A posteriori error analysis of multiscale operator decomposition methods for multiphysics models

D. Estep¹, V. Carey², V. Ginting³, S. Tavener², T. Wildey⁴

¹Department of Mathematics and Department of Statistics, Colorado State University, Fort Collins, CO 80523
²Department of Mathematics, Colorado State University, Fort Collins, CO 80523
³Department of Mathematics, University of Wyoming, Laramie, WY 82071
⁴The Institute for Computational Engineering and Sciences, The University of Texas, Austin, TX 78712

E-mail: estep@math.colostate.edu

Abstract. Multiphysics, multiscale models present significant challenges in computing accurate solutions and for estimating the error in information computed from numerical solutions. In this paper, we describe recent advances in extending the techniques of a posteriori error analysis to multiscale operator decomposition solution methods. While the particulars of the analysis vary considerably with the problem, several key ideas underlie a general approach being developed to treat operator decomposition multiscale methods. We explain these ideas in the context of three specific examples.

1. Multiscale, multiphysics problems
Multiphysics, multiscale models that couple different physical processes interacting across a wide range of scales abound in the application areas of the SCIDAC program. Such models present significant challenges for computing accurate numerical solutions, which makes it very important to obtain accurate estimates of error in computed information. But, multiphysics, multiscale problems also raise significant challenges for error estimation. In this paper, we describe recent advances in computing a posteriori error estimates for multiscale operator decomposition, which is a powerful solution method for multiphysics, multiscale problems that is in widespread use in high performance computing.

1.1. Three examples of multiphysics models
We briefly describe three examples of multiphysics phenomena that arise in SCIDAC application domains. These examples illustrate the myriad ways in which physical processes are coupled.

Example 1.1. A thermal actuator. This is a MEMS (microelectronic mechanical switch) device modeled by a system of three coupled equations, each representing a distinct physical process acting on its own scale.

\[
\begin{align*}
\nabla \cdot (\sigma \nabla u_1) &= 0, & x &\in \Omega, \\
\nabla \cdot (\kappa(u_2)\nabla u_2) &= \sigma(\nabla u_1 \cdot \nabla u_1), & x &\in \Omega, \\
\n\nabla \cdot (\lambda \text{tr}(E)I + 2\mu E - \beta(u_2 - u_{2,\text{ref}})I) &= 0, & E &= (\nabla u_3 + \nabla u_{3,\text{ref}})^2, & x &\in \Omega
\end{align*}
\]
The first is an electrostatic current equation governing potential $u_1$ (current $J = -\sigma \nabla u_1$), the second is a steady-state energy equation for the governing temperature $u_2$, and the third equation is linear elasticity describing the steady-state displacement $u_3$.

This is an example of “parameter passing,” in which the solution of one component is used to compute the parameters and/or data for another component. Note that $u_1$ can be calculated independently, while computing $u_2$ requires $u_1$. Computing $u_3$ requires $u_2$ and therefore $u_1$.

**Example 1.2. The Brusselator problem.** This is a model of chemical dynamics [1],

$$\begin{align*}
\frac{\partial u_1}{\partial t} - k_1 \frac{\partial^2 u_1}{\partial x^2} &= \alpha - (\beta + 1)u_1 + u_1^2 u_2, \quad x \in (0,1), t > 0, \\
\frac{\partial u_2}{\partial t} - k_2 \frac{\partial^2 u_2}{\partial x^2} &= \beta u_1 - u_1^2 u_2, \quad x \in (0,1), t > 0, \\
\{ u_1(0,t) = u_1(1,t) = \alpha, u_2(0,t) = u_2(1,t) = \beta/\alpha, \quad t > 0, \\
u_1(x,0) = u_{1,0}(x), \quad u_2(x,0) = u_{2,0}(x), \quad x \in (0,1),
\end{align*}$$

(2)

where $u_1$ and $u_2$ are the concentrations of species 1 and 2, respectively.

This problem combines different physics - in this case, reaction and diffusion - in one equation. The generic picture for a reaction-diffusion equation is a relatively fast, destabilizing reaction component interacting with a relatively slow, stabilizing diffusion component. Thus, the physical components have both different scales and different stability properties.

**Example 1.3. Conjugate heat transfer between a fluid and solid object.** We consider the flow of a heat-conducting Newtonian fluid past a solid cylinder. The model consists of the heat equation in the solid and the equations governing the conservation of momentum, mass and energy in the fluid, where we apply the Boussinesq approximation to the momentum equations in the fluid. The temperature field is advected by the fluid and couples back to the momentum equations through the buoyancy term.

Let $\Omega_F$ and $\Omega_S$ be polygonal domains in $\mathbb{R}^2$ with boundaries $\partial \Omega_F$ and $\partial \Omega_S$ intersecting along an interface $\Gamma_I = \partial \Omega_S \cap \partial \Omega_F$. The complete coupled problem is

$$\begin{align*}
-\mu \Delta u + \rho_0 (u \cdot \nabla) u + \nabla p + \rho_0 \beta T_F g &= \rho_0 (1 + \beta T_0) g, \quad x \in \Omega_F, \\
-\nabla \cdot u &= 0, \quad x \in \Omega_F, \\
-k_F \Delta T_F + \rho_0 c_p(u \cdot \nabla T_F) &= Q_F, \quad x \in \Omega_F, \\
\text{interface} \quad &\begin{cases} 
T_S = T_F, \\
T_F(n \cdot \nabla T_F) = k_S(n \cdot \nabla T_S),
\end{cases} \quad x \in \Gamma_I, \\
-k_S \Delta T_S &= Q_S, \quad x \in \Omega_S,
\end{align*}$$

(3)

where we supplement these equations by imposing Dirichlet and Neumann conditions for the velocity field respectively on boundaries $\Gamma_{u,D}$ and $\Gamma_{u,N}$ and Dirichlet and Neumann conditions for the temperature fields in the fluid and the solid respectively on boundaries $\Gamma_{T_F,D}$, $\Gamma_{T_F,N}$, $\Gamma_{T_S,D}$, and $\Gamma_{T_S,N}$. Here, $\rho_0$ and $T_0$ are reference values for the density and temperature, respectively, $\mu$ is the molecular viscosity, $\beta$ is the coefficient of thermal expansion, $c_p$ is the specific heat, $k_F$ and $k_S$ are the thermal conductivities of the fluid and solid, respectively, $Q_F$ and $Q_S$ are source terms and $n$ is the unit normal vector directed into the fluid. Note that $u$ is a vector.

