Adaptive multiscale model reduction for nonlinear parabolic equations using GMsFEM

Yiran Wang, Eric Chung and Shubin Fu

Department of Mathematics, The Chinese University of Hong Kong, Hong Kong SAR

Abstract. In this paper, we propose a coupled Discrete Empirical Interpolation Method (DEIM) and Generalized Multiscale Finite element method (GMsFEM) to solve nonlinear parabolic equations with application to the Allen-Cahn equation. The Allen-Cahn equation is a model for nonlinear reaction-diffusion process. It is often used to model interface motion in time, e.g. phase separation in alloys. The GMsFEM allows solving multiscale problems at a reduced computational cost by constructing a reduced-order representation of the solution on a coarse grid. In [15], it was shown that the GMsFEM provides a flexible tool to solve multiscale problems by constructing appropriate snapshot, offline and online spaces. In this paper, we solve a time dependent problem, where online enrichment is used. The main contribution is comparing different online enrichment methods. More specifically, we compare uniform online enrichment and adaptive methods. We also compare two kinds of adaptive methods. Furthermore, we use DEIM, a dimension reduction method to reduce the complexity when we evaluate the nonlinear terms. Our results show that DEIM can approximate the nonlinear term without significantly increasing the error. Finally, we apply our proposed method to the Allen Cahn equation.

Keywords: online adaptive model reduction · Discrete Empirical Interpolation Method · flows in heterogeneous media· Exponential Time Differencing

1 Introduction

In this paper, we consider the Generalized Multiscale Finite element method (GMsFEM) for solving nonlinear parabolic equations. The main objectives of the paper are the following: (1) to demonstrate the main concepts of GMsFEM and brief review of the techniques; (2) to compare various online enrichment techniques; (3) to discuss the use of the Discrete Empirical Interpolation Method (DEIM) and present its performance in reducing complexity. GMsFEM is a flexible general framework that generalizes the Multiscale Finite Element Method (MsFEM) by systematically enriching the coarse spaces. The main idea of this enrichment is to add extra basis functions that are needed to reduce the error substantially. Once the offline space is derived, it stays fixed and unchanged in the online stage. In [3], it is shown that a good approximation from the reduced model can be expected only if the offline information is a good representation of the problem. For time dependent problems, online enrichment is necessary. We compare two kinds of online enrichment methods: uniform and adaptive enrichment, where the latter focuses on where to add online basis. We will discuss it in numerical results with more details. When a general nonlinearity is present, the cost to evaluate the projected nonlinear function still depends on the dimension of the original system, resulting in simulation times that can hardly improve over the original system. One approach to reduce computational cost is the POD-Galerkin method [5,6,7,8], which is applied to many applications, for example, in [9,10,11,12,13]. DEIM focuses on approximating each nonlinear function so that a certain coefficient matrix can be precomputed and, as a result, the complexity in evaluating the nonlinear term becomes proportional to the small number of selected spatial indices. In this paper, we will compare various approximations of the DEIM projection. We will illustrate these concepts by applying our proposed method to the Allen Cahn equation. The remainder of the paper is organized as follows. In section 2, we present the problem setting and main ingredients of GMsFEM. In section 3, we consider the methods to solve the Allen-Cahn equation.

2 Multiscale model reduction using the GMsFEM

In this section, we will give the construction of our GMsFEM for nonlinear parabolic equations. First, we present some basic notations and the coarse grid formulation in Section 2.1. Then, we present the

construction of the multiscale snapshot functions and basis functions in Section 2.2. The online enrichment process is introduced in Section 2.3.

2.1 Preliminaries

Consider the following parabolic equation in the domain $\Omega \subset \mathbb{R}^d$

$$
\begin{aligned}
\frac{\partial u}{\partial t} - \operatorname{div}(\kappa \nabla u) &= f & &\text{in } \Omega \times [0, T], \\
u(x, 0) &= g(x) & &\text{in } \Omega, \\
u(x, t) &= 0 & &\text{on } \partial \Omega \times [0, T].
\end{aligned}
\tag{1}
$$

Here, we denote the exact solution of (1) by u, $\kappa(x)$ is a high-contrast and heterogeneous permeability field, $f = f(x, u)$ is the nonlinear source function depending on the u variable, $g(x)$ is a given function and $T > 0$ is the final time. We denote the solution and the source term at $t = t_n$ by $u(\cdot, t_n)$ and $f(u(\cdot, t_n))$ respectively. The variational formulation for the problem (1) is: find $u(\cdot, t) \in H_0^1(\Omega)$ such that

$$
\begin{aligned}
\left\langle \frac{\partial u}{\partial t}, v \right\rangle + \mathcal{A}(u, v) &= \langle f, v \rangle & &\text{in } \Omega \times [0, T], \quad \forall v \in H_0^1(\Omega), \\
u(x, 0) &= g(x) \quad \text{in } \Omega, \\
u(x, t) &= 0 \quad \text{on } \partial \Omega \times [0, T].
\end{aligned}
\tag{2}
$$

where $\mathcal{A}(u, v) = \int_\Omega \kappa \nabla u \cdot \nabla v \, dx$.

In order to discretize (2) in time, we need to apply some time differencing methods. For simplicity, we first apply the implicit Euler scheme with time step $\Delta t > 0$ and in section 3, we will consider a different differencing method: ETD. We obtain the following discretization for each time $t_n = n\Delta t, n = 1, 2, \cdots, N$ ($T = N\Delta t$),

$$
\frac{u(\cdot, t_n) - u(\cdot, t_{n-1})}{\Delta t} = \operatorname{div}(\kappa \nabla u(\cdot, t_n)) + f(u(\cdot, t_n)).
$$

Let T^h be a partition of the domain Ω into fine finite elements. Here $h > 0$ is the fine grid mesh size. The coarse partition, T^H of the domain Ω, is formed such that each element in T^H is a connected union of fine-grid blocks. More precisely, $\forall K_j \in T^H$, $K_j = \bigcup_{F \in I_j} F$ for some $I_j \subset T^h$. The quantity $H > 0$ is the coarse mesh size. We will consider the rectangular coarse elements and the methodology can be used with general coarse elements. An illustration of the mesh notations is shown in the Figure 1. We denote the interior nodes of T^H by $x_i, i = 1, \cdots, N_{\text{in}}$, where N_{in} is the number of interior nodes. The coarse elements of T^H are denoted by $K_j, j = 1, 2, \cdots, N_e$, where N_e is the number of coarse elements. We define the coarse neighborhood of the nodes x_i by $D_i := \cup \{K_j \in T_H : x_i \in \overline{K_j}\}$.

