Adaptive Variational Multiscale Methods for Convection-Diffusion Problems

Axel Målqvist,

Michael Holst, and Mats G. Larson

axel@cam.ucsd.edu

University of California, San Diego and Umeå University

Outline

- The Model Problem
- The Multiscale Method
- Implementation
- Error Representation Formula
- Adaptive Algorithm
- Numerical Examples
- Comments
- Future Work

The model problem

Model problem: Convection-Diffusion problem with multiscale features in b, $\epsilon > 0$,

$$-\epsilon \triangle u + \nabla \cdot (bu) = f \quad \text{in } \Omega,$$
$$u = 0 \quad \text{on } \Gamma.$$

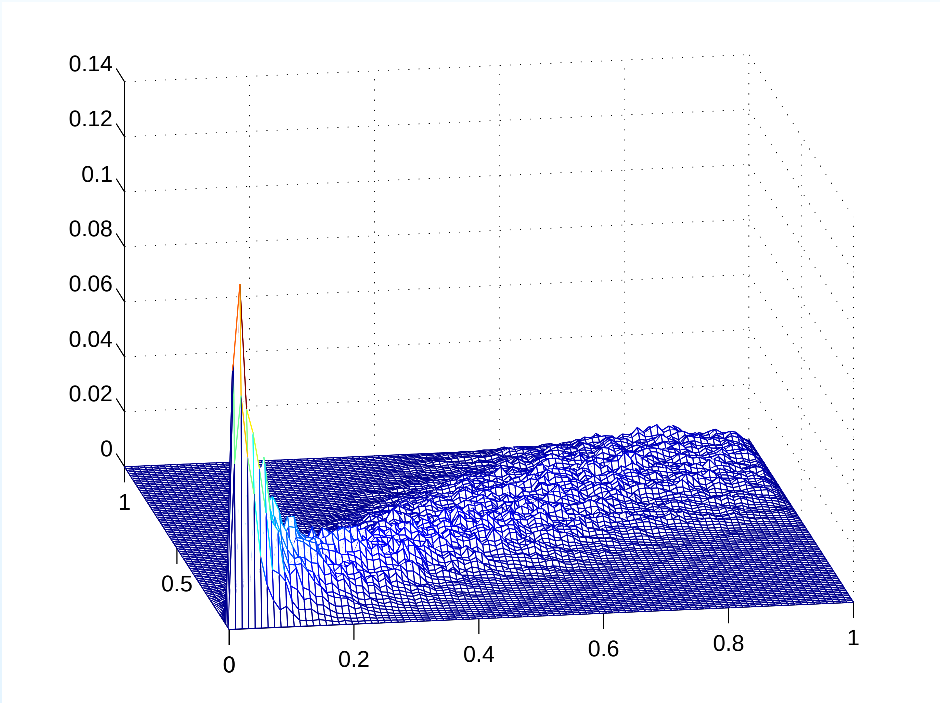
Weak form: Find $u \in V = H_0^1(\Omega)$ such that,

$$a(u, v) = l(v) \quad \text{for all } v \in H_0^1(\Omega),$$

where $a(v, w) = \int_\Omega \epsilon \nabla v \cdot \nabla w \, dx + \int_\Omega \nabla \cdot (bv) w \, dx$ and $l(v) = \int_\Omega f v \, dx$.

Example of a Solution

Let $\epsilon = 0.01$, $B =$ rand(96), $b = [B(i,j), B(i,j)]$ for $i/n < x < (i+1)/n$ and $j/n < y < (j+1)/n$, and $f = I_{\{x+y<0.05\}}$.



Mesh size: $h = 1/96$.

Our Goal

- We assume that we can form matrices and solve linear systems of equations on a coarse mesh with mesh parameter H.
- We introduce $h_{\text{ref}} < H$ as a reference mesh on which we would like to make our computations.
- By solving several "small" local problems and one coarse global problem we aim at getting a good approximation of the reference solution.