This presents a class of problems where physics in different physical domains are coupled across a common boundary.

1.2. Challenges and goals for multiscale, multiphysics models

Challenges arising in the numerical solution of multiscale, multiphysics problems include producing accurate numerical solutions efficiently, dealing with complex stability properties,
and transforming information across scales. Another complication is the range of applications of multiphysics models, including model prediction, sensitivity analysis, and parameter optimization, that require computation of solutions corresponding to wide range of data and parameters. We expect the solution behavior to vary significantly and the ability to obtain accurate numerical solutions therefore to vary as well.

On the other hand, predictive science and engineering have come to rely on high performance simulation of complex physical phenomena. In this context, it is critically important to accurately quantify the numerical error in any computed information. Another important goal is determining efficient ways to compute specific information accurately.

2. Multiscale operator decomposition

Multiscale operator decomposition is a widely used technique for solving multiphysics, multiscale models in SCIDAC simulations. The general approach is to decompose the multiphysics and/or multiscale problem into components involving simpler physics over a relatively limited range of scales, and then to seek the solution of the entire system through some sort of iterative procedure involving numerical solutions of the individual components (see figure 1). In general, different components are solved with different numerical methods as well as with different scale discretizations.

Reasons that this approach is appealing include:

- There is generally a good understanding of how to solve a broad spectrum of single physics problems accurately and efficiently.
- It provides an alternative to accommodating multiple scales in one discretization.
- It allows for efficient high performance simulations.

However, multiscale operator decomposition generally has significant impact on the accuracy and stability of numerical solutions.

3. Analysis of multiscale operator decomposition

In general, multiscale, multiphysics problems have complex stability properties that defy accurate description via classic a priori analysis. We use duality and adjoint operators to quantify stability a posteriori. We focus on computing a particular quantity of interest determined by a linear functional. We combine these tools with variational analysis to produce accurate estimates of the effects of perturbation and error. By numerically solving the adjoint problem, we can compute very accurate error estimates.

Duality and adjoint operators have a long history of application in model sensitivity analysis and optimization, dating back to Lagrange: see [2, 3, 4, 5] for example. The application of these tools to a posteriori error estimation has a more recent history, see [6, 7, 8, 9, 10, 11, 12, 13].
Our main purpose is to describe recent advances in extending the techniques of a posteriori error analysis to multiscale operator decomposition solutions of multiphysics, multiscale problems. While the particulars of the analysis vary considerably with the problem, there are several key ideas underlaying a general approach being developed to treat operator decomposition multiscale methods, including the following:

• We identify auxiliary quantities of interest associated with information passed between physical components and solve auxiliary adjoint problems to estimate the error in those quantities.
• We deal with scale differences by introducing projections between discrete spaces used for component solutions and estimate the effects of those projections.
• The standard linearization argument used to define an adjoint operator associated with error analysis for a nonlinear problem may fail, requiring a new approach. The adjoint operator associated with a multiscale operator decomposition solution method is often significantly different than the adjoint associated with the original problem, which affects the stability of the method.
• In practice, solving the adjoint associated with the original fully-coupled problem may present the same kinds of computational challenges posed by the original problem, so attention must be paid to the solution of the adjoint problem.

We explain these ideas in the context of three examples.

3.1. Multiscale decomposition of triangular systems of elliptic problems

We capture the essential features of the thermal actuator model in Example 1.1 using a “one-way” coupled system

\[
\begin{align*}
- \nabla \cdot a_1 \nabla u_1 + b_1 \cdot \nabla u_1 + c_1 u_1 &= f_1(x), & x &\in \Omega, \\
- \nabla \cdot a_2 \nabla u_2 + b_2 \cdot \nabla u_2 + c_2 u_2 &= f_2(x, u_1, Du_1), & x &\in \Omega, \\
u_1 = u_2 &= 0, & x &\in \partial \Omega,
\end{align*}
\]

(4)

where \(a_i, b_i, c_i, f_i\) are smooth functions, with \(a_1, a_2 \geq \alpha > 0\) on a bounded domain \(\Omega\) in \(\mathbb{R}^N\) with boundary \(\partial \Omega\), and \(\alpha\) is a constant.

We introduce the finite-element space \(S_{h,1}(\Omega) \subset H^1_0(\Omega)\), corresponding to a discretization \(T_{h,1}\) of \(\Omega\) for the first component, and another finite element space \(S_{h,2}(\Omega)\), on a different mesh \(T_{h,2}\), for the second component. To evaluate integrals involving functions defined on different meshes, we introduce projections \(\Pi_{i-j}\) from \(S_{h,i}\) to \(S_{h,j}\), for example, interpolants or an \(L^2\) orthogonal projection. We also define the bilinear forms, \(A_i(u_i, v_i) = (a_i \nabla u_i, \nabla v_i) + (b_i(x) \cdot \nabla u_i, v_i) + (c_i u_i, v_i)\), \(i = 1, 2\). We present the multiscale operator decomposition algorithm in figure 1.

**Algorithm 1** Multiscale Operator Decomposition for Triangular Systems of Elliptic Equations

1. Construct discretizations \(T_{h,1}, T_{h,2}\) and corresponding spaces \(S_{h,1}, S_{h,2}\).
2. Compute \(U_1 \in S_{h,1}(\Omega)\) satisfying \(A_1(U_1, v_1) = (f_1, v_1)\), for all \(v_1 \in S_{h,1}(\Omega)\).
3. Compute \(U_2 \in S_{h,2}(\Omega)\) satisfying \(A_2(U_2, v_2) = (f_2(x, \Pi_{1-2} U_1, \Pi_{1-2} Du_1), v_2)\), for all \(v_2 \in S_{h,2}(\Omega)\).

We observe that any errors made in the solution of the first component affect the solution of the second component. This turns out to be a crucial fact for a posteriori error analysis.
Example 3.1. We solve a system

\[
\begin{aligned}
-\Delta u_1 &= \sin(4\pi x)\sin(\pi y), & x \in \Omega \\
-\Delta u_2 &= b \cdot \nabla u_1 = 0, & x \in \Omega, \\
u_1 &= u_2 = 0, & x \in \partial \Omega,
\end{aligned}
\]

using a standard piecewise linear, continuous finite element method to compute the quantity of interest, we denote the primary quantity of interest by \((\psi, e)\). We solve for \(u_1\) first and then solve for \(u_2\) using independent meshes.