2.2 The GMsFEM and the multiscale basis functions

In this paper, we will apply the GMsFEM to solve nonlinear parabolic equations. The method is motivated by the finite element framework. First, a variational formulation is defined. Then we construct some multiscale basis functions. Once the fine grid is given, we can compute the fine-grid solution. Let $\gamma_1, \cdots, \gamma_n$ be the standard finite element basis, and define $V_f = \operatorname{span}\{\gamma_1, \cdots, \gamma_n\}$ to be the fine space. We obtained the fine solution denoted by u_f^n at $t = t_n$ by solving

$$
\begin{aligned}
\frac{1}{\Delta t}\langle u_f^n, v \rangle + \mathcal{A}\left(u_f^n, v\right) &= \left\langle \frac{1}{\Delta t} u_f^{n-1} + f(u_f^n), v \right\rangle, \quad \forall v \in V_f, \\
u_f^0 &= g_h,
\end{aligned}
\tag{3}
$$

where g_h is the V_f based approximation of g. The construction of multiscale basis functions follows two general steps. First, we construct snapshot basis functions in order to build a set of possible modes of the solutions. In the second step, we construct multiscale basis functions with a suitable spectral problem
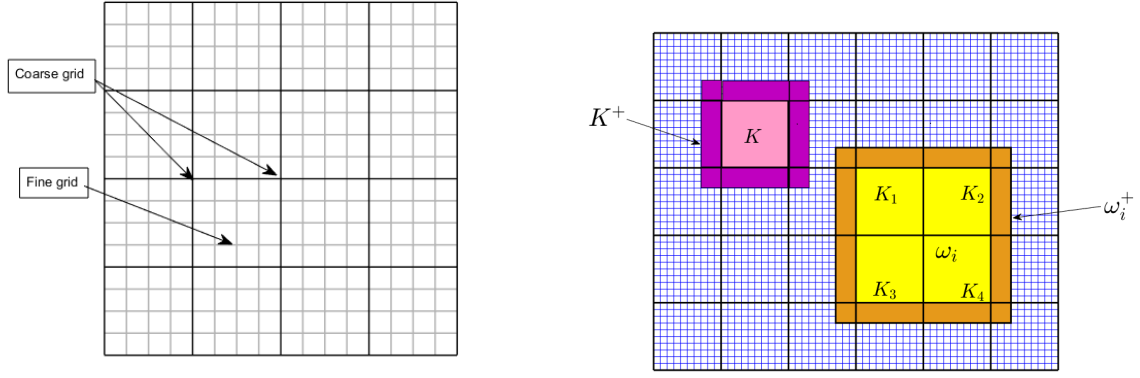
Fig. 1: Left: an illustration of fine and coarse grids. Right: an illustration of a coarse neighborhood, coarse element, and oversampled domain

defined in the snapshot space. We take the first few dominated eigenfunctions as basis functions. Using the multiscale basis functions, we obtain a reduced model.

More specifically, once the coarse and fine grids are given, one may construct the multiscale basis functions to approximate the solution of (2). To obtain the multiscale basis functions, we first define the snapshot space. For each coarse neighborhood D_i, define $J_h(D_i)$ as the set of the fine nodes of T^h lying on ∂D_i and denote the its cardinality by $L_i \in \mathbb{N}^+$. For each fine-grid node $x_j \in J_h(D_i)$, we define a fine-grid function δ_j^h on $J_h(D_i)$ as $\delta_j^h(x_k) = \delta_{j,k}$. Here $\delta_{j,k} = 1$ if $j = k$ and $\delta_{j,k} = 0$ if $j \neq k$. For each $j = 1, \cdots, L_i$, we define the snapshot basis functions $\psi_j^{(i)}$ $(j = 1, \cdots, L_i)$ as the solution of the following system

$$
\begin{aligned}
-\mathrm{div}\left(\kappa \nabla \psi_j^{(i)}\right) &= 0 \quad \text{in } D_i \\
\psi_j^{(i)} &= \delta_j^h \quad \text{on } \partial D_i.
\end{aligned}
\tag{4}
$$

The local snapshot space $V_{\mathrm{snap}}^{(i)}$ corresponding to the coarse neighborhood D_i is defined as follows $V_{snap}^{(i)} := \mathrm{span}\{\psi_j^{(i)} : j = 1, \cdots, L_i\}$ and the snapshot space reads $V_{\mathrm{snap}} := \bigoplus_{i=1}^{N_{\mathrm{in}}} V_{\mathrm{snap}}^{(i)}$.

In the second step, a dimension reduction is performed on V_{snap}. For each $i = 1, \cdots, N_{\mathrm{in}}$, we solve the following spectral problem:

$$
\int_{D_i} \kappa \nabla \phi_j^{(i)} \cdot \nabla v = \lambda_j^{(i)} \int_{D_i} \hat{\kappa} \phi_j^{(i)} v \quad \forall v \in V_{\mathrm{snap}}^{(i)}, \quad j = 1, \ldots, L_i
\tag{5}
$$

where $\hat{\kappa} := \kappa \sum_{i=1}^{N_{in}} H^2 |\nabla \chi_i|^2$ and $\{\chi_i\}_{i=1}^{N_{in}}$ is a set of partition of unity that solves the following system:

$$
\begin{aligned}
-\nabla \cdot (\kappa \nabla \chi_i) &= 0 \quad \text{in } K \subset D_i \\
\chi_i &= p_i \quad \text{on each } \partial K \text{ with } K \subset D_i \\
\chi_i &= 0 \quad \text{on } \partial D_i
\end{aligned}
$$

where p_i is some polynomial functions and we can choose linear functions for simplicity. Assume that the eigenvalues obtained from (5) are arranged in ascending order and we may use the first $1 < l_i \leq L_i$ (with $l_i \in \mathbb{N}^+$) eigenfunctions (related to the smallest l_i eigenvalues) to form the local multiscale space $V_{\mathrm{off}}^{(i)} := \mathrm{snap}\{\chi_i \phi_j^{(i)} : j = 1, \cdots, L_i\}$. The mulitiscale space $V_{\mathrm{off}}^{(i)}$ is the direct sum of the local mulitiscale spaces, namely $V_{\mathrm{off}} := \bigoplus_{i=1}^{N_{\mathrm{in}}} V_{\mathrm{off}}^{(i)}$. Once the multiscale space V_{off} is constructed, we can find the GMsFEM solution u_{off}^n at $t = t_n$ by solving the following equation

$$
\begin{aligned}
\frac{1}{\Delta t} \langle u_{\mathrm{off}}^n, v \rangle + \mathcal{A}(u_{\mathrm{off}}^n, v) &= \left\langle \frac{1}{\Delta t} u_{\mathrm{off}}^{n-1} + f(u_{\mathrm{off}}^n), v \right\rangle, \\
\langle u_{\mathrm{off}}^0, v \rangle &= \langle g, v \rangle, \quad \forall v \in V_{\mathrm{off}}.
\end{aligned}
\tag{6}
$$

2.3 Online enrichment

We will present the constructions of online basis functions [1] in this section.

Online adaptive algorithm In this subsection, we will introduce the method of online enrichment. After obtaining the multiscale space V_{off}, one may add some online basis functions based on local residuals. Let $u_{\text{off}}^n \in V_{\text{off}}$ be the solution obtained in (6) at time $t = t_n$. Given a coarse neighborhood D_i, we define $V_i := H_0^1(D_i) \cap V_{\text{snap}}$ equipped with the norm $\|v\|_{V_i}^2 := \int_{D_i} \kappa |\nabla v|^2$. We also define the local residual operator $R_i^n : V_i \to \mathbb{R}$ by

$$\mathcal{R}_i^n (v; u_{\text{off}}^n) := \int_{D_i} \left(\frac{1}{\Delta t} u_{\text{off}}^{n-1} + f(u_{\text{off}}^n) \right) v - \int_{D_i} \left(\kappa \nabla u_{\text{off}}^n \cdot \nabla v + \frac{1}{\Delta t} u_{\text{off}}^n v \right), \quad \forall v \in V_i. \tag{7}$$

The operator norm R_i^n, denoted by $\|R_i^n\|_{V_i^*}$, gives a measure of the quantity of residual. The online basis functions are computed during the time-marching process for a given fixed time $t = t_n$, contrary to the offline basis functions that are pre-computed.

