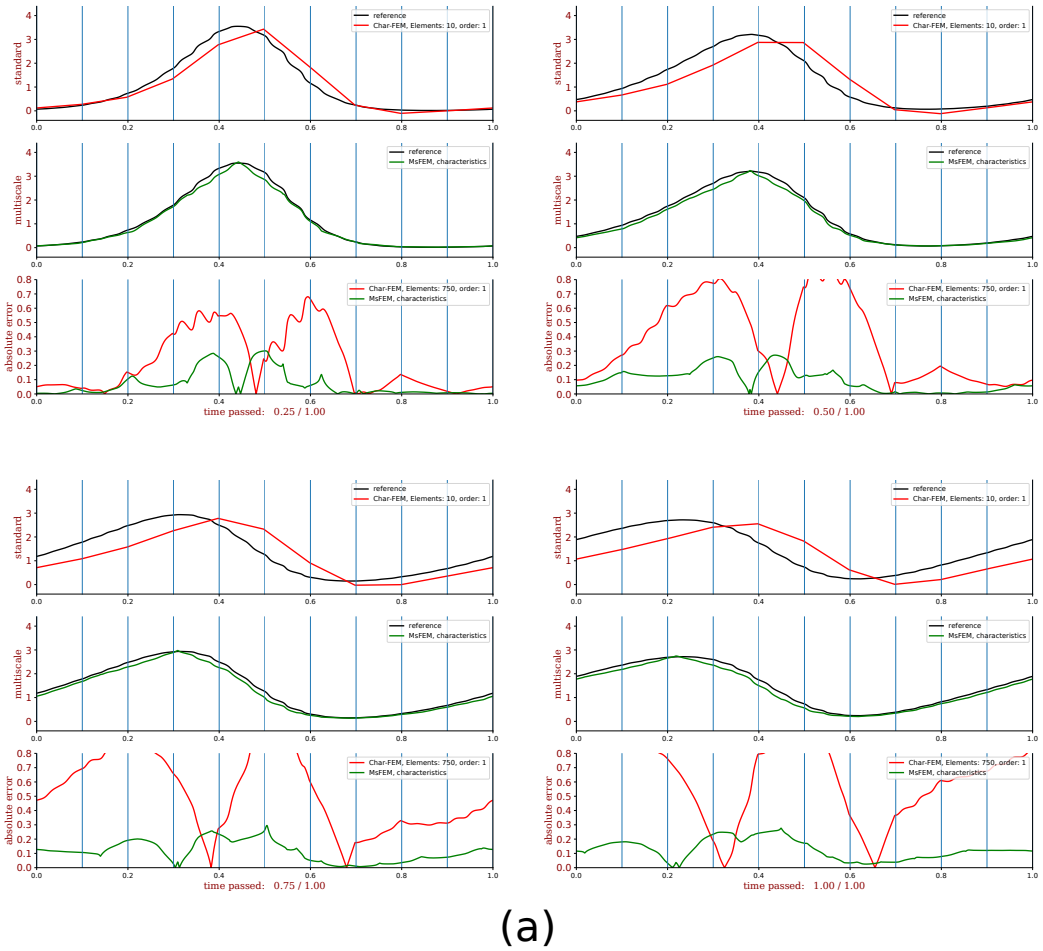
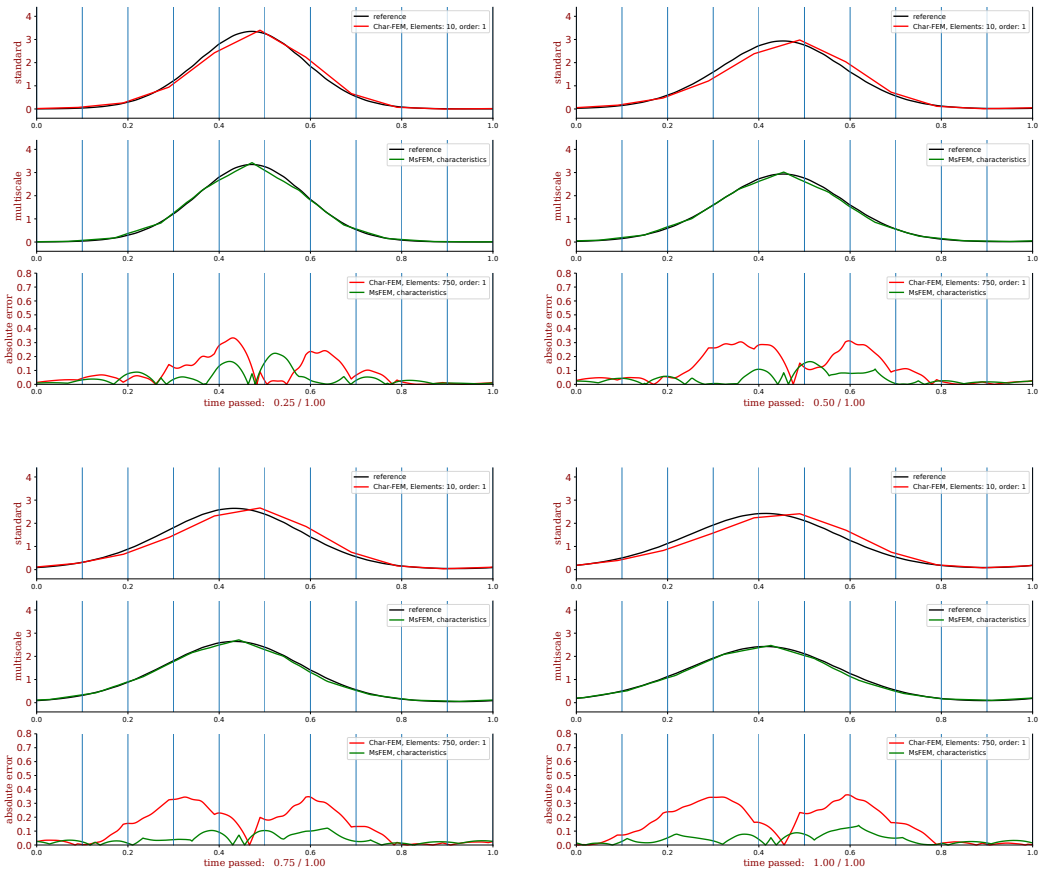