Using uniform meshes, evaluating the standard a posteriori error estimate for the second component problem while ignoring any effect arising from error in the solution of the first component, we estimate the error in the quantity of interest to be \(\approx 0.0042\). However, the true error is \(\approx 0.0048\), so there is discrepancy of \(\approx 0.0006\) \((\approx 13\%)\) in the estimate. This is a consequence of ignoring the error “transferred” from the solution of the first component.

If we adapt the mesh for the solution of the second component based on the a posteriori error estimate of the error in that component while neglecting the effects of the decomposition, the discrepancy becomes alarmingly worse. For example, we can refine the mesh until the estimate of the error in the second component is \(\approx 0.0001\). But, we find that the true error is \(\approx 0.2244\)! See figure 2.

**Figure 2.** Multiscale operator decomposition solutions \((u_1\) on the left, \(u_2\) in the middle) of the component problems of (5). We solve for \(u_2\) using a mesh adapted to control the pointwise error of the numerical solution of \(u_2\) while ignoring the effect of errors in the approximation of \(u_1\). On the right, we plot the pointwise error of the computed \(u_2\).

3.1.1. Description of the a posteriori analysis  We seek the error in a quantity of interest given by a functional of \(e_2 = U_2 - u_2\). Since we introduce some additional, auxiliary quantities of interest, we denote the primary quantity of interest by \((\psi, e) = (\psi^{(1)}_2, e_2)\). We use the adjoint operators

\[
A^*_i (\phi_i, v_i) = (a_i \nabla \phi_i, \nabla v_i) - (\text{div}(b_i \phi_i), v_i) + (c_i \phi_i, v_i), \quad i = 1, 2.
\]

We also use the linearization \(L_{f_2}(u_1)(u_1 - U_1) = \int_0^1 \frac{\partial f_2}{\partial u}(u_1 s + U_1 (1 - s)) \, ds\).

Noting that the solution of the first adjoint component is not needed to compute the quantity of interest, we define the primary adjoint problem to be

\[
A^*_2 (\phi^{(1)}_2, v_2) = (\psi^{(1)}_2, v_2), \quad \text{for all } v_2 \in \tilde{W}^1_2(\Omega).
\]

The error representation formula for the transfer error is

\[
(D f_2(U_1) \times e_1, \Pi_{2-1} \phi^{(1)}_2) + (D f_2(U_1) \times e_1, (I - \Pi_{2-1}) \phi^{(1)}_2),
\]

(6)
where $\Pi_{i \rightarrow j}$ is a projection from discretization $i$ to discretization $j$. The first term in (6) is the error contribution arising from the transfer while the second term is significant when the approximation spaces are different. The transfer error is a (nominally nonlinear) functional of the error in $u_1$, defining an auxiliary quantity of interest. We approximate it by a linear functional and define the auxiliary quantity of interest,

$$
(f_2(u_1) - f_2(U_1), \phi_2^{(1)}) \approx (Df_2(U_1) \times e_1, \Pi_{2 \rightarrow 1} \phi_2^{(1)}) = (\psi_1^{(2)}, e_1).
$$

We define the corresponding transfer error adjoint problem

$$
A_1^*(\phi_1^{(2)}, v_1) = (\psi_1^{(2)}, v_1) \text{ for all } v_1 \in \tilde{W}_2^1(\Omega). \tag{7}
$$

The additional term $(Df_2(U_1) \times e_1, (I - \Pi_{2 \rightarrow 1}) \phi_2^{(1)})$ is a linear functional, so we define another auxiliary quantity of interest

$$
(\psi_1^{(3)}, e_1) = (Df_2(U_1) \times e_1, (I - \Pi_{2 \rightarrow 1}) \phi_2^{(1)})
$$

and the corresponding adjoint problem

$$
A_1^*(\phi_1^{(3)}, v_1) = (\psi_1^{(3)}, v_1) \text{ for all } v_1 \in \tilde{W}_2^1(\Omega). \tag{8}
$$

We then define the weak residuals

$$
R_1(U_1, \chi; \nu) = (f_1(\nu), \chi) - A_1(U_1, \chi), \quad R_2(U_2, \chi; \nu) = (f_2(\Pi_{1 \rightarrow 2} \nu), \chi) - A_2(U_2, \chi).
$$

The final error representation is therefore [14]

**Theorem 3.2**

$$
(\psi, e) = R_2(U_2, (I - \Pi_2) \phi_2^{(1)}; U_1) + R_1(U_1, (I - \Pi_1)(\phi_1^{(2)} + \phi_1^{(3)})) + (\Pi_{1 \rightarrow 2} f_2(U_1) - f_2(\Pi_{1 \rightarrow 2} U_1), \phi_2^{(1)}) + ((I - \Pi_{1 \rightarrow 2}) f_2(U_1), \phi_2^{(1)}). \tag{9}
$$

We emphasize that evaluating the integrals in (9) is far from trivial. We have used Monte-Carlo techniques with good results; see [14].

**Example 3.3.** In Example 3.1, we estimate the contributions to the error reported in that computation using the relevant portions of (9). To produce the adaptive mesh results shown in figure 2, we construct the adapted mesh using equidistribution based on a bound derived from the first term in (9), that is, neglecting the terms that estimate the transfer error.

Instead, we consider the system (5) for the quantity of interest equal to the average value of $U_2$. We begin with the same initial coarse meshes as in figure 2, but add the transfer error expression to the mesh refinement criterion. Adapting the mesh so that the total error in the quantity of interest for $U_2$ has error estimates less than $10^{-4}$ yields the meshes shown in figure 3. We see that the first component solve requires significantly more refinement than does the second component.

### 3.2. Multiscale decomposition of reaction-diffusion problems

Operator splitting for reaction-diffusion problems is the classic example of multiscale operator decomposition. We consider

$$
\begin{align*}
\frac{\partial u}{\partial t} &= \epsilon \Delta u + f(u), & x \in \Omega, 0 < t, \\
& \text{suitable boundary conditions}, & x \in \partial \Omega, 0 < t, \\
u(\cdot, 0) &= u_0(\cdot)
\end{align*}
$$

\[\tag{10}\]
where $\Omega \subset \mathbb{R}^d$ is a spatial domain and $f$ is a smooth function. Employing the method of lines, we discretize in space using a continuous, piecewise linear finite element method with $M$ elements and obtain the initial value problem: Find $y \in \mathbb{R}^M$ such that

$$
\begin{cases}
\dot{y} = Ay(t) + F(y(t)), & 0 < t \leq T, \\
y(0) = y_0,
\end{cases}
$$

where $A$ is an $l \times l$ constant matrix representing a “diffusion component” and $F(y) = (F_1(y), F_2(y), \ldots, F_l(y))^T$ is a vector of nonlinear functions representing a “reaction component”.

