A Posteriori Finite Element Output Bounds with Adaptive Mesh Refinement: Application to a Heat Transfer Problem in a Three Dimensional Rectangular Duct [*]

Hae-Won Choi [†] and Marius Paraschivoiu [‡]

Abstract

Numerical simulations based on an *a posteriori* finite element bound method with adaptive mesh refinement are presented for the three dimensional convection–diffusion equation. The bound method provides relevant, quantitative, inexpensive, and rigorous lower and upper bounds to the output on a very fine discretization ("truth" discretization) at a cost close to the coarse mesh calculation ("working" discretization). To achieve a desired bound gap (i.e., difference between upper and lower bounds) at the lowest cost, an adaptive mesh refinement technique is used to refine the mesh only where needed. An optimal stabilization parameter is also applied to improve the sharpness of the bound gap. In this paper, the output of a heat transfer problem in a rectangular duct with a given velocity field is investigated. The average temperature at one section of the duct is bounded for a given inlet temperature and heat flux. For this problem, the adaptive mesh refinement strategy provides the same bound gap with only half the number of elements required by an uniform mesh refinement strategy. The hybrid flux calculations on the coarse mesh introduced for the domain decomposition approach are compared with

[*] accepted for publication in Computer Methods in Applied Mechanics and Engineering

[†] Department of Mechanical and Industrial Engineering, University of Toronto. E-mail: haewon@mie.utoronto.ca

[‡] Corresponding author, Department of Mechanical and Industrial Engineering, University of Toronto. E-mail: marius@mie.utoronto.ca

hybrid flux calculations on the fine mesh to analyze the contribution of the hybrid flux to the bound gap.

1 Introduction

The obvious limitations of analytical solutions for complicated problems force engineers to use numerical or experimental tools. For a large number of applications, numerical simulations provide a fast, inexpensive and flexible alternative to experiments. As a consequence, numerical simulations are an important technology for engineering applications. The main parameter of an approximation method is the discretization size δ. For a given approximation domain and function space, the accuracy of the approximation of the solution is closely related to the discretization size δ. In this study the finite element method is used. As the number of finite elements is increased (i.e., the discretization size δ is decreased), the accuracy of numerical approximation is improved and the requirement of CPU time is increased (which means the computational cost is increased). Therefore an engineer faces a trade–off between computational cost and numerical accuracy. A coarse "working" mesh ($\delta = H$) approximation is relatively inexpensive but generates a solution which is not sufficiently accurate. A fine "truth" mesh ($\delta = h$) approximation is sufficiently accurate (i.e., $\Theta_h(x) \approx \Theta(x)$) but is very expensive, possibly unfeasible, in terms of computational resources.

Numerical methods are exploited in two different categories: analysis and design. In analysis the entire field such as velocity, temperature, or displacement is relevant. For design and more general optimization, only a few parameters are important. A design problem exploits an input–output relationship between design variables and performance requirements. The performance can be an output or a combination of outputs which are functionals of field solutions obtained from ordinary or partial differential equations. These functionals can either be linear or nonlinear. In the context of design and optimization, the final goal of the numerical simulation is not the field solution, but rather to minimize a quantitative design output value.

The technique adopted in this paper does not aim at finding the field solution, but rather seeks to calculate bounds for an output of interest which is derived from the solution of a PDE (Partial Differential Equation). Recently, an *a posteriori* error estimation technique has been extended to address error metrics more closely relevant to engineering design. It is assumed that the quantity of interest in engineering design is not the field variable or the error in the energy norm, but rather the output of the system performance which reflects specific goals and objectives of the design and optimization problem [12, 16, 17]. The idea behind the bound method is to replace a fine mesh output (s_h "truth" output) by relevant, quantitative, inexpensive, and rigorous bounds to s_h (i.e., lower and upper bounds). The bound method offers precise and reliable information at a cost of the same order as the coarse mesh calculation. In design problems, engineers will be interested in bounds with a specific bound gap. By adapting the coarse mesh, sharper bounds can be obtained. The goal is to improve the bound method by using adaptive refinement to bound the fine mesh output (i.e., s_h "truth" output) and thereby provide the "truth" validation.