The variational multiscale method

Find $u_c \in V_c$ and $u_f \in V_f$, $V_c \oplus V_f = V$ such that,

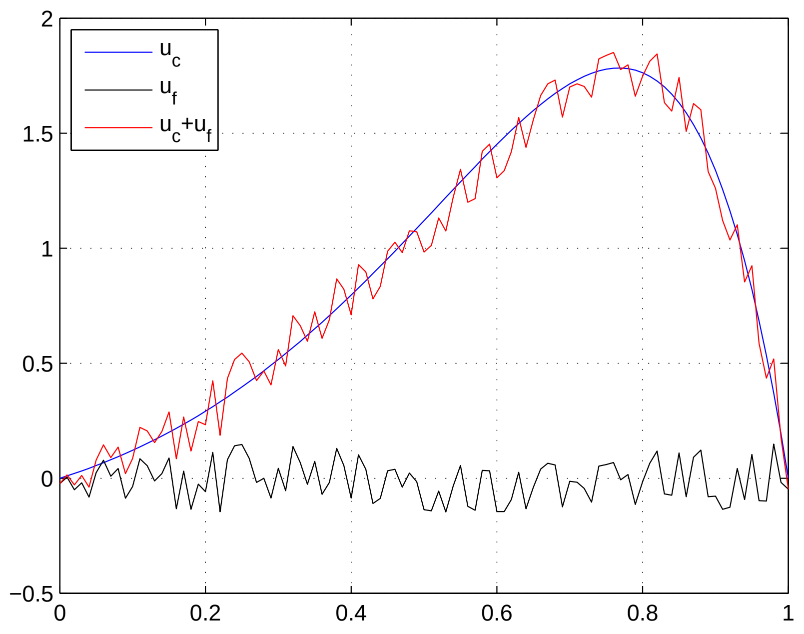
$$a(u_c + u_f, v_c + v_f) = l(v_c + v_f),$$

for all $v_c \in V_c$ and $v_f \in V_f$.

$$\begin{aligned} a(u_c, v_c) + a(u_f, v_c) &= l(v_c) \quad \text{for all } v_c \in V_c, \\ a(u_f, v_f) &= (R(u_c), v_f) \quad \text{for all } v_f \in V_f. \end{aligned}$$

where we introduce the residual distribution $R : V \to V'$,
$(R(v), w) = l(w) - a(v, w)$, for all $v, w \in V$.

The variational multiscale method



We plot u_c, u_f, and $u_c + u_f$ in a typical situation.

General framework for approximation

We derive the method in two steps.

- We decouple the fine scale equations by introducing a partition of unity $\sum_{i \in \mathcal{N}} \varphi_i = 1$,

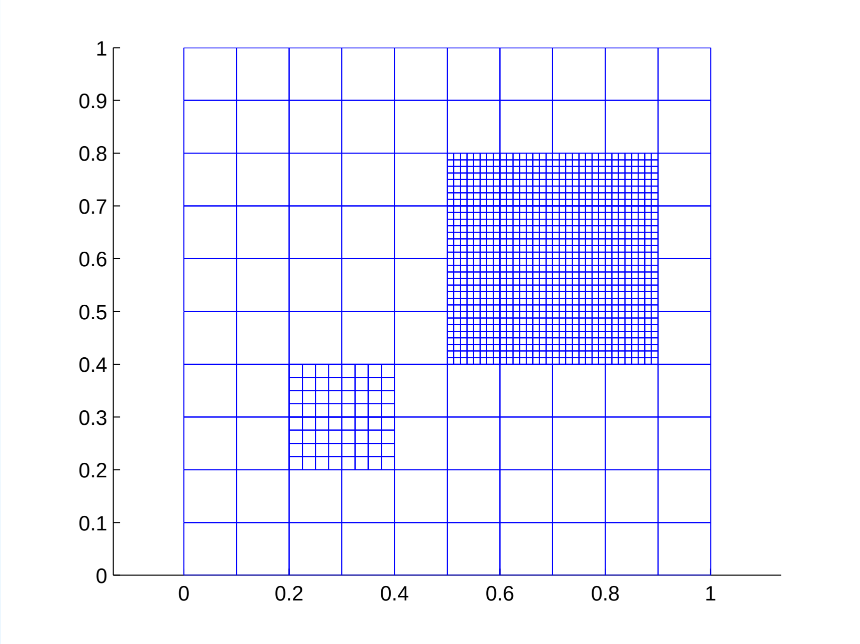
$$a(u_{f,i}, v_f) = (\varphi_i R(u_c), v_f) \quad \text{for all } v_f \in V_f.$$