Figure 8: Errors of the solution of case 2 with data given by (43) for **(a)** $k = 3$ and **(b)** $k = 60$.



(a)



(b)

Figure 9: Comparison of snapshots of the solution of case 3 with data given by (44) at time stamps $t = 0.25$, $t = 0.5$, $= 0.75$ and $T = 1$ for **(a)** $v = 4$ and **(b)** $v = 16$. Each image contains a comparison of a standard FEM (red) and the Char-MsFEM (green) to a reference solution (black). The absolute errors with respective colors are shown in the last row of each image.

Case 4. This test is to demonstrate consistency of the Char-MsFEM in the limit of a large scale separation. This means that we will demonstrate that our method yields a solution that correctly follows the reference solution as the diffusion coefficient becomes more and more oscillatory. It is not our aim to find the limiting equation, i.e., a homogenized equation, and to demonstrate that our numerical solution enjoys the correct behavior with respect to the solution of the homogenized equation. Finding the correct homogenized equation is often a difficult task, even in simple settings such as purely diffusive ones. Additionally to the oscillating diffusive term we will have a background velocity that is resolved by the coarse grid since our motivation comes from climate simulations. There the velocity data is usually taken from a coarse grid while the fine scale part is hidden in the parametrizations. The coefficients that we use for this case are

$$\begin{aligned} c(x,t) &= (2t+0.5)(3+\cos(2\pi x)+\cos(60\pi x))\,, \\ \mu(x,t) &= 0.01+0.0099\cos(2k\pi x)\,. \end{aligned} \tag{45}$$

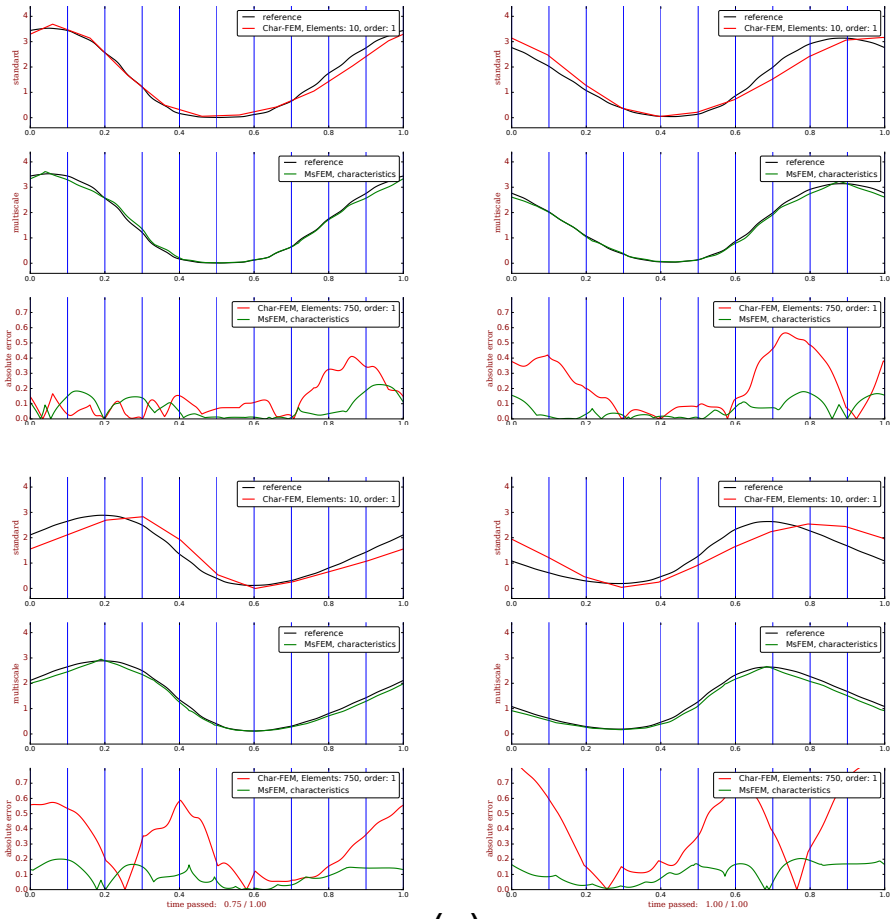
Note that the standard solution also runs faster than the Char-MsFEM. This effect is much more obvious than the wrong representation of the fine scale diffusion on the coarse grid, see Figure 10. Similarly to case 1 but less obvious the standard solution does not capture the oscillations in the diffusive term well. This effect is less severe since the velocity coefficient "wipes" these oscillations out. Table 4 shows the relative deviation of the maximum of the coarse standard solution and of the Char-MsFEM with respect to the reference solution. Errors in L^2 and L^∞ are summarized in Table 5.