The key idea of adaptive mesh refinement techniques is the definition of a local error associated with the error to contributions from each element. Adaptive mesh refinement methods accomplish computational cost reduction by varying the mesh size inside the computational domain based on the local error indicator. In this work, adaptive techniques are exploited in a slightly different context: the method searches the ideal coarse mesh to be used as subdomains for bound calculations. The objective of this paper is to develop an error estimator for three dimensional problems, to construct a methodology for adaptive mesh refinement on tetrahedron, to calculate the bounds to an output of interest and to analyze hybrid flux calculations.

The bound method in this paper is based upon a Lagrangian with a modified energy objective where the constraints are the finite element error equation and the inter–subdomain continuity requirement of the error field. The lower and upper bounds are then derived by evoking the dual maximum–minimum problem for appropriately chosen candidate Lagrange multipliers. The bound calculation is composed of two main steps: first, several global computations on the coarse "working" mesh ($\mathcal{T}_H$); second, local decoupled computations on the fine "truth" mesh ($\mathcal{T}_h$).

The bound method is an application of the quadratic–linear duality theory proposed in [1, 5] to an augmented Lagrangian. On one hand, Ladevèze procedure used to approximate the inter–subdomain connectivity in earlier work is only adequate for two-dimensional space [1, 5]. On the other hand, the "flux free" method suggested in [7] may become expensive in three space dimensions. Herein the finite element tearing and interconnecting (FETI) procedure is implemented to extend the method to three-dimensional problems. This domain decomposition method is ideal to approximate the inter–subdomain connectivity (i.e., hybrid flux) for the bound method. The FETI procedure was first introduced as an effective parallel solver for structural problems and is well established in [3, 4]. The FETI procedure reformulated for the bound method for outputs of a three-dimensional convection-diffusion equation is adopted as proposed in [2, 9].

The Lagrangian chosen in this work allows general error measures, such as non-linear functional outputs, and general equations, in particular non-symmetric operators [8, 12, 10, 9]. For more information about *lower* and *upper bounds* for outputs of coercive partial differential equations, the interested reader should refer to [8, 12, 10, 13, 11] for one and two-dimensional problems and to [2, 9] for three-dimensional problems. The bound method has been extended to address a wide range of problems including the Helmholtz and Burgers equations [15] and Stokes equations [8, 13, 11]. Furthermore, a general technique has been developed to address functional outputs of the incompressible Navier-Stokes equations [6].

In this work, the *a posteriori* error estimation procedure [2, 14] is combined with the three-dimensional bound extension to obtain an optimal local subdomain decomposition for the bound method. The adaptive mesh refinement technique based on local bound–gap error indicators is applied as suggested in [2, 14, 6]. Previous local bound–gap error indicators were only constructed for a two-dimensional space [14, 6]. In this paper, the local bound gap error indicator is extended to a convection-diffusion problem in a three-dimensional domain.

The outline of the remainder of this paper is as follows. In Section 2, an *a posteriori* error estimation is introduced for the convection-diffusion equation. The FETI procedure for the new Lagrangian formulation of the bound method is derived. In Section 3, an adaptive mesh refinement technique based on local error indicators is presented. In Section 4, numerical results are reported for the coarse mesh hybrid flux calculations. Finally in Section 5, the hybrid flux is analyzed by calculating more accurate hybrid fluxes directly on the fine mesh.

2 Bound Procedure

In this section, an augmented Lagrangian that permits the calculation of the error estimator is presented. The constraints of this Lagrangian are the finite element error equation and the modified inter–subdomain continuity condition. This section contains the bound method reformulated for this new Lagrangian using the FETI procedure for convection-diffusion equation in three space dimensions. The formulation is constructed without any proof of the bounding properties. For the complete description and proofs of the bounding properties, the reader should refer to [7].

2.1 Problem Statement

The convection-diffusion equation is solved to obtain the temperature field in a rectangular duct with a given laminar velocity field known analytically. The adaptive mesh refinement technique is applied to reduce the bound gap and therefore improve the validated prediction for the output.