- For each $i \in \mathcal{N}$ we discretize V_f and solve the resulting problem on a patch ω_i, where $\varphi_i \subset \omega_i$, rather then Ω,

$$a(U_{f,i}, v_f) = (\varphi_i R(U_c), v_f) \quad \text{for all } v_f \in V_f^h(\omega_i).$$

We use homogeneous Dirichlet bc.

The patch ω_i



One and two layer mesh stars. The coarse mesh size is H the fine mesh size h is independent between the patches and $H > h \geq h_{\text{ref}}$.

Our method

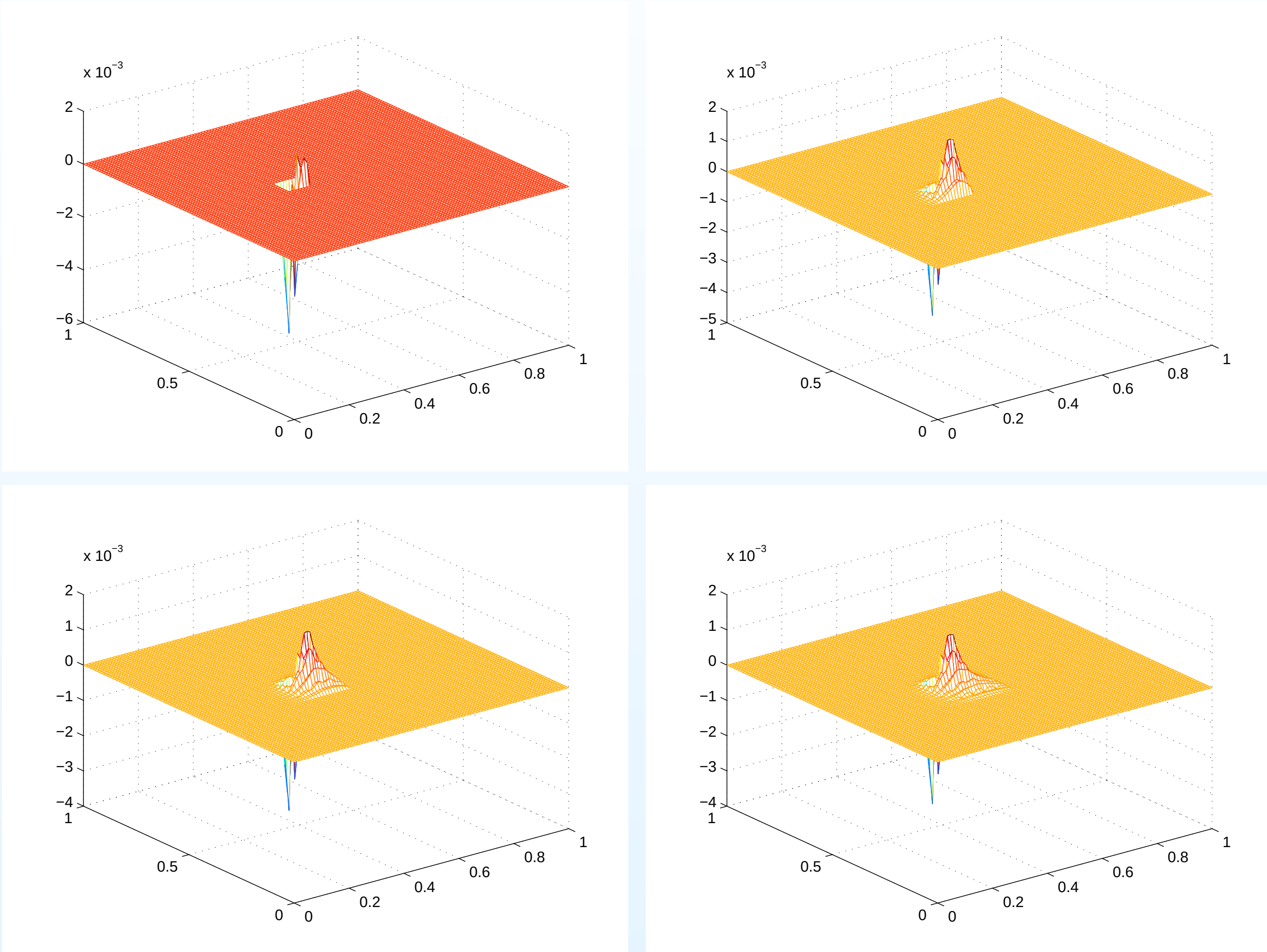
The resulting method reads: find $U_c \in V_c$ and $U_f = \sum_{i \in \mathcal{N}} U_{f,i}$ where $U_{f,i} \in V_f^h(\omega_i)$ such that