Suppose one needs to add one new online basis ϕ into the space V_i. The analysis in [1] suggests that the required online basis $\phi \in V_i$ is the solution to the following equation

$$\mathcal{A}(\phi, v) = \mathcal{R}_i^n (v; u_{\text{off}}^{n,\tau}) \quad \forall v \in V_i. \tag{8}$$

We refer to $\tau \in \mathbb{N}$ as the level of the enrichment and denote the solution of (6) by $u_{\text{off}}^{n,\tau}$. Remark that $V_{\text{off}}^{n,0} := V_{\text{off}}$ for time level $n \in \mathbb{N}$. Let $\mathcal{I} \subset \{1, 2, \ldots, N_{in}\}$ be the index set over some non-lapping coarse neighborhoods. For each $i \in \mathcal{I}$, we obtain a online basis $\phi_i \in V_i$ by solving (8) and define $V_{\text{off}}^{n,\tau+1} = V_{\text{off}}^{n,\tau} \oplus \text{span} \{\phi_i : i \in \mathcal{I}\}$. After that, solve (6) in $V_{\text{off}}^{n,\tau+1}$.

Two online adaptive methods In this section, we compare two ways to obtain online basis functions which are denoted by online adaptive method 1 and online adaptive method 2 respectively. Online adaptive method 1 is adding online basis using online adaptive method from offline space in each time step, which means basis functions obtained in last time step are not used in current time step. Online adaptive method 2 is keeping online basis functions in each time step. Using this accumulation strategy, we can skip online enrichment after a certain time period when the residual defined in (7) is under given tolerance. We also presents the results of these two methods in Figure 3 and Figure 4 respectively.

Numerical results In this section, we present some numerical examples to demonstrate the efficiency of our proposed method. The computational domain is $\Omega = (0,1)^2 \subset \mathbb{R}^2$ and $T = 1$. The medium κ_1 and κ_2 are shown in Figure 2, where the contrasts are 10^4 and 10^5 for κ_1 and κ_2 respectively. Without special descriptions, we use κ_1.

For each function to be approximated, we define the following quantities e_a^n and e_2^n at $t = t_n$ to measure energy error and L^2 error respectively.

$$e_a^n = \frac{\|u_{\text{f}}^n - u_{\text{off}}^n\|_{V(\Omega)}}{\|u_{\text{f}}^n\|_{V(\Omega)}} \quad e_2^n = \frac{\|u_{\text{f}}^n - u_{\text{off}}^n\|_{L^2(\Omega)}}{\|u_{\text{f}}^n\|_{L^2(\mathcal{D})}}$$

where u_{f}^n is the fine-grid solution (reference solution) and u_{off}^n is the approximation obtained by the GMsFEM method. We define the energy norm and L^2 norm of u by

$$\|u\|_{V(\Omega)}^2 = \int_\Omega \|\nabla u\|^2 \quad \|u\|_{L^2}^2 = \int_\Omega \|u\|^2.$$

Example 2.1. In this example, we compare the error using adaptive online method 1 and uniform enrichment under different numbers of initial basis functions. We set the mesh size to be $H = 1/16$ and $h = 1/256$. The time step is $\Delta t = 10^{-3}$ and the final time is $T = 1$. The initial condition is $u(x, y, t)|_{t=0} = 4(0.5 - x)(0.5 - y)$. We set the permeability to be κ_1. We set the source term $f = \frac{1}{\epsilon^2}(u^3 - u)$,

where $\epsilon = 0.01$. We present the numerical results for for the GMsFEM at time t=0.1 in Table 1,2,and 3. For comparison, we present the results where online enrichment is not applied in Table 4. We observe that the adaptive online enrichment converges faster. Furthermore, as we compare Table 4 and Table 1, we note that the online enrichment does not improve the error if we only have one offline basis function per neighborhood. Because the first eigenvalue is small, the error decreases in the online iteration is small. In particular, for each iteration, the error decrease slightly. As we increase the number of initial offline basis, the convergence is very fast and one online iteration is sufficient to reduce the error significantly.

Example 2.2. We compare online Method 1 and 2 under different tolerance. We keep H, h and the initial condition the same as in Example 2.1. We choose intial number of basis to be 450, which means we choose two initial basis per neighborhood. We keep the source term as $f = \frac{1}{\epsilon^2}(u^3 - u)$. When $\epsilon = 0.01$, we choose the time step Δt to be 10^{-4}. We plot the error and DOF from online Method 1 in Figure 3 and compare with results from online Method 2 in Figure 4. From Figure 3 and 4, we can see the error and DOF reached stability at $t = 0.01$. In Figure 4, we can see the DOF keeps increasing before turning steady. The error remains at a relatively low level without adding online basis after some time. As a cost, online method 2 suffers bigger errors than method 1 with same tolerance. We also apply our online adaptive method 2 under permeability κ_2 in Figure 5. The errors are relatively low for two kinds of permeability.

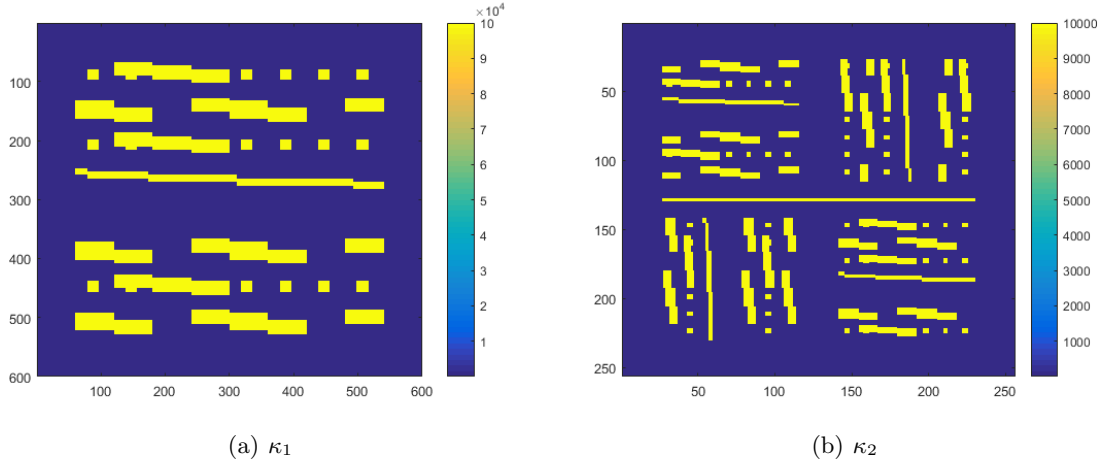(a) κ_1 (b) κ_2

Fig. 2: permeability field

Table 1: The errors for online enrichment when number of initial basis=1.
Left: Adaptive enrichment Right:Uniform enrichment.