2.1.1 Governing Equation

The convection-diffusion equation in three space dimensions are used to describe the behavior of temperature Θ in a rectangular duct flow. The temperature, Θ, satisfies

$$-\frac{\partial}{\partial x_i}(\alpha\frac{\partial\Theta}{\partial x_i}) + U_i\frac{\partial\Theta}{\partial x_i} = f \ \text{ in } \Omega, \ i = 1, 2, 3, \tag{1}$$

with boundary conditions

$$\begin{aligned}
\Theta &= \Theta_i, \quad \text{on } \Gamma_1, && (2)\\
\Theta &= 0, \quad \text{on } \Gamma_2 \text{ and } \Gamma_4, && (3)\\
\alpha\frac{\partial\Theta}{\partial n} &= q_j, \ j = 1, ..., 4, \quad \text{on } \Gamma_3, && (4)\\
\frac{\partial\Theta}{\partial n} &= 0, \quad \text{on } \Gamma_5, && (5)
\end{aligned}$$

where α is the thermal diffusivity and Ω is a bounded domain in $\mathbf{R}^3$ in which Γ_1 is an inhomogeneous Dirichlet boundary, Γ_2 and Γ_4 are homogeneous

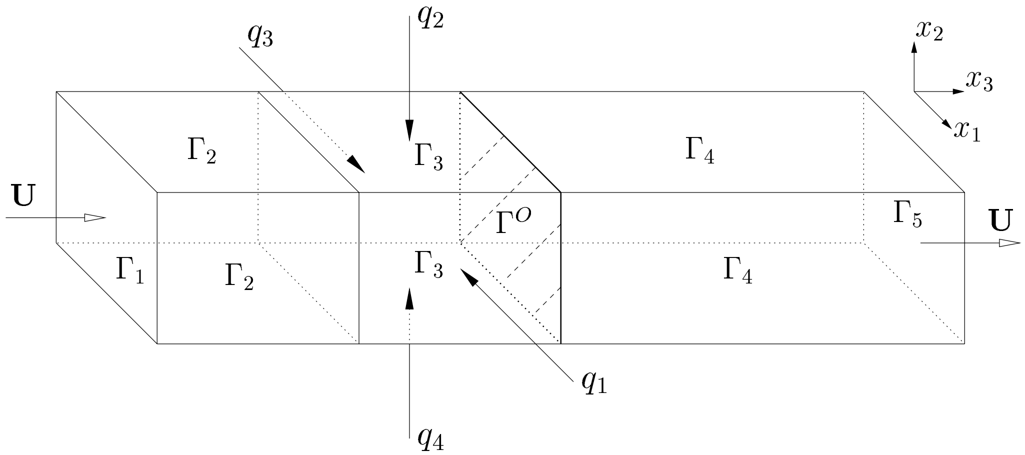

Figure 1: Duct geometry

Dirichlet boundaries, Γ_3 is an inhomogeneous Neumann boundary, and Γ_5 is a homogeneous Neumann boundary.

For this problem $\mathbf{x} = (x_1, x_2, x_3)$ has corresponding unit vectors $\hat{x}_1, \hat{x}_2, \hat{x}_3$. The domain Ω is the rectangular duct, $]0,1[\times]0,1[\times]0,4[$ with outer sides of duct denoted by $\Gamma_j,\ j = 1, ..., 5$, as shown in Figure 1. To avoid singularities on each side of this geometry, a sinusoidal distribution of inlet temperature and heat flux is given. The inlet temperature is given as $\Theta_i = \sin(\pi x_1) \times \sin(\pi x_2)$ and the heat flux is given as $q_{1,3} = 0.5 \times \sin(\pi x_2) \times \sin[\pi(x_3 - 1)]$ and $q_{2,4} = 0.5 \times \sin(\pi x_1) \times \sin[\pi(x_3 - 1)]$. The force term $f = 0$, which assumes no heat source inside the volume.

The variational weak form of the governing equation is as follows: Find $\Theta \in \mathcal{H}^1(\Omega)$ such that

$$\int_\Omega \alpha \frac{\partial v}{\partial x_j}\frac{\partial \Theta}{\partial x_j} + vU_j \frac{\partial \Theta}{\partial x_j}\, dV \; = \; \int_{\Gamma_3} vq\, dA, \quad \forall v \in \mathcal{H}^1_0(\Omega), \tag{6}$$

where dV is a differential volume element, dA is a differential area element, and $\mathcal{H}^1(\Omega)$ denotes the Hilbert space of functions that are square integrable and derivatives that are square integrable over Ω. The right hand side of equation (6) is an inhomogeneous Neumann boundary term arising from the boundary of left hand side.