We next discretize in time using a discontinuous (dG) Galerkin method [15, 8]. With appropriate choices of quadrature, we can treat standard integration methods. For each diffusion step, we choose a (small) time step $\Delta t = t_1 - t_0$. For each reaction step, we choose a (small) time step $\Delta s_n = \Delta t_n / M_n$ with $\Delta s = \max_{1 \leq n \leq N} \Delta s_n$, and the nodes $t_{n-1} = s_{0,n} < s_{1,n} < \ldots < s_{M_n,n} = t_n$. We define the time intervals $I_n = [t_{n-1}, t_n]$ and $I_{m,n} = [s_{m-1,n}, s_{m,n}]$. See Fig. 4. The finite-element approximate solutions are sought in piecewise polynomial spaces for the diffusion and reaction components, respectively:

$$\mathcal{P}^{(q_d)}(I_n)$$

$$\mathcal{P}^{(q_r)}(I_{m,n})$$

for $n = 1, \ldots, N$ and, for each $n$, $1 \leq m \leq M_n$. $\mathcal{P}^{(q)}(I)$ denotes the space of polynomials in $\mathbb{R}^d$ of degree $q$ on interval $I$. We let $U_n^+$ and $U_n^-$ denote the left- and right-hand limits of $U$ at $t_n$ and $[U] = U_n^+ - U_n^-$ the jump value of $U$ at $t_n$. 

**Figure 3.** The adapted meshes resulting from the full estimate that accounts for “primary” and “transfer” errors. The transfer error dominates and the mesh for the first component is refined while the mesh for the second component is not. Nonetheless, the simulation yields an accurate value for the quantity of interest. It is interesting to compare this result with that shown in Fig. 2. Since the quantity of interest is computed from the second component, intuition might suggest that the mesh for the second component needs refinement. But, it is the mesh for the first component that is critical for determining accuracy.

**Figure 4.** Discretization of time used for multiscale operator splitting.
Let \( \tilde{Y}(t) \) be the piecewise continuous finite element approximation of the operator splitting with

\[
\tilde{Y}(t) = \frac{t_n - t}{\Delta t_n} \tilde{Y}_{n-1} + \frac{t - t_{n-1}}{\Delta t_n} \tilde{Y}_n, \quad t_{n-1} \leq t \leq t_n.
\]

The nodal values \( \tilde{Y}_n \) are obtained from the following procedure:

**Algorithm 2** Multiscale Operator Splitting for Reaction-Diffusion Equations

Set \( \tilde{Y}_0 = y_0 \) for \( n = 1, \ldots, N \) do

Set \( Y^{r}_0, n = \tilde{Y}_{n-1} \)

for \( m = 1, \ldots, M_n \) do

Compute \( Y^r|_{I_{m,n}} \in \mathcal{P}(qr)(I_{m,n}) \) satisfying

\[
\int_{I_{m,n}} \left( \dot{Y}^r, W \right) dt + \left( [Y^r]_{m-1,n}, W^+_{m-1} \right) = \int_{I_{m,n}} (F(Y^r), W) dt \quad \forall \ W \in \mathcal{P}(qr)(I_{m,n})
\]

end for

Set \( Y^{d}_{n-1} = Y^r_{M_n, n} \), compute \( Y^d|_{I_n} \in \mathcal{P}(qd)(I_n) \) satisfying

\[
\int_{I_n} \left( \dot{Y}^d, V \right) dt + \left( [Y^d]_{n-1}, V^+_{n-1} \right) = \int_{I_n} (A Y^d, V) dt \quad \forall \ V \in \mathcal{P}(qd)(I_n)
\]

Set \( \tilde{y}_n = y^d(t_n^-) \)

end for

**Example 3.4.** We illustrate the instability of operator splitting applied to the Brusselator problem (2). We apply a standard first order splitting scheme to a space discretization of the Brusselator model with 500 discrete points with \( \alpha = .6, \beta = 2, k_1 = k_2 = .025 \) consisting of the trapezoidal scheme for the diffusion with time step of \(.2\) and backward Euler scheme for the reaction with time step of \(.004\). On the left of figure 5, we show a numerical solution that exhibits nonphysical oscillations that developed after some time. On the right, we show plots of the error versus time steps at different times. There is a critical time step above which the instability develops. Moreover, changing the space discretization does not improve the accuracy. In [16], it is demonstrated that a finer spatial discretization for a constant time step size leads to significantly more error in the long time solution.

**Figure 5.** Left: Typical instability arising from multiscale operator splitting applied to Brusselator problem. Solution is shown at time 80. Right: Plots of the error in the \( L_2 \) norm versus time step size at different times.
An accurate \textit{a posteriori} estimate of the error must account for the stability effects arising from operator splitting. The standard approach used to define an adjoint operator for analyzing nonlinear problems based on linearization of a perturbation equation fails. It turns out that the adjoint operators associated with the original problem and the multiscale operator decomposition method are fundamentally different. In the estimate below, this difference takes the form of "residuals" between certain adjoint operators. A practical difficulty with such a result is that solving the adjoint for the fully coupled problem poses the same multiphysics challenges as solving the original forward problem. We therefore develop a new hybrid \textit{a priori - a posteriori} estimate that combines a computable leading order expression obtained using \textit{a posteriori} arguments with a provably higher order bound obtained using \textit{a priori} convergence result.

Note that \( \dot{Y} \) is a consistent numerical solution for an analytic version of the operator splitting and the expression for its error can be estimated using the standard \textit{a posteriori} error analysis. We let \( \varphi_d \) define the adjoint solution associated with the diffusion component (13) and \( \varphi^r \) define the adjoint solution associated with the reaction component (12) satisfying respectively

\[
\begin{aligned}
-\dot{\varphi}_d &= A^T \varphi_d(t), \quad t_n > t \geq t_{n-1}, \\
\varphi_d(t_n^-) &= \psi_n, \quad n = N, \ldots, 1,
\end{aligned}
\]

\[
\begin{aligned}
-\dot{\varphi}^r &= \left( \dot{F}^r(Y^r) \right)^T \varphi^r(t), \quad s_{m,n} > t \geq s_{m-1,n}, \\
\varphi^r(s_{m,n}) &= \psi_{m,n}^r, \quad n = N, \ldots, 1, m = M_n, \ldots, 1,
\end{aligned}
\]

with \( \psi_{M_n,n} = \varphi_{n-1}^d \) and \( \psi_{m,n}^r = \psi_{m,n}^r \) for \( m < M_n \). Thus \( \varphi^r \) is continuous across the internal reaction time nodes \( s_{m,n}, m = 1, \ldots, M_n - 1 \). Here, \( \dot{F}^r(y^r, Y^r) = \int_0^1 F'(sy^r + (1-s)Y^r) ds. \)