	FEM	Char-MsFEM
$k=15$	$3.66\cdot10^{-2}$	$8.94\cdot10^{-3}$
$k=30$	$5.67\cdot10^{-2}$	$9.63\cdot10^{-3}$
$k=60$	$7.80\cdot10^{-2}$	$2.43\cdot10^{-2}$
$k=90$	$8.52\cdot10^{-2}$	$3.08\cdot10^{-2}$
$k=140$	$1.09\cdot10^{-1}$	$1.50\cdot10^{-2}$
$k=200$	$1.47\cdot10^{-1}$	$1.74\cdot10^{-2}$

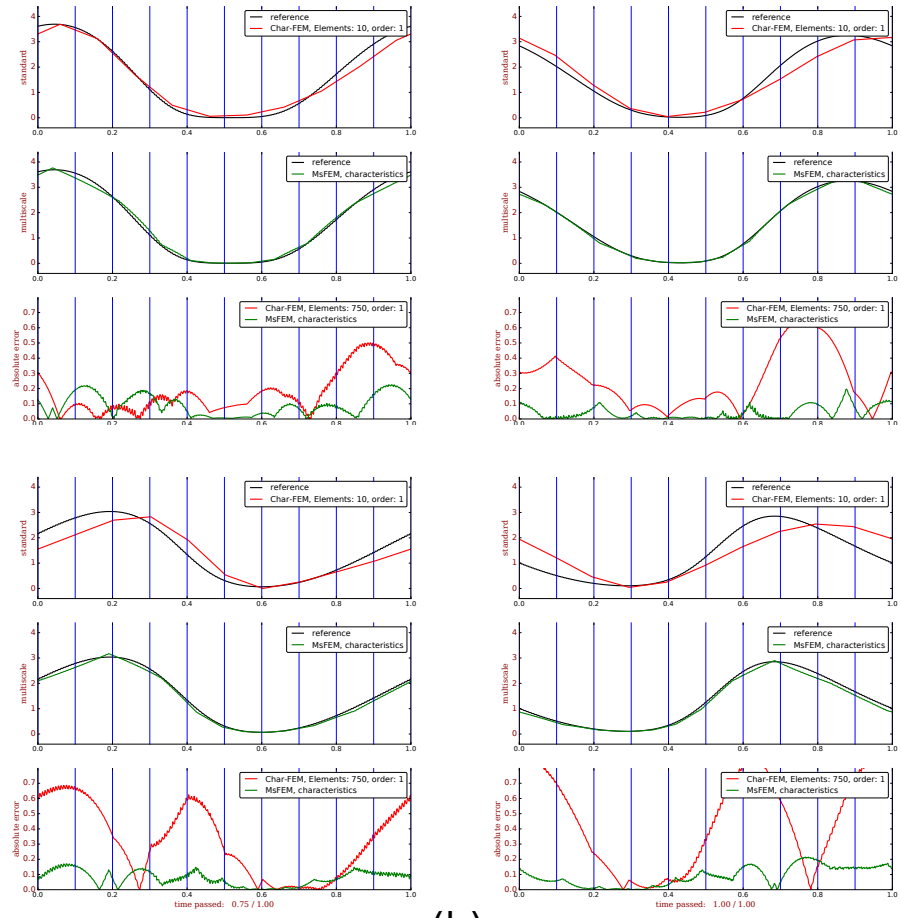
Table 4: Relative deviation of the maximum of case 4 at $T=1$. Coefficients are given by (45).

	$L^2(I)$		$L^\infty(I)$	
	FEM	Char-MsFEM	FEM	Char-MsFEM
$k=15$	$2.137\cdot10^{-4}$	$5.418\cdot10^{-5}$	$2.204\cdot10^{-4}$	$5.168\cdot10^{-5}$
$k=30$	$2.173\cdot10^{-4}$	$2.233\cdot10^{-5}$	$2.196\cdot10^{-4}$	$5.142\cdot10^{-5}$
$k=60$	$2.220\cdot10^{-4}$	$4.898\cdot10^{-5}$	$2.189\cdot10^{-4}$	$4.731\cdot10^{-5}$
$k=90$	$2.240\cdot10^{-4}$	$4.585\cdot10^{-5}$	$2.203\cdot10^{-4}$	$4.755\cdot10^{-5}$
$k=140$	$2.308\cdot10^{-4}$	$4.481\cdot10^{-5}$	$2.211\cdot10^{-4}$	$5.028\cdot10^{-5}$
$k=200$	$2.420\cdot10^{-4}$	$4.933\cdot10^{-5}$	$2.215\cdot10^{-4}$	$6.083\cdot10^{-5}$

Table 5: Relative error of case 4 at $T=1$. Coefficients are given by (45).