$$a(U_c, v_c) + a(U_f, v_c) = l(v_c),$$
$$a(U_{f,i}, v_f) = (\varphi_i R(U_c), v_f),$$

for all $v_c \in V_c$, $v_f \in V_f^h(\omega_i)$, and $i \in \mathcal{N}$.

The patch is chosen such that $\text{supp}(\varphi_i) \subset \omega_i \subset \Omega$.

The local solution $U_{f,i}$



The solution improves as the patch size increases.

Motivation of the method

Why do we expect the method to work?

- The right hand side of the fine scale equations, $\varphi_i R(U_c)$, has support on a coarse 1-ring if φ_i is a finite element basis function.
- The fine scale solution $U_{f,i} \in V_f^h(\omega_i)$ which is a slice space. In particular if we use the hierarchical split functions in $V_f^h(\omega_i)$ are forced to be zero in coarse nodes.

This makes $U_{f,i}$ decay, which makes it possible to get a good approximation using small patches. The size of ϵ will also affect the decay and therefore the optimal size of the patches.

Implementation

We have: find $U_{f,k} \in V_f^h(\omega_k)$ such that

$$a(U_{f,k}, v_f) = (f, v_f \varphi_k) - a(U_c, v_f \varphi_k)$$

for all $v_f \in V_f^h(\omega_k)$. Instead we solve: find $\chi_k^i, \eta_k \in V_f^h(\omega_k)$ such that

$$\begin{cases} a(\chi_k^i, v_f) = -a(\varphi_i, v_f \varphi_k) \\ a(\eta_k, v_f) = (f, v_f \varphi_k). \end{cases}$$

for all $v_f \in V_f^h(\omega_k)$ and $supp(\varphi_i) \cap supp(\varphi_k) \neq \emptyset$ i.e. $\sum_{i \in \mathcal{N}} U_c^i \chi_k^i + \eta_k$ solves:

$$a(\sum_{i \in \mathcal{N}} U_c^i \chi_k^i + \eta_k, v_f) = (f, v_f \varphi_k) - a(U_c, v_f \varphi_k),$$

Implementation

We identify $U_{f,k} = \sum_{i \in \mathcal{N}} U_c^i \chi_k^i + \eta_k$ and

$$U_f = \sum_{k \in \mathcal{N}} \sum_{i \in \mathcal{N}} U_c^i \chi_k^i + \eta_k = \sum_{i \in \mathcal{N}} U_c^i \chi^i + \eta,$$

where $\chi^i = \sum_{k \in \mathcal{N}} \chi_k^i$ and $\eta = \sum_{k \in \mathcal{N}} \eta_k$. We include this in the coarse scale equations: Find $U_c = \sum_{i \in \mathcal{N}} U_c^i \varphi_i$ such that,

$$\begin{aligned}(f, \varphi_j) =& a(U_c, \varphi_j) + a(U_f, \varphi_j) \\ &= a(\sum_{i \in \mathcal{N}} U_c^i \varphi_i, \varphi_j) + a(\sum_{i \in \mathcal{N}} U_c^i \chi^i + \eta, \varphi_j),\end{aligned}$$

for all $j \in \mathcal{N}$ or

$$\sum_{i \in \mathcal{N}} U_c^i a(\varphi_i + \chi^i, \varphi_j) = (f, \varphi_j) - a(\eta, \varphi_j).$$