The nonlinearity complicates the analysis because the choice of trajectory around which to linearize is not clear. We cannot use the standard approach of linearizing the error representation because of the operator splitting. Instead, we assume that both the original problem and the operator split version have a common solution and we linearize each problem in a neighborhood of this common solution. For example, we assume that \( y = 0 \) is a steady state solution of both problems, which can be achieved by assuming that

\[
\text{Homogeneity Assumption: } F(0) = 0,
\]

and we linearize in a region around 0. In terms of applications to reaction-diffusion problems, there are mathematical reasons for this assumption and it is satisfied in a great many applications. We can modify the analysis to allow for linearization around any known common solution, see [17].

For \( n = 1, \ldots, N \), we define the three adjoint problems. The diffusion problem is simplest because it is linear:

\[
\begin{aligned}
-\dot{\varphi}_d &= A^T \varphi_d(t), \quad t_n > t \geq t_{n-1}, \\
\varphi_d(t_n^-) &= \psi_d^r,
\end{aligned}
\]

It is convenient to let \( \Phi_n^d \) denote the solution operator, so \( \varphi_d(t_n^{-1}) = \Phi_n^d \psi_d^r \). We require two adjoint problems to treat the reaction component. The difference between the problems is the function around which they are linearized,

\[
\begin{aligned}
-\dot{\varphi}_1^r &= F'(Y)^T \varphi_1^r(t), \quad t_n > t \geq t_{n-1}, \\
\varphi_1^r(t_n^-) &= \psi_1^r,
\end{aligned}
\]

\[
\begin{aligned}
-\dot{\varphi}_2^r &= F'(Y^r)^T \varphi_2^r(t), \quad t_n > t \geq t_{n-1}, \\
\varphi_2^r(t_n^-) &= \psi_2^r.
\end{aligned}
\]

If \( \Phi_n^r(z) \) denotes the solution operator for the problem linearized around a function \( z \), then we have \( \varphi_1^r(t_n^{-1}) = \Phi_n^r(Y) \psi_n^d \) and \( \varphi_2^r(t_n^{-1}) = \Phi_n^r(Y^r) \psi_n^r \). We can now prove [17].
Theorem 3.5 A hybrid a posteriori - a priori error estimate

\[
(\tilde{Y}_N - y_N, \psi_N) = \sum_{n=1}^{N} \left( \int_{I_n} (\dot{Y}^d - AY^d, \phi^d - \Pi \phi^d) \, dt + ([Y^d]_{n-1}, \phi^d_{n-1} - \Pi \phi^d_{n-1}) \right) \\
+ \sum_{n=1}^{N} \sum_{m=1}^{M_n} \left( \int_{I_{m,n}} (\dot{Y}^r - F(Y^r), \phi^r - \Pi \phi^r) \, dt + ([Y^r]_{m-1,n}, \phi^r_{m-1,n} - \Pi \phi^r_{m-1,n}) \right) \\
+ \sum_{n=1}^{N} (\tilde{Y}_n - (E_1 + E_2) \psi_n) + O(\Delta t^{q_t+2}) + O(\Delta t \Delta s^{q_s+1}),
\]

where

\[
E_1 = \frac{1}{2} \Delta t_n \left( A^T \mathcal{F}(\tilde{Y}) - \mathcal{F}(Y) A^T \right), \quad \mathcal{F}(Y) = \int_{I_n} \overline{F'(Y)} \, dt, \quad E_2 = \left( \Phi_n^r(\tilde{Y}) - \Phi_n^r(Y^r) \right) \Phi_n^d.
\]

The first expression on the right is the error introduced by the numerical solution of the diffusion component while the second expression is the error introduced by the numerical solution of the reaction component. The third expression on the right measures the effects of operator splitting, where \( E_1 \) is a leading order estimate for the effects of operator splitting while \( E_2 \) accounts for issues arising from the differences in linearizing around the global computed solution as opposed to the solution of the reaction component. Both of these quantities are scaled by the solution itself, so that these effects become negligible when the solution approaches zero. Finally, the remaining terms represent bounds on terms that are not computable but are higher order. In practice, we neglect those terms when computing an estimate.

Example 3.6.

We consider the Brusselator problem (2) with \( \alpha = 2, \beta = 5.45, k_1 = 0.008, k_2 = 0.004 \) and initial conditions \( u_1(x, 0) = \alpha + 0.1 \sin(\pi x) \) and \( u_2(x, 0) = \beta/\alpha + 0.1 \sin(\pi x) \), which yields an oscillatory solution. In this case, the reaction is very mildly unstable, with at most polynomial rate accumulation of perturbations as time passes. We use a 32 node spatial finite element discretization, resulting in a differential equation system with dimension 62. We note that in original form, the reaction terms do not satisfy the requirement \( F(0) = 0 \) so we linearize around the steady state solution \( c \) with \( c_i = \alpha \) for \( i = 1, \ldots, N_c - 1 \) and \( c_i = \beta/\alpha \) for \( i = N_c, \ldots, 2N_c - 2 \), so that \( F(c) = 0 \).

Figure 6 compares the errors computed using \( \Delta t = 0.01 \) and \( M = 10 \) reaction time steps to the hybrid a posteriori error estimates. We show results for \([0, 2]\), when the solution is still in a transient stage, and at \( T = 40 \) when the solution has become periodic. All the results show that the exact and estimated errors are in remarkable agreement.