(a)



(b)

Figure 10: Comparison of snapshots of the solution of case 4 with data given by (45) at time stamps $t = 0.25$, $t = 0.5$, $= 0.75$ and $T = 1$ for **(a)** $k = 15$ and **(b)** $k = 140$. Each image contains a comparison of a standard FEM (red) and the Char-MsFEM (green) to a reference solution (black). The absolute errors with respective colors are shown in the last row of each image.

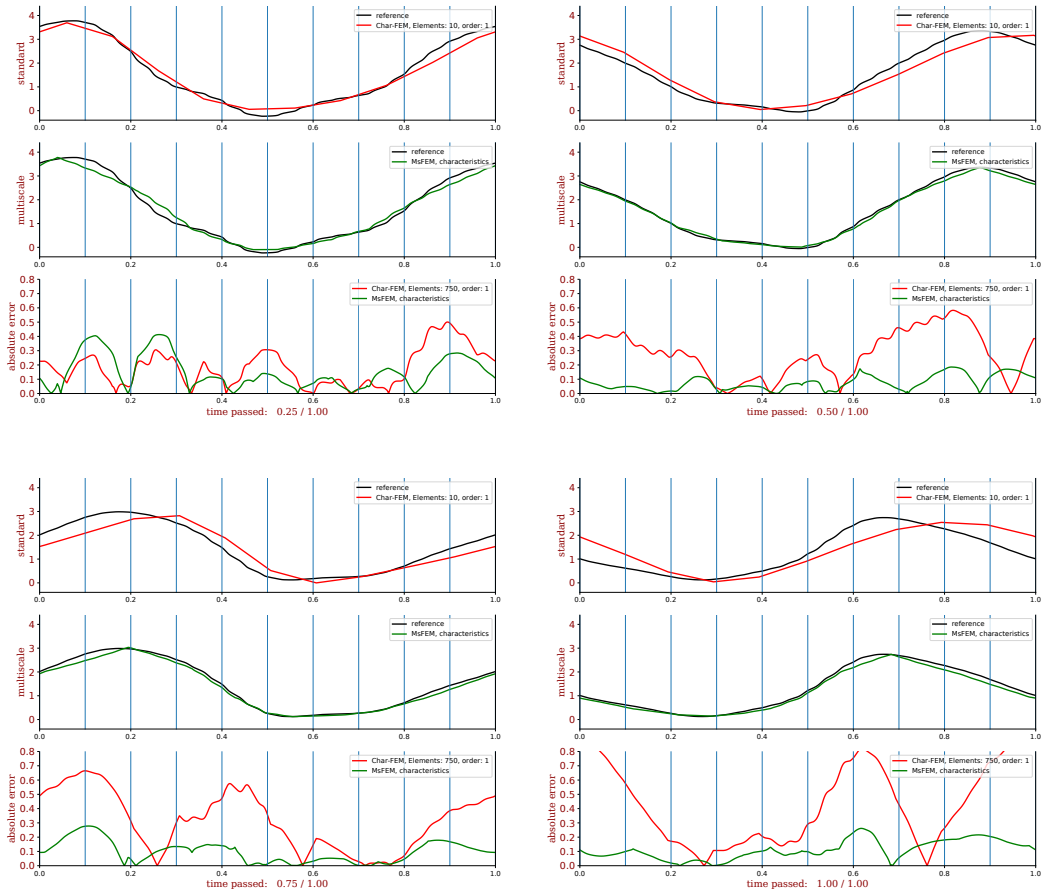


Figure 11: Comparison of snapshots of the solution of case 5 with data given by (46) at time stamps $t = 0.25$, $t = 0.5$, $= 0.75$ and $T = 1$. Each image contains a comparison of a standard FEM (red) and the Char-MsFEM (green) to a reference solution (black). The absolute errors with respective colors are shown in the last row of each image.

Case 5. We use a simple example to demonstrate that the Char-MsFEM also works with non-vanishing external forcing. The coefficients we choose are given by

$$\begin{aligned} g(x) =& 0.015 \sin(8\pi) \\ c(x,t) =& (2t + 0.5)(3 + \cos(2\pi x) + \cos(60\pi x)) \,, \\ \mu(x,t) =& \, 0.01 + 0.0099 \cos(2k\pi x) \,. \end{aligned} \tag{46}$$

Snapshots are shown in Figure 11. Remarkably, the Char-MsFEM also yields good results if we choose the above forcing to act on a scale much finer than resolved by the coarse grid. The resulting multiscale solution then of course does not capture the fine scale forcing since the basis functions do not (although they capture the fine scale diffusion) but we observed in tests that it still runs correctly with the reference solution in contrast to the solution of the standard FEM. However, we do not strive to investigate external forcing on subgrid scales and therefore pass on showing this.