DOF	e_a	e_2
225	14.47%	19.55%
460	2.23%	1.14%
550	1.20%	0.6%

DOF	e_a	e_2
225	14.48%	19.56%
450	8.39%	6.54%
675	2.45%	1.1%

Table 2: The errors for online enrichment when number of initial basis=2.
Left: Adaptive enrichment Right:Uniform enrichment

DOF	e_a	e_2
450	4.66%	2.64%
681	1.65%	0.52%

DOF	e_a	e_2
450	4.65%	2.64%
675	1.10%	0.669%

Table 3: The errors for online enrichment when number of initial basis=3.
Left: Adaptive enrichment Right: Uniform enrichment

DOF	e_a	e_2
675	2.89%	1.07%
903	0.944%	0.511%

DOF	e_a	e_2
675	2.89%	1.07%
900	1.13%	0.894%

Table 4: The errors for different ϵ in source term without online enrichment.
Up: energy error   Down: L^2 error

Source function	$t = 0.1$	$t = 0.2$
$\epsilon = 0.1$	5.97%	5.94%
$\epsilon = 0.01$	15.1%	15.3%
Source function	$t = 0.1$	$t = 0.2$
$\epsilon = 0.1$	4.57%	4.57%
$\epsilon = 0.01$	11.9%	12.0%

(a) error with tolerance 10^{-4}

(b) DOF with tolerance 10^{-4}

(c) error with tolerance 10^{-3}

(d) DOF with tolerance 10^{-3}
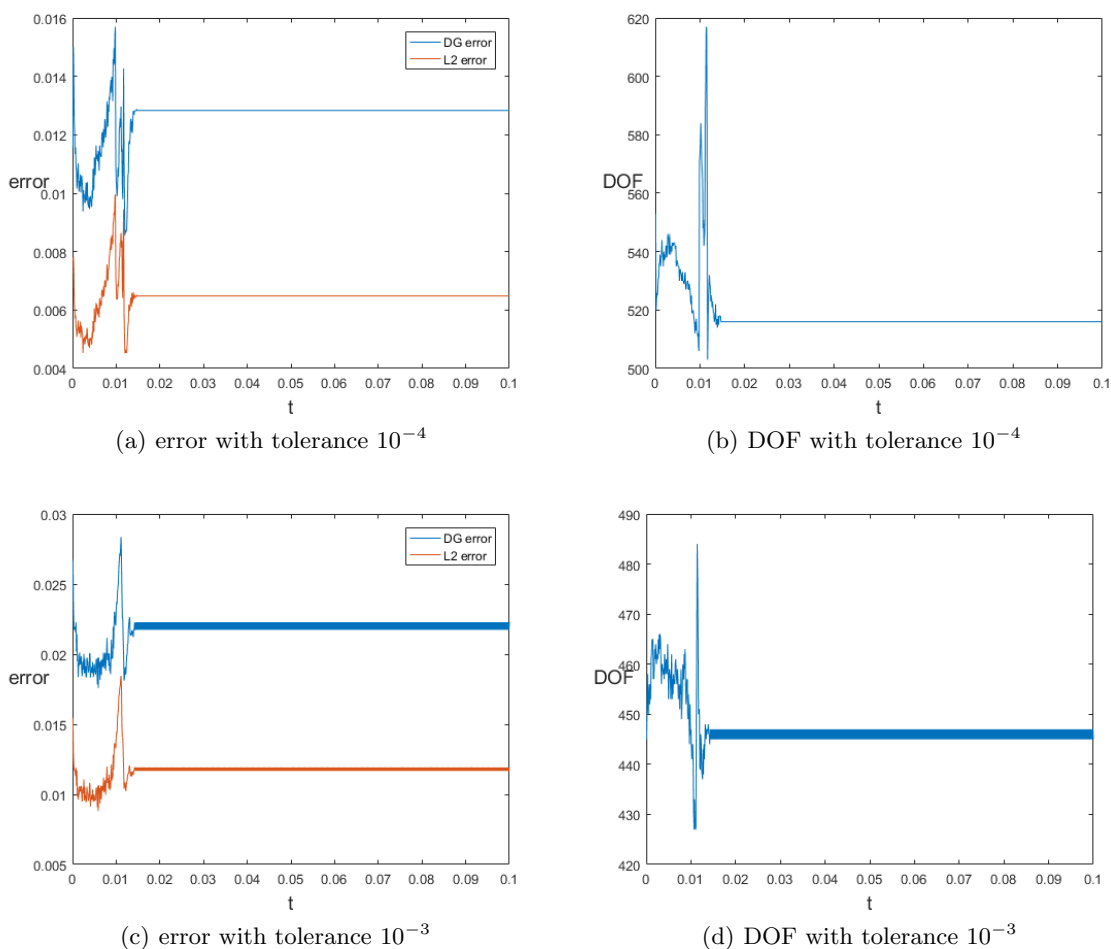
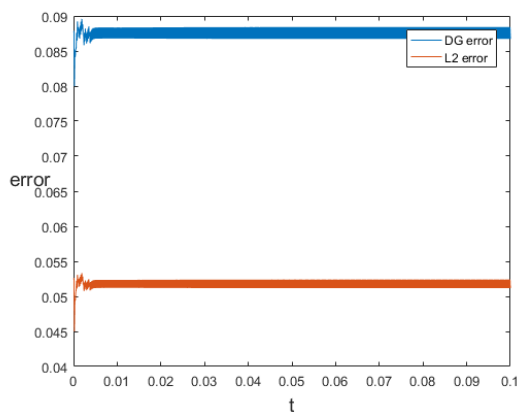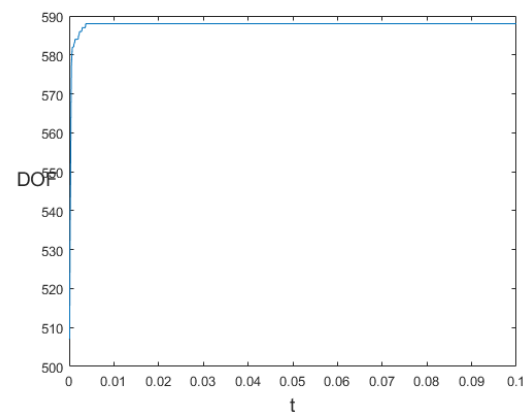Fig. 3: error and DOF obtained by online method 1 in Example 2.2
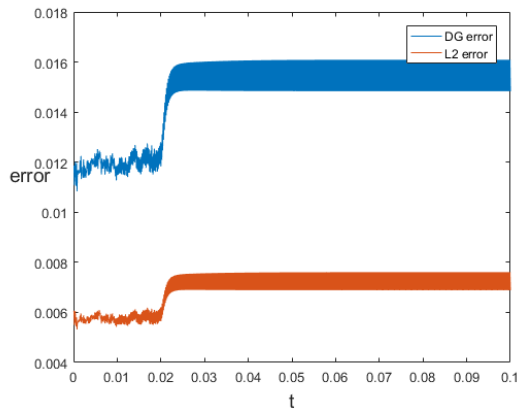
(a) error with tolerance 10^{-4}
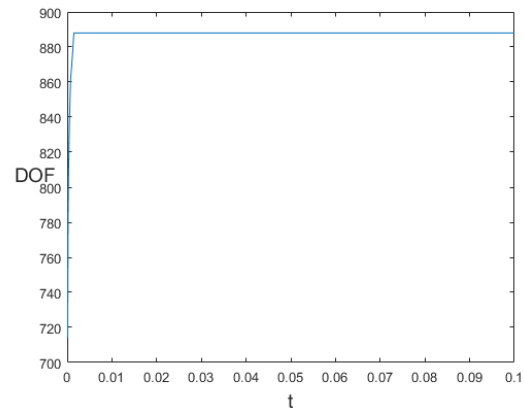
(b) DOF with tolerance 10^{-4}
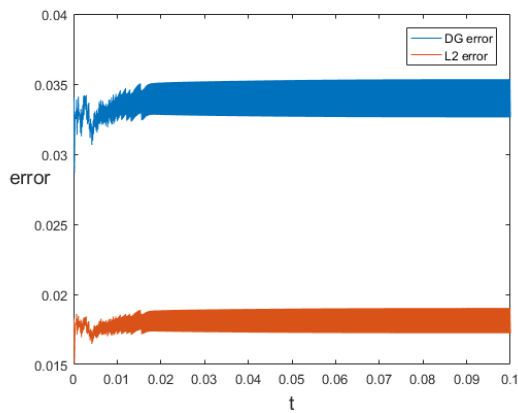
(c) error with tolerance 10^{-3}
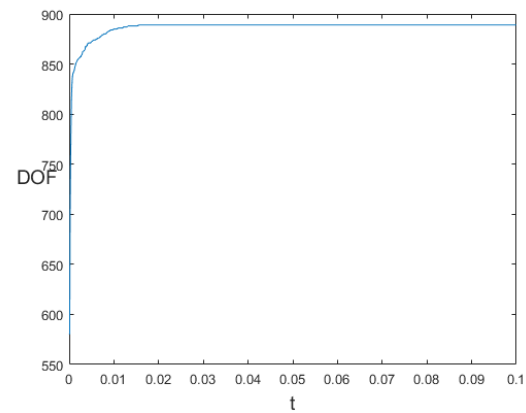
(d) DOF with tolerance 10^{-3}