Implementation

This can now be written on matrix form as,

$$(A + T)U_c = b - d$$

where,

$$\begin{cases} A_{ij} = a(\varphi_i, \varphi_j), \\ T_{ij} = a(\chi^i, \varphi_j), \\ b_j = (f, \varphi_j), \\ d_i = a(\eta, \varphi_j). \end{cases}$$

Implementing the method comes down to calculating T and d locally, $T = \sum_{k \in \mathcal{N}} T^k$ and $d = \sum_{k \in \mathcal{N}} d^k$.

$$T^k_{ij} = a(\chi^i_k, \varphi_j), \qquad d^k_j = a(\eta_k, \varphi_j).$$

These can be computed on the patches without knowing U_c.

Duality Based Error Analysis

Find $\phi \in V$ such that

$$a(w, \phi) = (w, \psi) \quad \text{for all } w \in V.$$

We end up with the following error representation formula,

$$\begin{aligned}(e, \psi) &= a(e, \phi) = l(\phi) - a(U, \phi) \\ &= \sum_{i \in \mathcal{N}} l(\varphi_i \phi) - a(U_c, \varphi_i \phi) - a(U_{f,i}, \phi).\end{aligned}$$

The oscillating coefficient b will most likely not be computed using exact quadrature. We introduce,

$$a_h(v, w) = (\epsilon \nabla v, \nabla w) + (\nabla \cdot (b_h v), w),$$

where b_h is a piecewise polynomial on the space $V_f^h(\omega_i)$.

Error Representation Formula

We continue the calculation using coarse and fine scale Galerkin Orthogonality,

$$
\begin{aligned}
(e, \psi) &= l(\phi - \pi_c \phi) - a_h(U, \phi - \pi_c \phi) + a_h(U, \phi) - a(U, \phi) \\
&= \sum_i l(\varphi_i(\phi_f - \pi^0_{f,i}\phi_f)) - a_h(U_c, \varphi_i(\phi_f - \pi^0_{f,i}\phi_f)) \\
&\qquad - a(U_{f,i}, \phi_f - \pi^0_{f,i}\phi_f) + (\nabla \cdot ((b - b_h)U), \phi),
\end{aligned}
$$

Where $\pi^0_{f,i}$ is the interpolant onto $V^h_f(\omega_i)$ i.e. zero on $\partial\omega_i$. Remember, any function in $V^h_f(\omega_i)$ can be subtracted.

We can also introduce $\pi_{f,i}$ as the nodal interpolant on the mesh associated with $V^h_f(\omega_i)$ and express the error representation formula in terms of $\pi_{f,i}$ and $\pi^0_{f,i} - \pi_{f,i}$.