A note on convergence. In Case 4 we demonstrated that the Char-MsFEM correctly captures fine scale features of the solution as the scale separation in the diffusion coefficient grows. Here we show indications for convergence of the Char-MsFEM as the coarse resolution $H \to 0$. The classical multiscale finite element method for elliptic problems as described in [17, 18] is shown to behave like a standard FEM as $H \to 0$. Here we see a similar behavior but the Char-MsFEM convergences slowly. The data for the tests is given by

$$\begin{aligned} c(x,t) =& (2t + 0.5)(1.5 + \frac{\alpha}{2} \cos(2\pi x)) \,, \\ \mu(x,t) =& \, 0.01 + 0.0099 \cos(2k\pi x) \,. \end{aligned} \tag{47}$$

The right-hand side g is taken to be zero. Snapshots for the cases $\alpha = 0$ and $\alpha = 1$ for $k = 10$ are shown for different coarse resolutions in Figure 13 as well as the respective mean square and maximum errors in Figure 12. The influence of the smoothness of the diffusion coefficient on the error taken with respect to a high-resolution reference solution is summarized in Table 6 for spatially constant background velocity as well as for spatially varying velocity. The tests indicate that the error is smaller the smoother the diffusion coefficient is relative to the coarse grid, i.e., the method converges faster for smooth diffusion. Clearly visible from Figure 13 is that once the coefficients are resolved by the coarse grid the Char-MsFEM solution is slower than the reference solution but this effect vanishes as H decreases. The time step was taken as $\delta t = 1/1000$ and the number of fine elements in each coarse block was taken to be constant for all different H. However, note that this regime is practically not relevant since here the standard FEM is sufficient if the data is resolved by the coarse grid. The advantage of the Char-MsFEM is at scales when data is not resolved by the coarse grid.

4 Discussion

We have suggested a flexible numerical multiscale method based on a finite element formulation that uses a set of non-polynomial basis functions to solve an advection-diffusion equation with a smooth, well-behaved and dominant background velocity and a heterogeneous diffusive term. The model problem is motivated by problems that arise in long-term climate simulations where prognostic variables like the background velocity live on a coarse grid while parametrized subgrid processes are modelled in the diffusive part.

The numerical method is motivated by a classical MsFEM suggested by the porous media community. We substantially modified the ideas to account for scenarios relevant to climate simulations. This modification is based on transforms suggested by the background velocity. In particular these transforms aim to make the dominant advective term milder. Our suggested multiscale methods are not overly complicated and consist of an offline phase in which many small local problems for the basis functions (with Dirichlet boundary conditions on curves suggested by the transform) are solved and an online phase that uses the basis in a finite element like framework in the global formulation. The offline phase is suitable for massive parallelization where as the online phase is fast. Also the global form is not restricted to use a finite element framework. It can potentially be used with finite volumes or discontinuous Galerkin formulations.

In tests we showed an improvement of numerical solutions using our new technique compared to standard solutions on the same coarse grid. Since no analytical solutions are available we compare our solutions to high-resolution standard FEM solutions. Tests indicate that our multiscale solutions enjoy the correct upscaling behavior in the limit of fast oscillations in the diffusion, i.e., the faster the diffusion oscillates relative to the coarse grid the more advantageous are our multiscale methods over standard methods.

Our multiscale techniques show good results but unfortunately their scope is limited to velocity fields that have a dominant mean value for all times. This has to do with the dynamics of the different characteristics. The less dominant the mean of the background velocity the closer characteristics emerging at coarse grid boundaries can come in short time. In particular time-independent zeros in the background velocity represent stable and unstable attractors of the advective part. Such features can make coarse blocks collapse and make the method unstable.

Furthermore, a direct generalization to higher dimensions is quite difficult. This is due to richer dynamics in the characteristics even if the velocity field is solenoidal. Also, our methods fail if the advection-diffusion equation is in conservation form. The reason is that a conservative transport term of the form $(cu)_x$ decomposes into an advective part cu_x and a reactive part $c_x u$ that compensates for non-divergence of the background velocity c. Such reaction terms we found in tests enjoy a different upscaling mechanism and can not be treated with our method.