3.3. Multiscale decomposition of a fluid-solid conjugate heat transfer problem

We next consider the multiscale decomposition solution of the heat transfer problem described in Example 1.1. The weak formulation of (3) consists of computing \( u \in V_F, p \in L^2_0(\Omega_F), T_F \in W_F \) and \( T_S \in W_S \) such that

\[
\begin{cases}
    a_1(u, v) + c_1(u, u, v) + b(v, p) + d(T_F, v) = (f, v), \\
    b(u, q) = 0, \\
    a_2(T_F, w_F) + c_2(u, T_F, w_F) + a_3(T_S, w_S) = (Q_F, w_F) + (Q_S, w_S),
\end{cases}
\]

for all \( v \in V_{F,0}, q \in L^2_0(\Omega_F), w_F \in W_{F,0} \) and \( w_S \in W_{S,0} \), where
Figure 6. Brusselator results. Left: Comparison of errors against the spatial location at $T = 2$.
Middle: Time history of errors at the midpoint location on $[0, 2]$. Right: Comparison of errors
against the spatial location at $T = 40$. The dotted line is the exact error, and the (+) is the
estimated error

\[
a_1(u, v) = \int_{\Omega_F} \mu (\nabla u \cdot \nabla v) \, dx, \quad a_2(T_F, w_F) = \int_{\Omega_F} k_F (\nabla T_F \cdot \nabla w_F) \, dx,
\]
\[
a_3(T_S, \omega_S) = \int_{\Omega_S} k_S (\nabla T_S \cdot \nabla \omega_S) \, dx,
\]
\[
c_1(u, v, z) = \int_{\Omega_F} \rho_0 (u \cdot \nabla) v \cdot z \, dx,
\]
\[
d(T, v) = \int_{\Omega_F} \rho_0 T g \cdot v \, dx,
\]
\[
b(v, q) = -\int_{\Omega_F} (\nabla \cdot v) q \, dx,
\]
\[
c_2(u, T, w) = \int_{\Omega_F} \rho_0 c_p (u \cdot \nabla T) \, w \, dx,
\]
\[
f = \rho_0 (1 + \beta T_0) g,
\]

and

\[
V_F = \{ v \in H^1(\Omega_F) \mid v = g_{u,D} \text{ on } \Gamma_{u,D} \}, \quad V_{F,0} = \{ v \in V_F \mid v = 0 \text{ on } \Gamma_{u,D} \},
\]
\[
W_F = \{ w \in H^1(\Omega_F) \mid w = g_{T,D} \text{ on } \Gamma_{T,D} \}, \quad W_{F,0} = \{ w \in W_F \mid w = 0 \text{ on } \Gamma_{T,D} \},
\]
\[
W_S = \{ w \in H^1(\Omega_S) \mid w = g_{T,S} \text{ on } \Gamma_{T,S} \}, \quad W_{S,0} = \{ w \in W_S \mid w = 0 \text{ on } \Gamma_{T,S} \}.
\]

To discretize, we construct independent locally-quasi-uniform triangulations $T_{F,h}$ and $T_{S,h}$ of
$\Omega_F$ and $\Omega_S$, respectively. We use the piecewise polynomial spaces

\[
V^h_F = \{ v \in V_F \mid v \text{ continuous on } \Omega_F, \ v_i \in P^2(K) \text{ for all } K \in \tau_{F,h} \},
\]
\[
Z^h = \{ z \in Z \mid z \text{ continuous on } \Omega_F, \ z \in P^1(K) \text{ for all } K \in \tau_{F,h} \},
\]
\[
W^h_F = \{ w \in W_F \mid w \text{ continuous on } \Omega_F, \ w \in P^2(K) \text{ for all } K \in \tau_{F,h} \},
\]
\[
W^h_S = \{ w \in W_S \mid w \text{ continuous on } \Omega_S, \ w \in P^2(K) \text{ for all } K \in \tau_{S,h} \},
\]

and the associated subspaces

\[
V^h_{F,0} = \{ v \in V^h \mid v = 0 \text{ on } \Gamma_{u,D} \}, \quad W^h_{F,0} = \{ w \in W^h_F \mid w = 0 \text{ on } \Gamma_{T,F,D} \},
\]
\[
W^h_{S,0} = \{ w \in W^h_S \mid w = 0 \text{ on } \Gamma_{T,S,D} \text{ and } w = 0 \text{ on } \Gamma_I \},
\]

where $P^q(K)$ denotes the space of polynomials of degree $q$ on an element $K$. Note that $W^h_{S,0}$
is different from $W^h_{F,0}$ in an important way since $\Gamma_{T,S,D}$ does not include $\Gamma_I$. We let $\pi_V$, $\pi_{W_F}$,
$\pi_{W_S}$, and $\pi_Z$ be projections into $V^h_F$, $W^h_F$, $W^h_S$ and $Z^h$ respectively. We also use $\pi_{W_F}$ and $\pi_{W_S}$
to denote projections into $W^h_F$ and $W^h_S$ respectively along the interface $\Gamma_I$.

To compute a stable solution of the fluid equations, we choose $V^h_F$ and $Z^h$ to be the Taylor-
Hood finite element pair satisfying the discrete inf-sup condition. In general, the convergence of
the iteration defined by this algorithm depends on the values of $k_S$ and $k_F$ along the interface
and the geometry of each region. Often, a “relaxation” is used to help improve convergence
properties, which affects the analysis [18, 19].
\begin{algorithm}
\textbf{Algorithm 3} Multiscale Decomposition Method for Conjugate Heat Transfer

\begin{algorithm*}
\begin{align}
&k = 0 \\
&\textbf{while} (\|T^{(k)}_S - \pi ST^{(k)}_F\|_{\Gamma_I} > TOL) \textbf{do} \\
&\quad k = k+1 \\
&\quad \text{Compute } T^{(k)}_S \in W^h_S \text{ such that } T^{(k)}_S = \pi W T^{(k-1)}_F \text{ along the interface } \Gamma_I \text{ and } \\
&\quad \quad a_3(T^{(k)}_S, w) = (Q_S, w), \quad \forall w \in W^h_S, \\
&\quad \text{Compute } u^{(k)}_h \in V^h_F, p^{(k)}_h \in Z^h \text{ and } T^{(k)}_F \in W^h_F \text{ such that} \\
&\quad \quad a_1(u^{(k)}_h, v) + c_1(u^{(k)}_h, u^{(k)}_h, v) + b(v, p^{(k)}_h) + d(T^{(k)}_F, v) = (f, v), \quad \forall v \in V^h_F, \\
&\quad \quad b(u^{(k)}_h, q) = 0, \quad \forall q \in Z^h, \\
&\quad \quad a_2(T^{(k)}_F, w) + c_2(u^{(k)}_h, T^{(k)}_F, w) = (Q_F, w) - (k_S(n \cdot \nabla T^{(k)}_S, w))_{\Gamma_I}, \quad \forall w \in W^h_F. \\
&\textbf{end while}
\end{align}
\end{algorithm*}