Error Representation Formula

We end up with three terms,

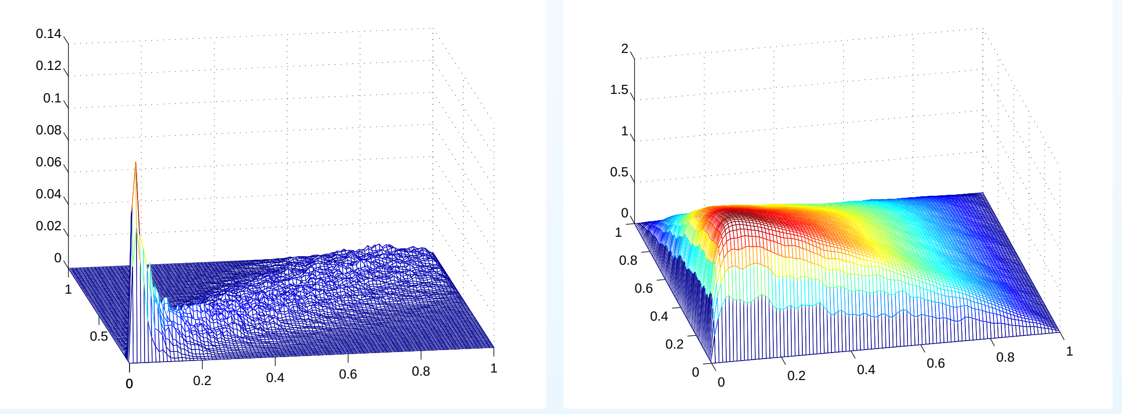
$$
\begin{aligned}
(e,\psi) = \sum_i l(\varphi_i(\phi_f - \pi_{f,i}\phi_f)) - a_h(U_c, \varphi_i(\phi_f - \pi_{f,i}\phi_f)) \\
- a(U_{f,i}, \phi_f - \pi_{f,i}\phi_f) \\
+ \sum_i (\nabla \cdot ((b - b_h)U_c), \varphi_i \phi) + (\nabla \cdot ((b - b_h)U_{f,i}), \phi) \\
+ \sum_i l(\varphi_i(\pi_{f,i}\phi_f - \pi^0_{f,i}\phi_f)) - a_h(U_c, \varphi_i(\pi_{f,i}\phi_f - \pi^0_{f,i}\phi_f)) \\
- a_h(U_{f,i}, \pi_{f,i}\phi_f - \pi^0_{f,i}\phi_f).
\end{aligned}
$$

The first term decreases with h, the second term decreases with the resolution of b_h, and the third term decreases as the patch size increases.

Solving the Dual Problem

Remember the dual problem: find $\phi \in V$ such that,

$$(\epsilon \nabla \phi, \nabla w) - (b \cdot \nabla \phi, w) = (1, w), \quad \text{for all } w \in V.$$



- Computing approximation Φ on the reference mesh or use AVMS with more refinement $\longrightarrow$ good approx. of the error.
- Computing Φ using the same method as the primal or $h = H/2$ for all local problems $\longrightarrow$ good indicator for adaptivity.

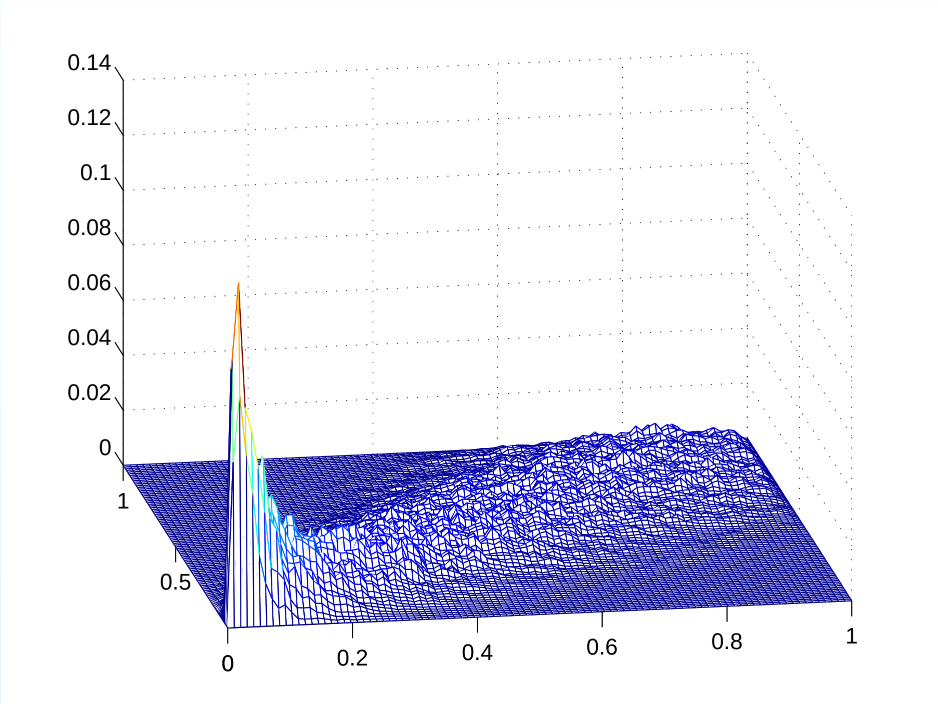
Adaptive Algorithm

$$(e, \psi) = \sum_{i \in \mathcal{N}} D_i(U, \Phi_f - \pi_{f,i}\Phi_f) + Q_i(U, \Phi) + P_i(U, \Phi_f).$$