However, although our method does not work in all cases we showed that it works in relevant scenarios and identified difficulties in their generalization. Being aware of such difficulties can facilitate the choice of modified ingredients to account for generalizations and for the development

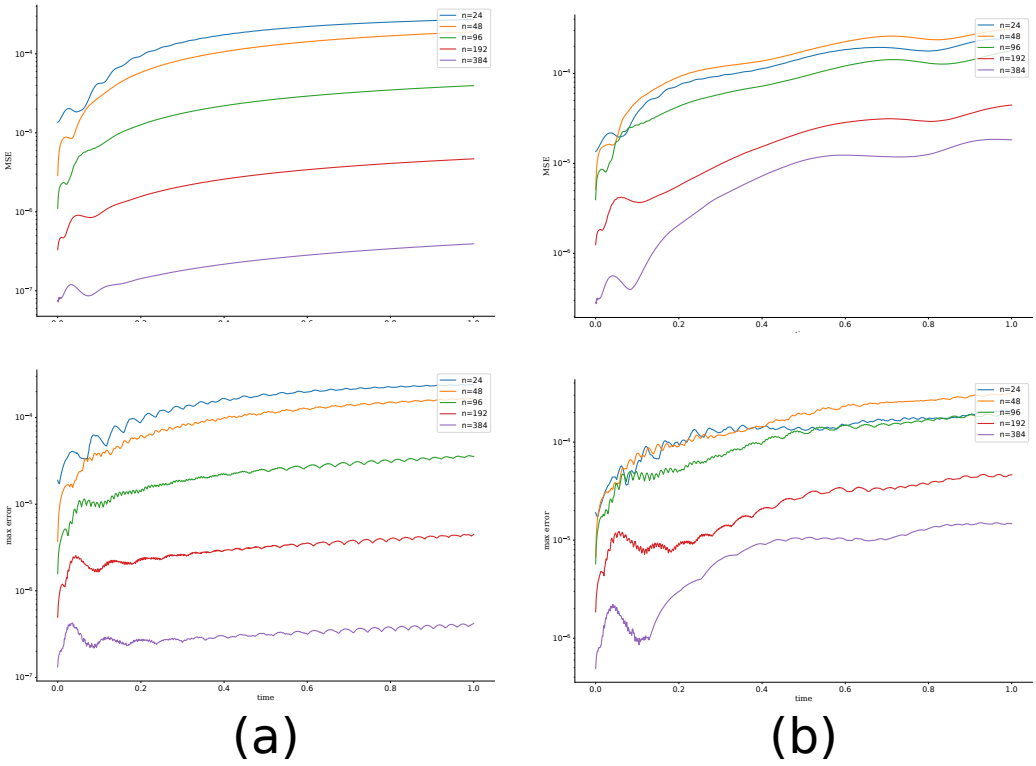


Figure 12: Relative errors of the solution with data given by (47) for $k = 10$ over time for **(a)** $\alpha = 0$ and **(b)** $\alpha = 1$.

k	N	$\|e\|_{L^2}$	$\|e\|_{L^\infty}$
10	24	$2.732 \cdot 10^{-4}$	$2.435 \cdot 10^{-4}$
	48	$1.898 \cdot 10^{-4}$	$1.692 \cdot 10^{-4}$
	96	$3.963 \cdot 10^{-5}$	$3.569 \cdot 10^{-5}$
	192	$4.688 \cdot 10^{-6}$	$4.474 \cdot 10^{-6}$
	384	$3.939 \cdot 10^{-7}$	$4.204 \cdot 10^{-7}$
15	24	$9.603 \cdot 10^{-5}$	$9.129 \cdot 10^{-5}$
	48	$3.400 \cdot 10^{-4}$	$2.911 \cdot 10^{-4}$
	96	$1.141 \cdot 10^{-4}$	$9.920 \cdot 10^{-5}$
	192	$1.731 \cdot 10^{-5}$	$1.566 \cdot 10^{-5}$
	384	$1.455 \cdot 10^{-6}$	$1.368 \cdot 10^{-6}$
20	24	$2.629 \cdot 10^{-5}$	$2.730 \cdot 10^{-5}$
	48	$3.819 \cdot 10^{-4}$	$3.250 \cdot 10^{-4}$
	96	$1.986 \cdot 10^{-4}$	$1.686 \cdot 10^{-4}$
	192	$4.110 \cdot 10^{-5}$	$3.616 \cdot 10^{-5}$
	384	$4.104 \cdot 10^{-6}$	$3.702 \cdot 10^{-6}$

(a) $\alpha = 0$

k	N	$\|e\|_{L^2}$	$\|e\|_{L^\infty}$
10	24	$2.409 \cdot 10^{-4}$	$2.105 \cdot 10^{-4}$
	48	$3.140 \cdot 10^{-4}$	$3.096 \cdot 10^{-4}$
	96	$1.783 \cdot 10^{-4}$	$1.888 \cdot 10^{-4}$
	192	$4.447 \cdot 10^{-5}$	$4.640 \cdot 10^{-5}$
	384	$1.828 \cdot 10^{-5}$	$1.465 \cdot 10^{-5}$
15	24	$1.647 \cdot 10^{-4}$	$1.335 \cdot 10^{-4}$
	48	$5.162 \cdot 10^{-4}$	$4.487 \cdot 10^{-4}$
	96	$4.159 \cdot 10^{-4}$	$4.128 \cdot 10^{-4}$
	192	$9.032 \cdot 10^{-5}$	$9.610 \cdot 10^{-5}$
	384	$2.001 \cdot 10^{-5}$	$1.965 \cdot 10^{-5}$
20	24	$7.963 \cdot 10^{-5}$	$6.090 \cdot 10^{-5}$
	48	$6.185 \cdot 10^{-4}$	$5.368 \cdot 10^{-4}$
	96	$6.595 \cdot 10^{-4}$	$6.122 \cdot 10^{-4}$
	192	$1.817 \cdot 10^{-4}$	$1.847 \cdot 10^{-4}$
	384	$2.930 \cdot 10^{-5}$	$3.160 \cdot 10^{-5}$