3.3.1. Description of an a posteriori error analysis We define the errors $e_u = u - u^{(k)}_h$, $e_p = p - p^{(k)}_h$, $e_{TF} = T_F - T^{(k)}_F$, and $e_{TS} = T_S - T^{(k)}_S$. The adjoint problem for the quantity of interest $(\psi, e) = (\psi_u, e_u) + (\psi_p, e_p) + (\psi_{TF}, e_{TF}) + (\psi_{TS}, e_{TS})$ is

\begin{equation}
\begin{cases}
-\mu \Delta \phi + \tau_1^\phi(\phi) + \nabla z + \tau_2^u(\theta_F) = \psi_u, & x \in \Omega_F, \\
-\nabla \cdot \phi = \psi_p, & x \in \Omega_F, \\
-k_F \Delta \theta_F + \tau_2^\theta(\theta_F) + \rho_0 \beta (g \cdot \phi) = \psi_{TF}, & x \in \Omega_F, \\
\theta_F = \theta_S, & x \in \Gamma_I, \\
k_F(n \cdot \nabla \theta_F) = k_S(n \cdot \nabla \theta_S), & x \in \Omega_S, \\
-k_S \Delta \theta_S = \psi_{TS}, & x \in \Omega_S.
\end{cases}
\end{equation}

Here, we have used the linearizations

$$
\tau_1^\phi(\phi) = \frac{1}{2} \rho_0 \nabla (u + u_h) \cdot \phi - \frac{1}{2} \rho_0 (u + u_h) \cdot \nabla \phi - \frac{1}{2} \rho_0 (\nabla \cdot (u + u_h)) \phi,
$$

$$
\tau_2^u(\theta) = \frac{1}{2} \rho_0 c_p \nabla (T + T_h) \theta,
\tau_2^\theta(\theta) = -\frac{1}{2} \rho_0 c_p (u + u_h) \cdot \nabla \theta - \frac{1}{2} \rho_0 c_p (\nabla \cdot (u + u_h)) \theta.
$$

We solve (19) numerically using an iterative operator decomposition approach as for the forward problem independently of the forward iterations. In [18, 19], we derive estimates that require only adjoint solutions of the two component problems.

To write out the a posteriori error representation, we introduce an additional projection $\pi^0_W : H^2 \rightarrow W^h_S$ defined such that for any node $x_i$, $\pi^0_W \theta_S(x_i) = \pi W \theta_S(x_i)$ for $x_i \notin \Gamma_I$, and 0 otherwise. We can now prove the following theorem [19].
Theorem 3.7

\[ (\psi, e) = (f, \phi - \pi V \phi) - a_1(u_h^{(k)}, \phi - \pi V \phi_1) - c_1(u_h^{(k)} T^{(k)}, \phi - \pi V \phi) \]
\[ - b(\phi - \pi V \phi, p_h) - d(T^{(k)}, \phi - \pi V \phi) - b(u_h^{(k)} z, z - \pi \nabla z) \]
\[ + (Q, \theta_F - \pi_{WF} \theta_F) - a_2(T^{(k)}, \theta_F - \pi_{WF} \theta_F) \]
\[ - c_2(u_h^{(k)} T^{(k)} T^{(k)} \theta_F - \pi_{WF} \theta_F) + (Q, \theta_S - \pi_{WS} \theta_S) - a_3(T^{(k)}, \theta_S - \pi_{WS} \theta_S) \]
\[ + (T^{(k)} - \pi_{ST} T^{(k)} \theta_F - \pi_{ST} \theta_F) + (\theta_S T^{(k)} - T^{(k)} \theta_F, \theta_S (\theta_S - \pi_{WS} \theta_S)) \]
\[ + (k_S (\theta_S - \pi_{WS} \theta_S)) \Gamma_I + (\theta_S, \pi_{WS} \theta_S) - a_3(T^{(k)}, \pi_{WS} \theta_S). \]

The contributions to the error are threefold.

- Equations (20)–(21) represents the contribution of the discretization error arising from each component solve.
- Equation (22) represents the contribution from the iteration.
- The first term in (23) represents contribution of the transfer error, while the remaining terms represent the contribution arising from projections between two different discretizations.

Example 3.8.

For the flow past a cylinder, we solve the steady nondimensionalized Boussinesq equations in the fluid domain and the nondimensional heat equation in the solid domain. To simulate the flow of water past a cylinder made from stainless steel, we set the dimensionless constants \( Pr = 6.6 \) and \( k_R = 30 \), and choose the inflow velocity and the temperature gradient so that \( Re = 75, \ Tc = 495, \ Fr = 0.001, \ Ra = 50 \). The temperature gradient is imposed by setting different temperatures along the top and bottom boundaries, with a linear temperature gradient on the inflow boundary, and an adiabatic condition on the outflow boundary.

We show results for two quantities of interest. The first is the temperature in a small region in the wake, located approximately one channel width downstream of the center of the cylinder and 1/4 of a channel width below the upper wall. The second is temperature at a small region in the center of the cylinder. In each case, we base adaptivity on an element tolerance of \( 1 \times 10^{-8} \).

We show the final adaptive meshes for the flow and solid in figure 7. For the first quantity of interest, the flow mesh is most refined near the region of interest and upstream of the region of interest, locating more elements between the cylinder and the top wall than the cylinder and the bottom wall since the flow advecting heat to the region of interest passes above rather than below the cylinder. For the solid, the mesh is highly refined along the top in order to increase the accuracy of the normal derivative that is computed in the solid and used as a boundary condition in the fluid computation. Evidently, the normal derivatives elsewhere on the interface have less of an influence on the first quantity of interest. For the second quantity of interest, the mesh is highly refined upstream of the cylinder. We note that the refinement downstream of the cylinder corresponds closely to the recirculation region, and the mesh refinement is slightly asymmetric about the midplane of the channel due to the asymmetric initial mesh. The mesh in the solid is refined uniformly near the boundary, reflecting the fact that the error in the finite element flux makes a significant contribution to the error in the quantity of interest.

3.3.2. Loss of order and flux correction  The meshes shown in figure 7 are highly refined near the interface. This result reflects the fact that there is significant error in the numerical flux passed between the components. It turns out that this pollutes the entire computation, so that overall the method loses an entire order of accuracy.
Figure 7. Upper: Final adaptive meshes in the fluid and solid when the quantity of interest is the temperature in a small region in the wake above the cylinder. Lower: Final adaptive meshes in the fluid and solid when the quantity of interest is the temperature in a small region in the center of the solid.

One way to compensate for the loss of order is by refining the mesh locally near the interface. Another way is to compute the particular information, in this case the flux on the interface, more accurately. It turns out that we can adapt a post-processing technique called flux correction developed originally by Wheeler [20] and Carey [21, 22] to recover boundary flux values with increased accuracy.