2.1.2 Output of Interest

The particular linear functional that is investigated herein is the average temperature on a specific slice (denoted as Γ^O) of the domain Ω. Figure 1 indicates the section Γ^O which is $]0,1[\times]0,1[$ at $x_3 = 2.0$.

The average temperature output linear functional, denoted by s, is simply written as

$$s = \ell(\Theta) = \frac{1}{|\Gamma^O|} \int_{\Gamma^O} \Theta \, dA, \tag{7}$$

where $|\Gamma^O|$ is the area of the surface Γ^O. In this case, the value of $|\Gamma^O|$ equals one.

2.1.3 Velocity Field

The velocity distribution is prescribed as $\mathbf{U} = (U_1, U_2, U_3)$ and is obtained analytically. It is assumed that the secondary flow in the rectangular duct is negligible, that is velocities in x_1 and x_2 directions are negligible if they are compared with velocity in x_3-direction. The velocity field is laminar and it is driven by a negative pressure gradient.

The velocity in the x_3-direction has a parabolic distribution and the analytical solution for a rectangular duct flow is formulated by

$$\begin{aligned} U_1 = U_2 &= 0, \\ U_3(x_1, x_2, x_3) &= -\frac{2}{\mu}\frac{dp}{dx_3}\left[\frac{1}{4}\left\{\frac{1}{4} - \left(x_1 - \frac{1}{2}\right)^2\right\}\right. \\ &\quad \left. -2\sum_{n=0}^{\infty} \frac{(-1)^n}{\lambda_n^3} \frac{\cosh \lambda_n \left(x_2 - \frac{1}{2}\right)}{\cosh \frac{\lambda_n}{2}} \cos \lambda_n \left(x_1 - \frac{1}{2}\right)\right], \end{aligned} \tag{8}$$

where

$$\lambda_n = 2\left(n\pi + \tfrac{\pi}{2}\right).$$

Note that, the velocity depends on the coordinates x_1 and x_2, μ (fluid viscosity), and $\frac{dp}{dx_3}$ (pressure gradient in the x_3-direction). For the analytical solution procedure, the reader should refer to [2].

2.2 Finite Element Spaces

Two tetrahedral discretizations of the computational domain Ω are considered: the coarse "working" or design H-mesh, $\mathcal{T}_H$, consisting of K_H elements T_H; and the fine "truth" h-mesh, $\mathcal{T}_h$, consisting of K_h elements T_h. The h-mesh tetrahedron $\mathcal{T}_h$ is a refinement of $\mathcal{T}_H$.

Regular piecewise–linear continuous finite element subspaces are associated to each of these meshes,

$$X_H = \{v \in \mathcal{H}^1(\Omega) \mid v|_{T_H} \in \mathbf{P}_1(T_H),\ \forall T_H \in \mathcal{T}_H\}, \quad (9)$$
$$X_h = \{v \in \mathcal{H}^1(\Omega) \mid v|_{T_h} \in \mathbf{P}_1(T_h),\ \forall T_h \in \mathcal{T}_h\}, \quad (10)$$

where $\mathbf{P}_1(T_\delta)$ denotes the space of linear polynomials over T_δ (where $\delta = H$ and $\delta = h$ respectively). $\mathcal{H}^1(\Omega)$ is the usual Hilbert space of function v such that $v \in L^2(\Omega)$ and $\nabla v \in L^2(\Omega)$ where $L^2(\Omega)$ is the space of square-integrable functions.

Additional "discontinuous subdomain" spaces $\hat{X}_H$ and $\hat{X}_h$ are needed,

$$\hat{X}_H = \{v \in L^2(\Omega)\ \ v|_{T_H} \in \mathcal{H}^1(T_H),\ \forall T_H \in \mathcal{T}_H\}, \quad (11)$$
$$\hat{X}_h = \{v \in L^2(\Omega)\ \ v|_{T_H} \in \mathcal{H}^1(T_H),\ \forall T_H \in \mathcal{T}_H\}. \quad (12)$$