(e) error with tolerance 10^{-2}

(f) DOF with tolerance 10^{-2}

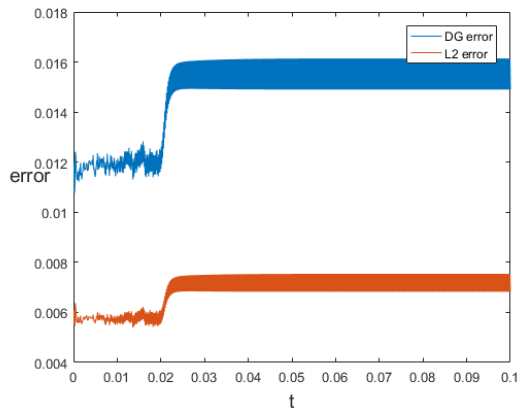Fig. 4: error and DOF obtained by online method 2 in Example 2.2
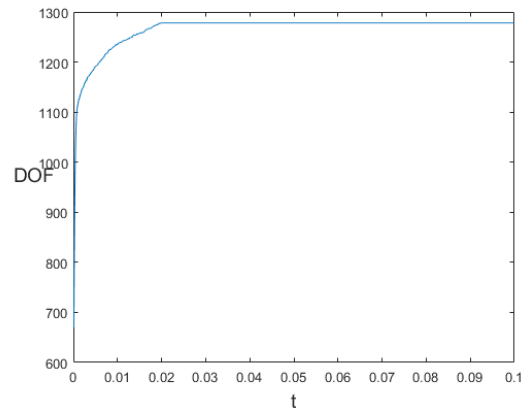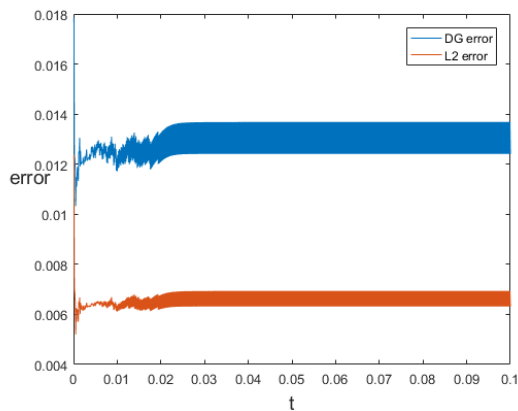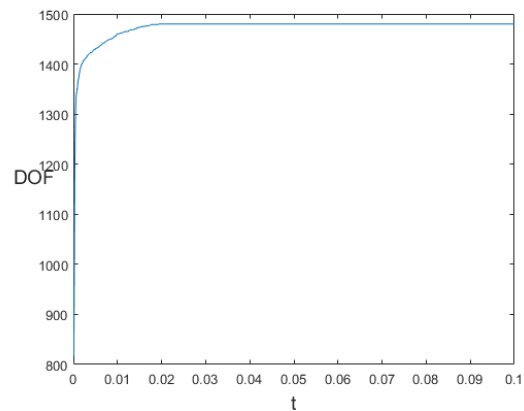
(a) error with κ_1

(b) DOF with with κ_1

(c) error with κ_2

(d) DOF with with κ_2

Fig. 5: error and DOF obtained by online method 2 in Example 2.2

3   Application to the Allen-Cahn equation

In this section, we apply our proposed method to the Allen Cahn equation. We use the Exponential Time Differencing (ETD) for time dsicretization. To deal with the nonlinear term, DEIM is applied. We will present the two methods in the following subsections.

3.1   Derivation of Exponential Time Differencing

Let τ be the time step. Using ETD, u_{off}^n is the solution to (9)

$$\langle u_{\text{off}}^n, v \rangle + \tau \mathcal{A}(u_{\text{off}}^n, v) = \langle \exp(-\frac{\tau}{\epsilon^2} \frac{f(u_{\text{off}}^{n-1})}{u_{\text{off}}^{n-1}}) u_{\text{off}}^{n-1}, v \rangle$$

$$\langle u_{\text{off}}^0, v \rangle = \langle g, v \rangle \quad \forall v \in V_{\text{off}}$$

(9)

Next, we will derive this equation. We have

$$u_t - \text{div}(\kappa \nabla u) + \frac{1}{\epsilon^2} f(u) = 0$$

Multiplying the equation by integrating factor $e^{p(u)}$, we have

$$e^{p(u)} u_t + e^{p(u)} \frac{1}{\epsilon^2} f(u) = e^{p(u)} \text{div}(\kappa \nabla u)$$

We require the above to become

$$\frac{d(e^{p(u)}u)}{dt} = e^{p(u)} \text{div}(\kappa \nabla u)$$

(10)

By solving

$$\frac{d(e^{p(u)}u)}{dt} = e^{p(u)} u_t + e^{p(u)} (\frac{d}{dt} p(u)) u,$$

we have

$$p(u(tn, \cdot)) - u(0, \cdot)) = \int_0^{t_n} \frac{1}{\epsilon^2} \frac{f(u)}{u}.$$

Using Backward Euler method in (10), we have

$$u_n - \tau \text{div}(\kappa \nabla u_n) = e^{-p(u)_n} u_{n-1}$$

(11)

where $p(u)_n = p(u(t_n) - u(t_{n-1}))$. To solve (11), we approximate (11) as follows:

$$e^{-p(u)_n} u_{n-1} \approx e^{-\frac{\tau}{\epsilon^2} \frac{f(u(t_{n-1}))}{u(t_{n-1})}} u(t_{n-1}).$$

(12)

Using above approximation, we have

$$u_{\text{off}}^n - \tau \text{div}(\kappa \nabla u_{\text{off}}^n) = \exp(-\frac{\tau}{\epsilon^2} \frac{f(u_{\text{off}}^{n-1})}{u_{\text{off}}^{n-1}}) u_{\text{off}}^{n-1}.$$

(13)

3.2   DEIM method

When we evaluate the nonlinear term, the complexity is $O(\alpha(n) + c \cdot n)$, where α is some function and c is a constant. To reduce the complexity, we approximate local and global nonlinear functions with the Discrete Empirical Interpolation Method (DEIM)[2]. DEIM is based on approximating a nonlinear function by means of an interpolatory projection of a few selected snapshots of the function. The idea is to represent a function over the domain while using empirical snapshots and information at some locations (or components).The key to complexity reduction is to replace the orthogonal projection of POD with the interpolation projection of DEIM in the same POD basis.