1. Start with given r_i and L_i where $h_i = H/2^{r_i}$.
2. Calculate U and Φ.
3. Calculate D_i, Q_i, and P_i.
4. Stop if they are small enough, else order the indicators by size and let $r_i := 2r_i$ for large values in $D_i + Q_i$ and let $L_i = L_i + 1$ for large values in P_i, return to 2.

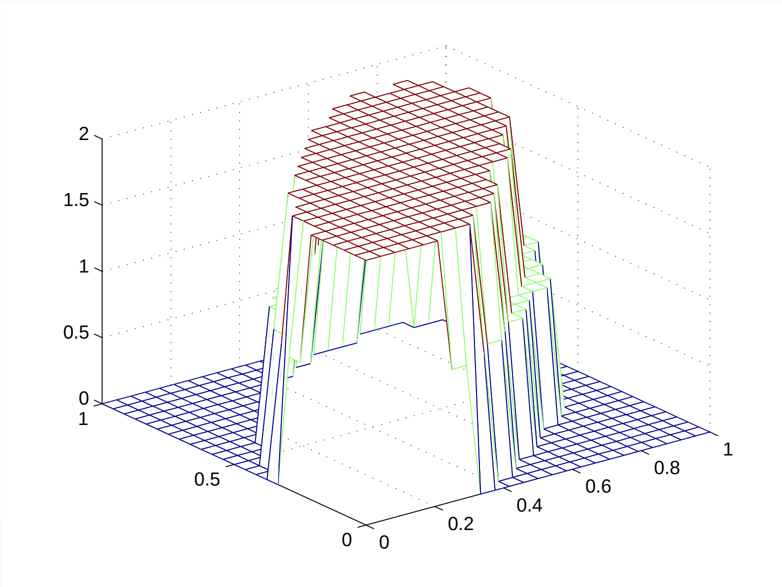
Numerical Examples

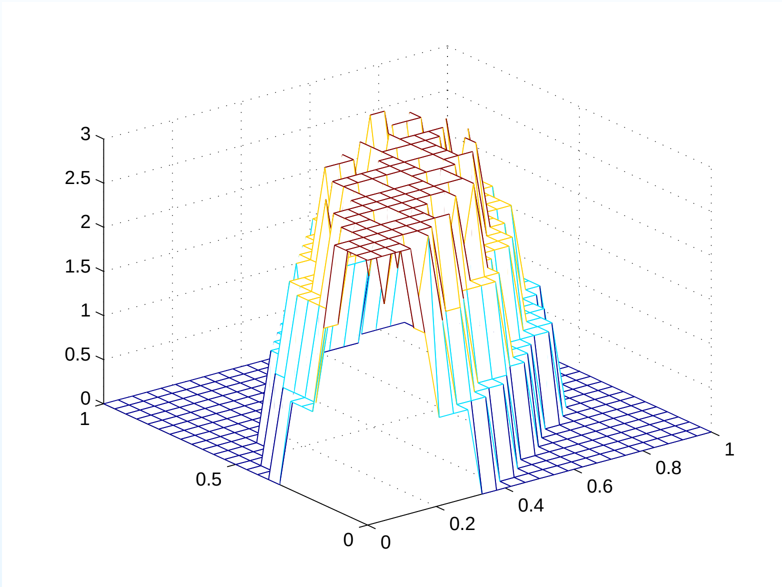
We let $\epsilon = 0.01$, $f = I_{\{x+y<0.05\}}$, and $B = \text{rand}(96)$, $b = [B(i,j), B(i,j)]$ for $i/n < x < (i+1)/n$ and $j/n < y < (j+1)/n$.