Let $\mathcal{E}(\mathcal{T}_H)$ and $\mathcal{E}(\mathcal{T}_h)$ denote the set of open faces in the tetrahedron $\mathcal{T}_H$ and $\mathcal{T}_h$. Then the spaces of functions over the element faces γ_H and γ_h are introduced as follows

$$\mathcal{Q}_H = \{y|_{\gamma_H} \in \mathbf{P}_1(\gamma_H),\ \forall \gamma_H \in \mathcal{E}(\mathcal{T}_H), y|_{\gamma_N} = 0\}, \quad (13)$$
$$\mathcal{Q}_h = \{y|_{\gamma_H} \in \mathbf{P}_1(\gamma_H),\ \forall \gamma_H \in \mathcal{E}(\mathcal{T}_h) \cap \mathcal{E}(\mathcal{T}_H), y|_{\gamma_N} = 0\}. \quad (14)$$

It follows that $\mathcal{Q}_H \subset \mathcal{Q}_h \subset \mathcal{H}^{-1/2}(\mathcal{E}(\mathcal{T}_H))$; the functions in these spaces can, of course, be discontinuous.

2.3 Lagrangian Formulation

The Lagrangian formulation is presented here using the stabilization parameter κ. The strategy is to write all variables as linear functions in κ and then derive the bounds as a function in κ. This procedure does not change the bounding theory and the bounds remain rigorous. The optimal stabilization parameter κ^* can therefore be calculated. Note that, in case $\kappa = 1$ the usual, non-optimal bounds are obtained. For the purpose of allowing straightforward implementation, the optimal bound formulation is presented in matrix form.

The Lagrangian is expressed with the stabilization parameter κ and variables $v \in \hat{X}_h$, $\mu \in X_h$, and $q \in \mathcal{Q}_h$ as

$$\begin{aligned}\mathcal{L}^{\pm}(v,\mu,q) \;=\; & \sum_{k=1}^{N_k}\left[\kappa\left(v^{(k)^T}A^{(k)}v^{(k)} - f^{(k)^T}v^{(k)} + v^{(k)^T}L^{(k)}\Theta_H^{(k)}\right) \pm \ell^{(k)^T}\left(\Theta_H^{(k)} + v^{(k)}\right)\right] \\ & + \sum_{k=1}^{N_k}\mu^{(k)^T}\left[L^{(k)}(\Theta_H^{(k)} + v^{(k)}) - f^{(k)}\right] + q^T\sum_{k=1}^{N_k}B^{(k)}v^{(k)}, \qquad (15)\end{aligned}$$

where, for N_k subdomains, $v = \{v^{(1)}, ..., v^{(N_k)}\}$ and $\mu = \{\mu^{(1)}, ..., \mu^{(N_k)}\}$. Also, $A^{(k)}$ is the finite element discretization of the symmetric part of $L^{(k)}$ and $B^{(k)}$ is the sign Boolean matrix which localizes the "jumps" at the interface. $\Theta_H^{(k)}$ is the coarse mesh field solution vector. There are two candidate Lagrangian multipliers, which are the adjoint $\mu^{(k)}$ and the hybrid flux $q^{(k)}$. Note that, $(+)$ sign indicates the lower bound and $(-)$ sign denotes the upper bound, respectively.

By evoking stationarity conditions of the Lagrangian (15) on the coarse mesh, three equations are obtained for $e_H^{\pm} \in \hat{X}_H$, $\psi_H^{\pm} \in X_H$ and $\lambda_H^{\pm} \in \mathcal{Q}_H$, such that

$$\begin{aligned} B_H^T\lambda_H^{\pm} + \kappa\left(2A_He_H^{\pm} - f_H + L_H\Theta_H\right) + {L_H}^T\psi_H^{\pm} \pm \ell_H &= 0, & (16)\\ L_H\Theta_H + L_He_H^{\pm} - f_H &= 0, & (17)\\ {B_H}^Te_H^{\pm} &= 0. & (18)\end{aligned}$$

2.4 Adjoint Calculation

The H-mesh adjoint calculation is considered here. From equations (17) and (18), it can be derived that $e_H^{\pm} = 0$ and $L_H\Theta_H = f_H$. For the continuous space X_H, $B_H^T\lambda_H^{\pm} = 0$. Therefore equation (16) is reformulated as

$$ {L_H}^T\psi_H + \ell_H = 0. \qquad (19)$$

Note that, the adjoint can be evaluated as $\psi_H^{\pm} = \pm\psi_H$ and equation (19) is independent of the stabilization parameter κ.