We briefly review the DEIM. Let $f(\tau)$ be the nonlinear function. We are desired to find an approximation of $f(\tau)$ at a reduced cost. To obtain a reduced order approximation of $f(\tau)$, we first define a reduced dimentional space for it. We would like to find m basis vectors (where m is much smaller than n), $\phi_1, \cdots, \phi_m$, such that we can write

$$f(\tau) = \Phi d(\tau),$$

where $\Phi = (\phi_1, \cdots, \phi_m)$. We employ POD to obtain Φ and use DEIM (refer Table 5) to compute $d(\tau)$ as follows. In particular, we solve $d(\tau)$ by using m rows of Φ. This can be formalized using the matrix P

$$\mathrm{P} = [e_{\wp_1}, \ldots, e_{\wp_m}] \in \mathbb{R}^{n \times m},$$

where $e_{\wp_i} = [0, \cdots, 1, 0, \cdots, 0] \in \mathbb{R}^n$ is the $\wp_i^{th}$ column of the identity matrix $I_n \in \mathbb{R}^{n \times n}$ for $i = 1, \cdots, m$. Using $P^T f(\tau) = P^T \Phi d(\tau)$, we can get the approximation for $f(\tau)$ as follows:

$$f(\tau) \approx \tilde{f}(\tau) = \Phi d(\tau) = \Phi \left(\mathrm{P}^T \Phi \right)^{-1} \mathrm{P}^T f(\tau)$$

Table 5: DEIM algorithm

DEIM	Algorithm
Input	$\Phi = (\phi_1, \cdots, \phi_m)$ obtained by applying POD on a sequence of n_s functions evaluations
Output	The interpolation indices $\overrightarrow{\lambda} = (\lambda_1, \cdots, \lambda_m)^T$
	1. Set $[\rho, \lambda_1] = \max\{|\phi_1|\}$.
	2. Set $\Phi = [\phi_1]$, $P = [e_{\lambda_1}]$, and $\overrightarrow{\lambda} = (\lambda_1)$
	3. for $i = 2, \cdots, m$, do
	Solve $(P^T \Phi)w = P^T \phi_i$ for some i.
	Compute $r = \phi_i - \Phi w$
	Compute $[\rho, \lambda_i] = \max\{|r|\}$
	Set $\Phi = [\Phi, \phi_i]$, $P = [P, e_{\lambda_i}]$, and $\overrightarrow{\lambda} = \begin{pmatrix} \overrightarrow{\lambda} \\ \lambda_i \end{pmatrix}$
	end for

3.3 Numerical results

Example 3.3. In this example, we apply the DEIM under the same setting as in Example 2.2 and we did not use the online enrichment procedure. We compare the results in Figure 8. To test the DEIM, we first consider the solution using DEIM where the snapshot are obtained by the same equation. First, we set $\epsilon = 0.01$. We first solve the same equation and obtain the snapshot Φ. Secondly, we use DEIM to solve the equation again. The two results are presents in Figure 6. The first picture are the errors we get when DEIM are not used while used in second one. The errors of these two cases differs a little since the snapshot obtained in the same equation. Then we consider the cases where the snapshots are obtained:

1. Different right hand side functions.
2. Different initial conditions.
3. Different permeability field.
4. Different time steps.

Different right hand side Since the solution for different ϵ can have some similarities, we can use the solution from one to solve the other. In particular, since it will be more time-consuming to solve the case when ϵ is smaller. We can use the $f(u)$ for $\epsilon = 0.09$ to compute the solution for $\epsilon = 0.1$ since solutions for these two cases can only vary a little. I show the results in Figure 7.

Different initial conditions In this section, we consider using the snapshot from different initial conditions, we record the results in Figure 9. We first choose the initial condition to be compared Figure 9 and Figure 6, we can see that different initial conditions can have less impact on the final solution since the solution is close to the one where the snapshot is obtained in the same equation.

Different permeability field In this section, we consider using the snapshot from different permeability, we record the results in Figure 10. For reference, the first two figures plots the fine solution and multiscale solution without using DEIM. And we construct snapshot from another permeability κ_1 and we apply it to compute the solution in κ_2. The last figure shows the of using DEIM is relatively small.

Different time steps In this section, we construct the snapshot by using nonlinear function obtained in previous time step for example when $t < 0.05$. Then we apply it to DEIM to solve the equation in $0.05 < t < 0.06$. We use these way to solve the equation with permeability κ_1 and κ_2 respectively. We plot the results in Figure 11 and 12. From these figures, we can see that DEIM have different effects applied to different permeability. With κ_1, the error increases significantly when DEIM are applied. But with κ_2, the error decreased to a lower level when we use DEIM.
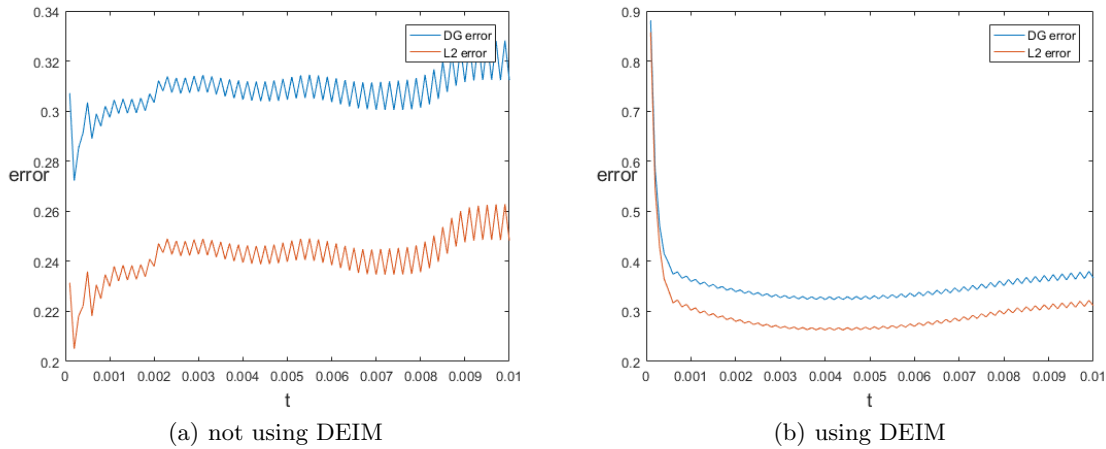

(a) not using DEIM   (b) using DEIM

Fig. 6: error for same ϵ

Acknowledgement

(a) not using DEIM   (b) using DEIM
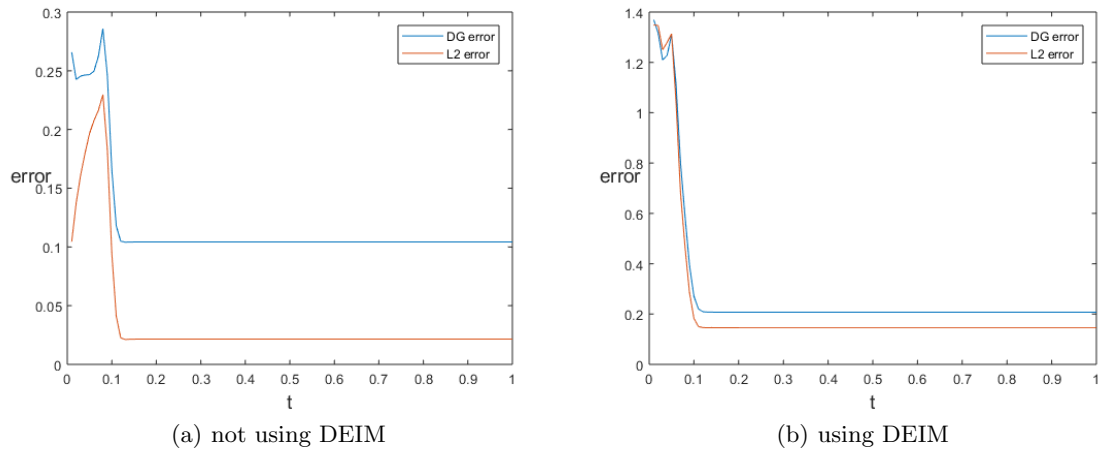
Fig. 7: error for different ϵ