We denote the set of elements in $\tau_{S,h}$ that intersect the interface boundary by $\tau_{I}^{S,h} = \{ K \in \tau_{S,h} | \Gamma \neq \emptyset \}$, and we consider the corresponding finite element space $\Sigma_{h} = \{ v \in P^{2}(K) | v(\eta_{i}) = 0 \text{ if } \eta_{i} \notin \Gamma \}$, where $\{ \eta_{i} \}$ denotes the nodes of element $K$.

The degrees of freedom correspond to the nodes on the boundary. We compute $\sigma^{(k)} \in \Sigma_{h}$ satisfying

$$-(\sigma^{(k)}), v)_{I} = (Q_{S}, v) - a_{3}(T_{S,h}^{(k)}), v), \text{ for all } v \in \Sigma_{h},$$

where $T_{S,h}^{(k)}$ solves (17). Since the dimension of the problem scales with the number of nodes on a boundary, it is relatively inexpensive to solve.

In the algorithm, (18) is replaced by the following: Compute $u_{h}^{(k)} \in V_{F}^{h}, p_{h}^{(k)} \in Z^{h}$ and $T_{F,h}^{(k)} \in W_{F}^{h}$ such that

$$\begin{cases}
    a_{1}(u_{h}^{(k)}, v) + c_{1}(u_{h}^{(k)}, u_{h}^{(k)}, v) + b(v, p_{h}^{(k)}) + d(T_{F,h}^{(k)}, v) = (f, v), \\
    b(u_{h}^{(k)}, q) = 0, \\
    a_{2}(T_{F,h}^{(k)}, w) + c_{2}(u_{h}^{(k)}, T_{F,h}^{(k)}, w) = (Q_{F}, w) - (\sigma^{(k)}, w)_{I}, \forall v \in V_{F,0}^{h}, q \in Z^{h}, w \in W_{F,0}^{h}.
\end{cases}$$

Using the recovered boundary flux leads to a cancelation of the “transfer error” term in the error representation formula (23), which is the source of the loss of order. We can prove that using the recovered flux recovers the expected cubic order of convergence; see [19].

**Example 3.9.** The recovered accuracy is easily demonstrated by considering the adapted meshes produced by the modified algorithm. We show the final adaptive meshes for the solid in figure 8. There is no mesh refinement near the boundaries, indicating that the flux error is no longer dominant.
4. Conclusion

Multiphysics, multiscale models present significant challenges in terms of computing accurate solutions and for estimating the error in information computed from numerical solutions. In this paper, we describe recent advances in extending the techniques of a posteriori error analysis to multiscale operator decomposition solution methods. While the particulars of the analysis vary considerably with the problem, several key ideas underlie a general approach to treat operator decomposition multiscale methods that is under development. We explain these ideas in the context of three specific examples.

In this paper, we have minimized the effects arising from the solution of nonlinear and/or fully coupled systems by carefully choosing the models and results that are discussed. Referring back to figure 1, we generally expect multiscale operator decomposition to require a number of iterations between the physical components. This raises additional issues that are discussed in [18, 19, 23, 17, 24].

Acknowledgments

This work was supported in part by DOE (DE-FG02-04ER25620, DE-FG02-05ER25699, DE-FC02-07ER54909), NASA (NNG04GH63G), NSF (DMS-0107832, DMS-0715135, DGE-0221595003, MSPA-CSE-0434354, ECCS-0700559), INL (00069249), LLNL (B573139) and SNL (PO299784).

References

[1] Prigogine I and Lefever R 1968 J. Chem. Phys 48 1695–1700
[2] Marchuk G I, Agoshkov V I and Shutyaev V P 1996 Adjoint equations and perturbation algorithms in nonlinear problems (Boca Raton: CRC Press) ISBN 0-8493-2871-3
[3] Lanczos C 1997 Linear Differential Operators (Dover Publications)
[4] Capecchi D 1997 Sensitivity and Uncertainty Analysis: Theory vol I (Chapman & Hall/CRC)
[5] Marchuk G I 1995 Adjoint equations and analysis of complex systems (Kluwer)
[6] Eriksson K, Estep D, Hansbo P and Johnson C 1996 Computational differential equations (Cambridge: Cambridge University Press) ISBN 0-521-56312-7; 0-521-56738-6
[7] Eriksson K, Estep D, Hansbo P and Johnson C 1995 Acta numerica, 1995 Acta Numer. (Cambridge: Cambridge Univ. Press) pp 105–158
[8] Estep D, Larson M G and Williams R D 2000 Mem. Amer. Math. Soc. 146 viii+109 ISSN 0065-9266
[9] Becker R and Rannacher R 2001 Acta Numerica 1–102
[10] Giles M and Süli E 2002 Acta Numerica 145–236
[11] Bangerth W and Rannacher R 2003 Adaptive Finite Element Methods for Differential Equations (Birkhauser Verlag)
[12] Paraschivoiu M, Peraire J and Patera A 1997 Comput. Methods Appl. Mech. Engrg. 150 289–312
[13] Barth T J 2004 A-Posteriori Error Estimation and Mesh Adaptivity for Finite Volume and Finite Element Methods (Lecture Notes in Computational Science and Engineering vol 41) (New York: Springer)
[14] Carey V, Estep D and Tavener S 2006 SNUM To appear
[15] Estep D and Stuart A M 2002 Math. Comp. 71 1075–1103 (electronic) ISSN 0025-5718
[16] Ropp D L, Shadid J N and Ober C C 2004 J. Comput. Phys. 194 544–574 ISSN 0021-9991

Figure 8. Left: final adaptive mesh in the solid when the quantity of interest is the temperature in a small region in the wake above the cylinder. Right: final adaptive mesh in the solid when the quantity of interest is the temperature in a small region in the center of the solid.
[17] Ginting V, Estep D, Shadid J and Tavener S 2006 *SINUM* To appear
[18] Estep D, Tavener S and Wildey T 2006 *SINUM* To appear
[19] Estep D, Tavener S and Wildey T 2008 *JCP* Submitted
[20] Wheeler M 1974 *Numer. Math.* 2 99–109
[21] GF Carey SS Chow M S 1985 *Comp. Meth. in Applied Mech. and Engr.* 50 107–120
[22] Carey G 1982 *Comp. Meth. in Applied Mech. and Engr.* 35 1–14
[23] Carey V, Estep D and Tavener S 2008 *In preparation*
[24] Ginting V, Estep D and Tavener S 2008 *IMA J. Numer. Analysis* Submitted