2.5 Hybrid Flux Calculation

Knowing the adjoint and applying $e_H^{\pm} = 0$ in (16), the hybrid flux can be evaluated: Find $\lambda_H^{\pm} \in \mathcal{Q}_H$ such that

$$B_H^T \lambda_H^{\pm} = \kappa \left(f_H - L_H \Theta_H \right) \mp \ell_H - {L_H}^T \psi_H^{\pm}. \tag{20}$$

Alternatively,

$$\begin{aligned} B_H^T \lambda_{0H} &= -\ell_H - {L_H}^T \psi_H, & (21) \\ B_H^T \lambda_{1H} &= f_H - L_H \Theta_H, & (22) \end{aligned}$$

and then set $\lambda_H^{\pm} = \pm \lambda_{0H} + \kappa \lambda_{1H}$. The right hand side of equations (21) and (22) are denoted as the residual of the adjoint and the field solution, respectively. Note that, in equations (21) and (22), B_H^T is singular for redundant constraints and therefore (21) and (22) are not solvable.

If the FETI method is applied, equations (21) and (22) become

$$\begin{aligned} 2\, A_H^{(k)} e_{0H}^{(k)} + {B_H^{(k)}}^T \lambda_{0H} &= -\ell_H^{(k)} - {L_H^{(k)}}^T \psi_H^{(k)}, \quad k = 1, ..., N_k, \\ \sum_{k=1}^{N_k} {B_H^{(k)}}^T e_{0H}^{(k)} &= 0. \end{aligned} \tag{23}$$

Similarly,

$$\begin{aligned} 2\, A_H^{(k)} e_{1H}^{(k)} + {B_H^{(k)}}^T \lambda_{1H} &= f_H^{(k)} - L_H^{(k)} \Theta_H^{(k)}, \quad k = 1, ..., N_k, \\ \sum_{k=1}^{N_k} {B_H^{(k)}}^T e_{1H}^{(k)} &= 0. \end{aligned} \tag{24}$$

Note that, equations (23) and (24) can be reformulated into an interface problem for the FETI procedure.

2.6 The FETI Procedure

The FETI approach is applied to calculate the hybrid flux on the coarse "working" mesh $\mathcal{T}_H$. Equations (21) and (22) can now be treated as interface problems using the FETI procedure.

The superscript (k) denotes each local subdomain problem $(1 \leq k \leq N_k)$. Defining $q_{0H}^{(k)}$ and $q_{1H}^{(k)}$ as

$$\begin{aligned} q_{0H}^{(k)} &= -\ell_H^{(k)} - {L_H^{(k)}}^T \psi_H^{(k)}, & (25) \\ q_{1H}^{(k)} &= f_H^{(k)} - L_H^{(k)} \Theta_H^{(k)}, & (26) \end{aligned}$$

and denoting $R_H^{(k)} = \text{Ker}\left(A_H^{(k)}\right)$. Then equations (23) and (24) are reformulated as

$$\begin{aligned} K_H^{(k)} e_{0H}^{(k)} + {B_H^{(k)}}^T \lambda_{0H} &= q_{0H}^{(k)}, \quad k = 1, ..., N_k, \\ \sum_{k=1}^{N_k} B_H^{(k)} e_{0H}^{(k)} &= 0, \end{aligned} \tag{27}$$

and

$$\begin{aligned} K_H^{(k)} e_{1H}^{(k)} + {B_H^{(k)}}^T \lambda_{1H} &= q_{1H}^{(k)}, \quad k = 1, ..., N_k, \\ \sum_{k=1}^{N_k} B_H^{(k)} e_{1H}^{(k)} &= 0, \end{aligned} \tag{28}$$

where $K_H^{(k)} = 2\, A_H^{(k)}$. Combining all subdomain equations and enforcing solvability lead to

$$\sum_{k=1}^{N_k} B_H^{(k)} e_{0H}^{(k)} = \sum_{k=1}^{N_k} B_H^{(k)} \left[{K_H^{(k)}}^+ \left(q_{0H}^{(k)} - {B_H^{(k)}}^T \lambda_{0H} \right) + R_H^{(k)} a_{0H}^{(k)} \right] = 0, \tag{29}$$