(a) fine solution when $\epsilon = 0.01$ (b) multiscale solution when $\epsilon = 0.01$
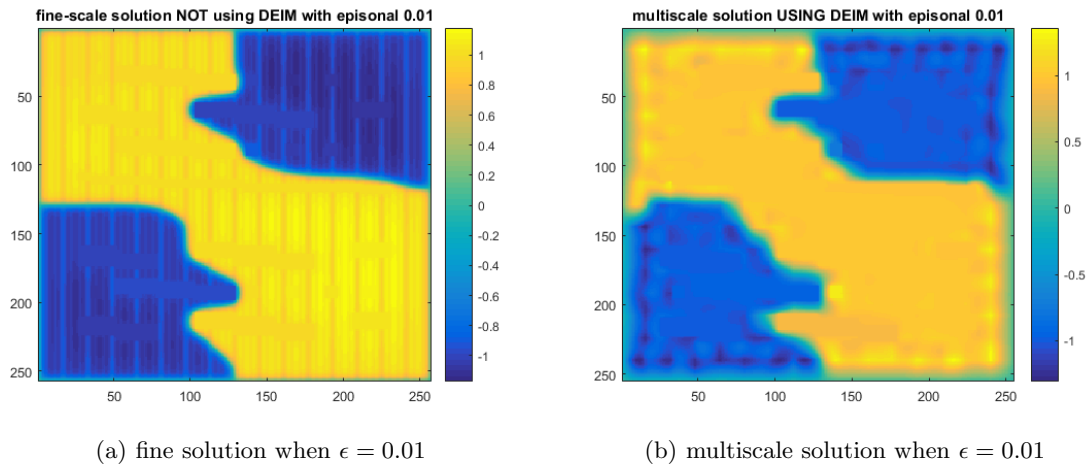
Fig. 8: Comparing fine and multiscale solutions.

(a) solution   (b) error
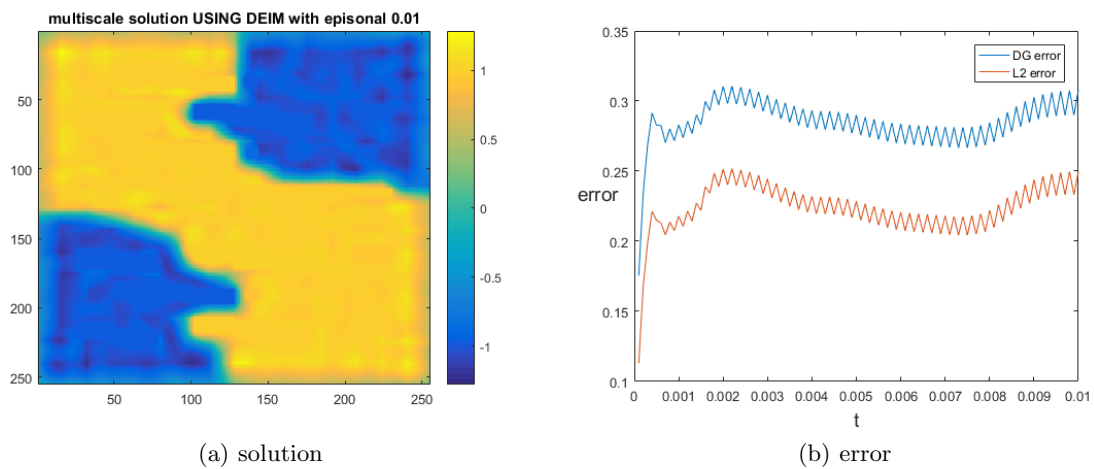
Fig. 9: using DEIM for different initial conditions

(a) fine solution under permeability κ_2

(b) multiscale solution under permeability a_2

(c) solution under permeability κ_2 using DEIM

(d) error for solution using DEIM
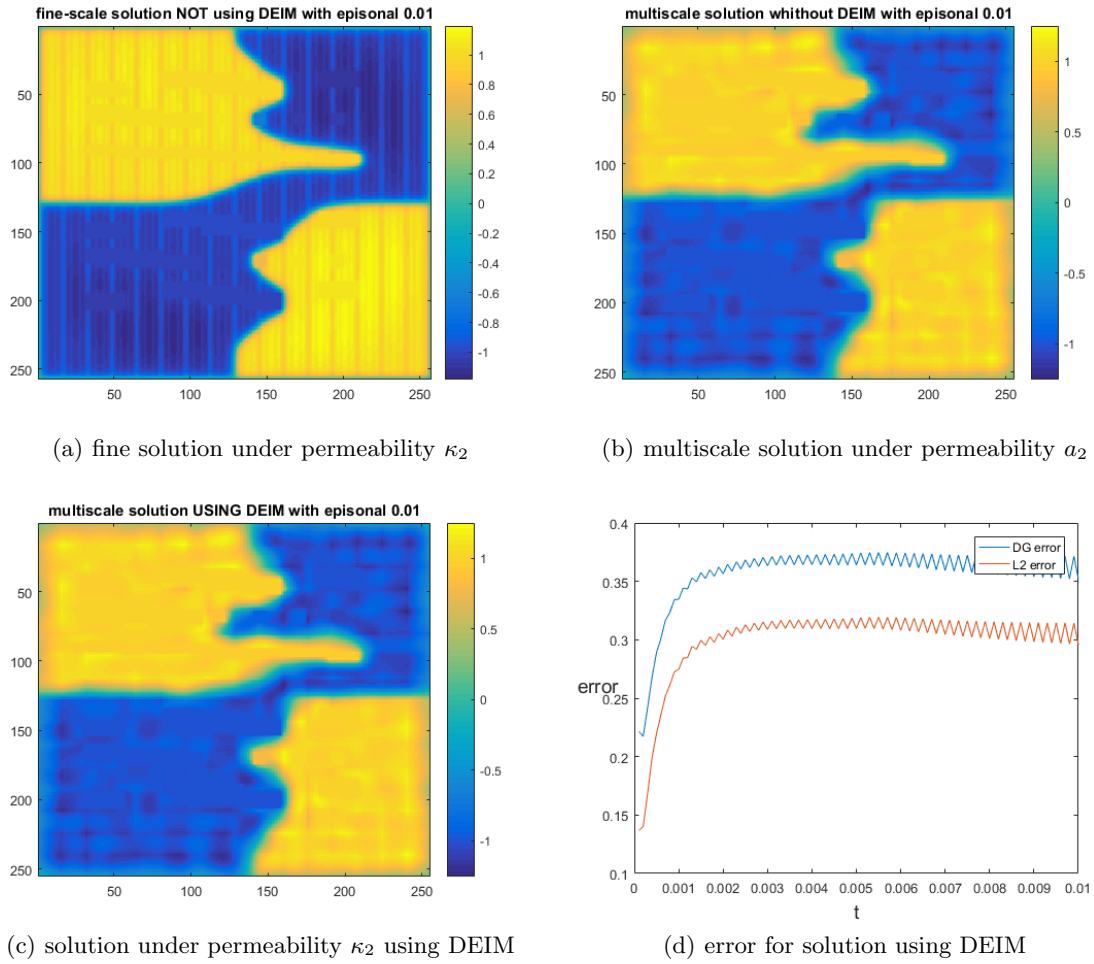
Fig. 10: using DEIM for different permeability field