(b) $\alpha = 1$

Table 6: Relative errors of the numerical solution for data given by (47) for **(a)** $\alpha = 0$ and **(b)** $\alpha = 1$ as H becomes small, i.e., N becomes large. The table shows the influence of increasing frequency of the diffusion coefficient.

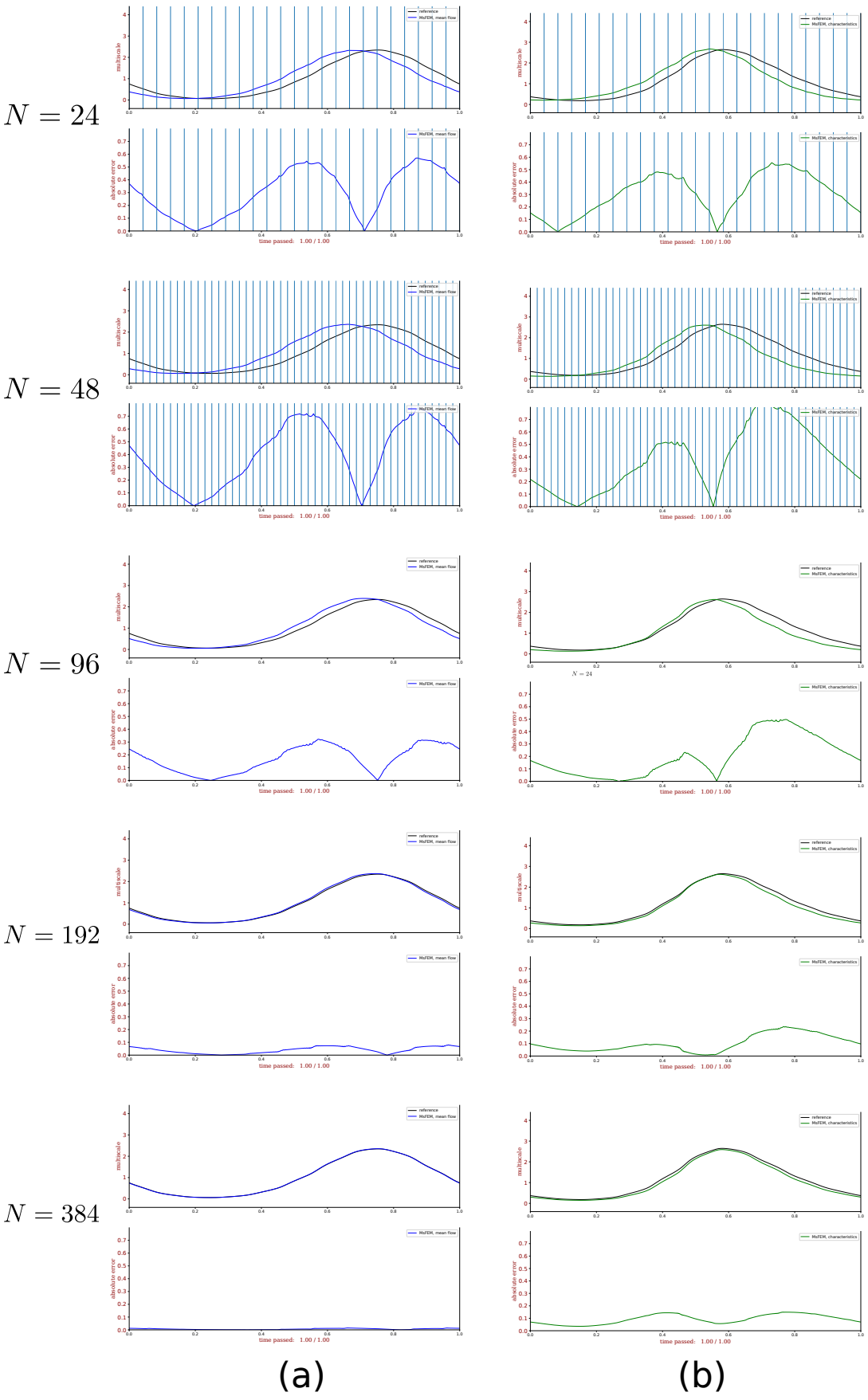


Figure 13: Comparison of snapshots of the solution with data given by (47) at time $T = 1$ and $k = 10$ for **(a)** $\alpha = 0$ and **(b)** $\alpha = 1$. The multiscale solution is compared to a reference solution (black). The absolute errors are shown in the second row of each image.

of suitable multiscale methods. For example, the reader may think of a combination of semi-Lagrangian methods to deal with the problem of converging or crossing characteristics and our multiscale methods. This is left for future research. With this work we hope to spark a debate on Galerkin based multiscale methods in the climate simulation community and to contribute to their success.