Numerical Examples

We let $\psi = 1$ and use a refinement level of 40%.

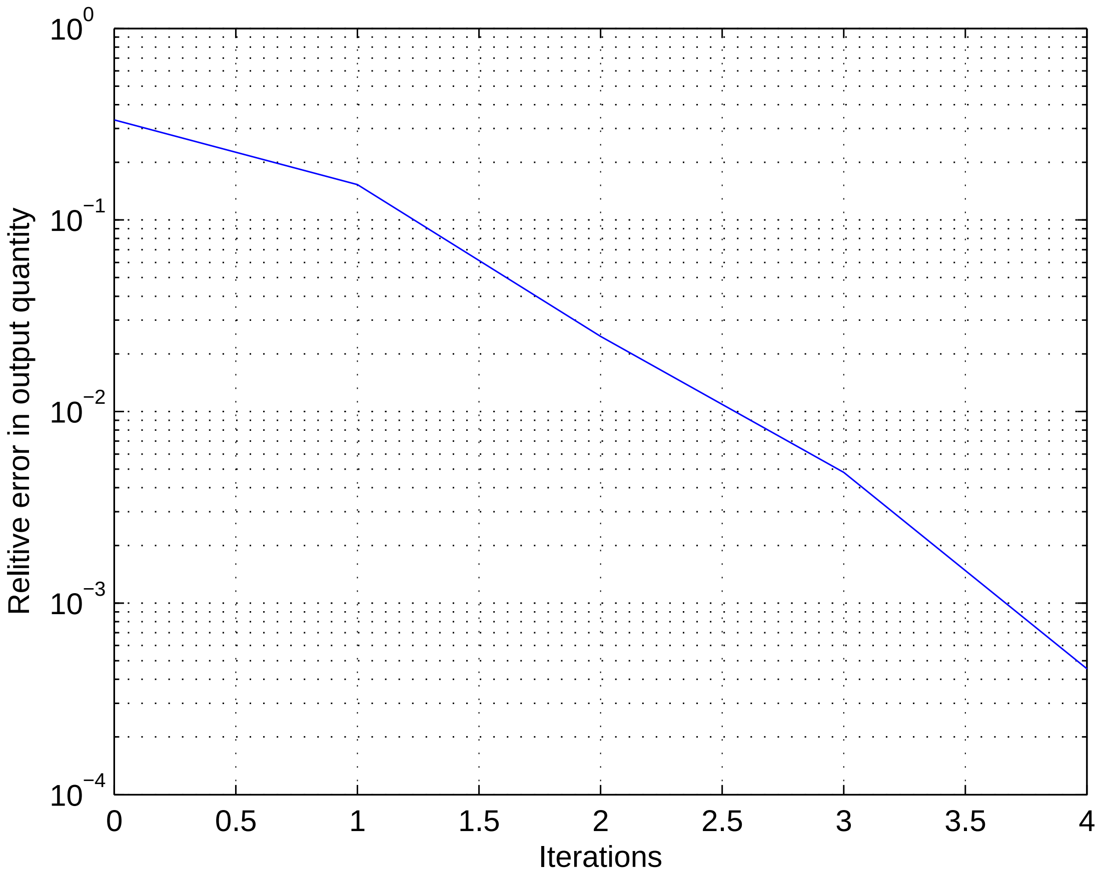




Refinements and Patchsizes.

Numerical Examples

We plot the relative error compared to a reference solution in the quantity of interest. We solve the dual and the primal using the same method.



Improvements, comments, and future work

- Patches shaped adaptively to suite $U_{f,i}$.
- A split between V_c and V_f that in a better way captures mean values of the coarse solution and perhaps depends on b.

- A poorly computed dual solution often gives a bad approximation of the error but serves as a good indicator for adaptivity.

- Prove a priori error estimates for the multiscale method.
- Use more then two scales and consider more extreme scale separation.
- Make an evaluation of how the method performs compared to other methods.