(a) solution

(b) error
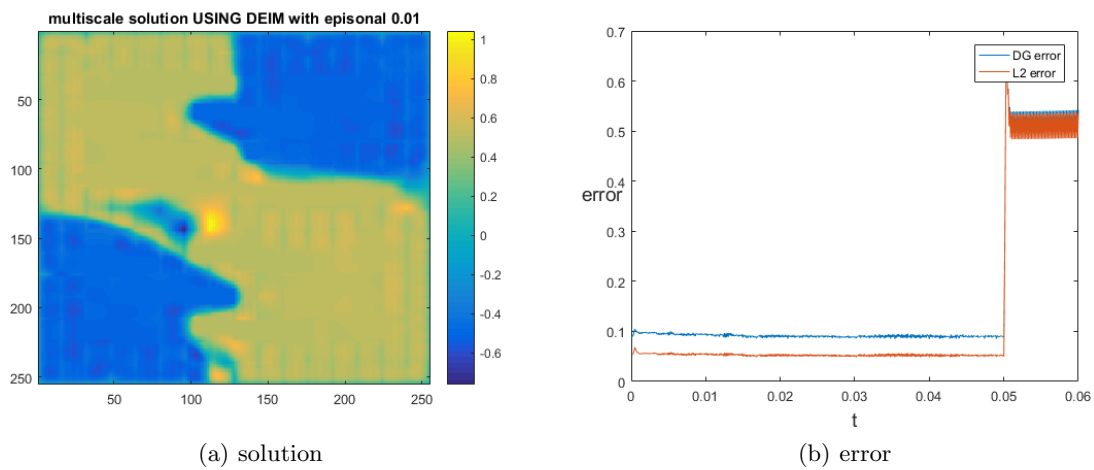
Fig. 11: using DEIM for under different time step for κ_1
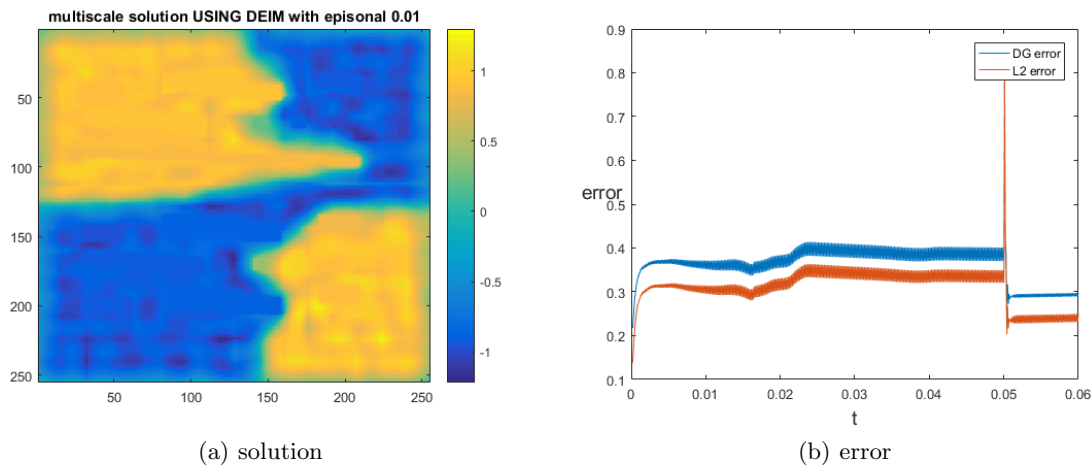
(a) solution

(b) error

Fig. 12: using DEIM for under different time step for κ_2

References

1. Eric T.Chung, F., Yalchin Efendiev, S., Wing Tat Leung, T.: Residual-driven online generalized multiscale finite element methods. J. Comput. Phys **302**, 176–190(2015)
2. Saifon Chaturantabut, F., Danny C. Sorensen, S.: Nonlinear model reduction via discrete empirical interpolation. SIAM J. Sci. Comput **32**(5), 2737–2764(2010)
3. Eric T. Chung, F.,Yalchin Efendiev, S., Thomas Y. Hou, T.: Adaptive multiscale model reduction with generalized multiscale finite element methods. J. Comput. Phys **320**, 69-95(2016)
4. Yalchin Efendiev, F., Eduardo Gildin, S., Yanfang Yang, T.: Online Adaptive Local-Global Model Reduction for Flows in Heterogeneous Porous Media. Computation **4**, 22(2016)
5. M. Hinze, F., S. Volkwein, S.:Proper orthogonal decomposition surrogate models for nonlinear dynamical systems: Error estimates and suboptimal control in Dimension Reduction of Large-Scale Systems. Lect. Notes Comput. Sci. Eng. 45, pp. 261–306. Springer, Berlin(2005)
6. K. Kunisch, F., S. Volkwein, S.: Control of the Burgers equation by a reduced-order approach using proper orthogonal decomposition. J. Optim. Theory Appl **102**, 345–371(1999)
7. W. R. Graham, F., J. Peraire, S., K. Y. Tang, T.: Optimal control of vortex shedding using loworder models. Part I—Open-loop model development. Internat. J. Numer. Methods Engrg **44**, 945–972(1999)
8. S. S. Ravindran, F.: A reduced-order approach for optimal control of fluids using proper orthogonal decomposition. Internat. J. Numer. Methods Fluids **34**, 425–448(2000)
9. C. Prud'homme, F., D. V. Rovas, S., K. Veroy, T., L. Machiels, F., Y. Maday, F., A. T. Patera, S., G. Turinici, S.: Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. J. Fluids Eng **124**, 70–80(2002)
10. L. Machiels, F., Y. Maday, S., I. B. Oliveira, T., A. T. Patera, F., D. V. Rovas, F.: Output bounds for reduced-basis approximations of symmetric positive definite eigenvalue problems. Comptes Rendus de l Académie des Sciences - Series I - Mathematics **331**, 153-158(2000)
11. Y. Maday, F., A. T. Patera, S., G. Turinici, T.: A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. J. Sci. Comput **17**, 437–446(2002)
12. K. Veroy, F., D. V. Rovas, S., A. T. Patera, T.: A posterior error estimation for reduced-basis approximation of parametrized elliptic coercive partial differential equations: "Convex inverse" bound conditioners. ESAIM Control Optim. Calc. Var **8**, 1007–1028(2002)
13. N. C. Nguyen, F., G. Rozza, S., A. T. Patera, T.: Reduced basis approximation and a posteriori error estimation for the time-dependent viscous Burgers' equation. Calcolo **46**, 157–185(2009)
14. Gregory Beylkin, F., James M. Keiser, S., Lev Vozovoiy, T.: A New Class of Time Discretization Schemes for the Solution of Nonlinear PDEs. J. Comput. Phys **147**, 362–387(1998)
15. Y. Efendiev, F., J. Galvis, S., T. Hou, T.: Generalized Multiscale Finite Element Methods. arXiv:1301.2866 [math.NA], http://arxiv.org/submit/631572.