Acknowledgements. This work was supported by German Federal Ministry of Education and Research (BMBF) as Research for Sustainability initiative (FONA); *www.fona.de* through Palmod project (FKZ: 01LP1513A).

References

[1] Assyr Abdulle, Weinan E, Björn Engquist, and Eric Vanden-Eijnden. The heterogeneous multiscale method. *Acta Numerica*, 21:1–87, 2012.

[2] Todd Arbogast. Numerical subgrid upscaling of two-phase flow in porous media. *LECTURE NOTES IN PHYSICS-NEW YORK THEN BERLIN-*, pages 35–49, 2000.

[3] Todd Arbogast and Kirsten J Boyd. Subgrid upscaling and mixed multiscale finite elements. *SIAM Journal on Numerical Analysis*, 44(3):1150–1171, 2006.

[4] Ivo Babuška, Gabriel Caloz, and John E Osborn. Special finite element methods for a class of second order elliptic problems with rough coefficients. *SIAM Journal on Numerical Analysis*, 31(4):945–981, 1994.

[5] Ivo Babuška and John E Osborn. Generalized finite element methods: their performance and their relation to mixed methods. *SIAM Journal on Numerical Analysis*, 20(3):510–536, 1983.

[6] Jörn Behrens. *Adaptive Atmospheric Modeling*. Springer, 2006.

[7] Alain Bensoussan, Jacques-Louis Lions, and George Papanicolaou. *Asymptotic analysis for periodic structures*, volume 374. American Mathematical Soc., 2011.

[8] Franco Brezzi. Interacting with the subgrid world. *CHAPMAN AND HALL CRC RESEARCH NOTES IN MATHEMATICS*, pages 69–82, 2000.

[9] Michael A Celia, Thomas F Russell, Ismael Herrera, and Richard E Ewing. An eulerian-lagrangian localized adjoint method for the advection-diffusion equation. *Advances in Water Resources*, 13(4):187–206, 1990.

[10] Aijie Cheng, Kaixin Wang, and Hong Wang. A preliminary study on multiscale ellam schemes for transient advection-diffusion equations. *Numerical Methods for Partial Differential Equations*, 26(6):1405–1419, 2010.

[11] Weinan E. *Principles of multiscale modeling*. Cambridge University Press, 2011.

[12] Weinan E, Bjorn Engquist, et al. The heterognous multiscale methods. *Communications in Mathematical Sciences*, 1(1):87–132, 2003.

[13] Yalchin Efendiev and Thomas Y Hou. *Multiscale finite element methods: theory and applications*, volume 4. Springer Science & Business Media, 2009.

[14] Lawrence C. Evans. *Partial differential equations*. American Mathematical Society, 2010.

[15] Ivan G Graham, Thomas Y Hou, Omar Lakkis, and Robert Scheichl. *Numerical analysis of multiscale problems*, volume 83. Springer Science & Business Media, 2012.

[16] Ismael Herrera, Richard E Ewing, Michael A Celia, and Thomas F Russell. Eulerian-lagrangian localized adjoint method: The theoretical framework. *Numerical Methods for Partial Differential Equations*, 9(4):431–457, 1993.

[17] Thomas Y Hou and Xiao-Hui Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *Journal of computational physics*, 134(1):169–189, 1997.

[18] Thomas Y Hou, Xiao-Hui Wu, and Zhiqiang Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Mathematics of Computation of the American Mathematical Society*, 68(227):913–943, 1999.

[19] Thomas JR Hughes. Multiscale phenomena: Green's functions, the dirichlet-to-neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer methods in applied mechanics and engineering*, 127(1-4):387–401, 1995.

[20] Thomas JR Hughes, Gonzalo R Feijóo, Luca Mazzei, and Jean-Baptiste Quincy. The variational multiscale method – a paradigm for computational mechanics. *Computer methods in applied mechanics and engineering*, 166(1-2):3–24, 1998.

[21] Christiane Jablonowski. *Adaptive grids in weather and climate modeling.* PhD thesis, University of Michigan, 2004.

[22] Vasilii Vasil'evich Jikov, Sergei M Kozlov, and Olga Arsen'evna Oleinik. *Homogenization of differential operators and integral functionals.* Springer Science & Business Media, 2012.

[23] Grigoris Pavliotis and Andrew Stuart. *Multiscale methods: averaging and homogenization.* Springer Science & Business Media, 2008.

[24] Thomas F Russell and Michael A Celia. An overview of research on eulerian–lagrangian localized adjoint methods (ellam). *Advances in Water Resources*, 25(8):1215–1231, 2002.

[25] Hong Wang, Yabin Ding, Kaixin Wang, Richard E Ewing, and Yalchin R Efendiev. A multiscale eulerian–lagrangian localized adjoint method for transient advection–diffusion equations with oscillatory coefficients. *Computing and Visualization in Science*, 12(2):63–70, 2009.