and similarly

$$\sum_{k=1}^{N_k} B_H^{(k)} e_{1H}^{(k)} = \sum_{k=1}^{N_k} B_H^{(k)} \left[{K_H^{(k)}}^+ \left(q_{1H}^{(k)} - {B_H^{(k)}}^T \lambda_{1H} \right) + R_H^{(k)} a_{1H}^{(k)} \right] = 0, \tag{30}$$

where $a_H^{(k)}$ is the set of amplitudes that specifies the contribution of the $R_H^{(k)}$ to the solution. These coefficients can be determined by requiring that each subdomain problem be mathematically solvable — i.e., each floating subdomain be self-equilibrated — which is

$$\begin{aligned} {R_H^{(k)}}^T \left(q_{0H}^{(k)} - {B_H^{(k)}}^T \lambda_{0H} \right) &= 0, \\ {R_H^{(k)}}^T \left(q_{1H}^{(k)} - {B_H^{(k)}}^T \lambda_{1H} \right) &= 0, \\ k = 1, ..., N_k. \end{aligned} \tag{31}$$

Rearranging equations (29), (30) and (32) gives

$$\begin{aligned}\sum_{k=1}^{N_k} B_H^{(k)} K_H^{(k)^+} B_H^{(k)^T} \lambda_{0H} - \sum_{k=1}^{N_k} B_H^{(k)} R_H^{(k)} a_{0H}^{(k)} &= \sum_{k=1}^{N_k} B_H^{(k)} K_H^{(k)^+} q_{0H}^{(k)}, \\ -R_H^{(k)^T} B_H^{(k)^T} \lambda_{0H} &= -R_H^{(k)^T} q_{0H}^{(k)}, \qquad (32) \\ k &= 1, ..., N_k,\end{aligned}$$

and

$$\begin{aligned}\sum_{k=1}^{N_k} B_H^{(k)} K_H^{(k)^+} B_H^{(k)^T} \lambda_{1H} - \sum_{k=1}^{N_k} B_H^{(k)} R_H^{(k)} a_{1H}^{(k)} &= \sum_{k=1}^{N_k} B_H^{(k)} K_H^{(k)^+} q_{1H}^{(k)}, \\ -R_H^{(k)^T} B_H^{(k)^T} \lambda_{1H} &= -R_H^{(k)^T} q_{1H}^{(k)}, \qquad (33) \\ k &= 1, ..., N_k.\end{aligned}$$

This leads to the FETI interface problem, i.e., equations (32) and (33), in matrix form are:

$$\begin{bmatrix} F_H & -G_H \\ -G_H^T & 0 \end{bmatrix} \begin{bmatrix} \lambda_{0H} \\ a_{0H} \end{bmatrix} = \begin{bmatrix} d_{0H} \\ -b_{0H} \end{bmatrix}, \qquad (34)$$

and

$$\begin{bmatrix} F_H & -G_H \\ -G_H^T & 0 \end{bmatrix} \begin{bmatrix} \lambda_{1H} \\ a_{1H} \end{bmatrix} = \begin{bmatrix} d_{1H} \\ -b_{1H} \end{bmatrix}, \qquad (35)$$

where each of these terms is given by

$$\begin{aligned} F_H &= \sum_{k=1}^{N_k} B_H^{(k)} K_H^{(k)^+} B_H^{(k)^T}, \qquad (36) \\ G_H &= \left[B_H^{(1)} R_H^{(1)} \ \ \ldots \ \ B_H^{(N_k)} R_H^{(N_k)} \right], \qquad (37) \\ a_{0H} &= \left[a_{0H}^{(1)} \ \ldots \ a_{0H}^{(N_k)} \right], \\ a_{1H} &= \left[a_{1H}^{(1)} \ \ldots \ a_{1H}^{(N_k)} \right], \qquad (38) \\ d_{0H} &= \sum_{k=1}^{N_k} B_H^{(k)} K_H^{(k)^+} q_{0H}^{(k)}, \\ d_{1H} &= \sum_{k=1}^{N_k} B_H^{(k)} K_H^{(k)^+} q_{1H}^{(k)}, \qquad (39) \end{aligned}$$

$$\begin{aligned} b_{0H} &= \left[R_H^{(1)^T} q_{0H}^{(1)} \ \ldots \ R_H^{(N_k)^T} q_{0H}^{(N_k)}\right], \\ b_{1H} &= \left[R_H^{(1)^T} q_{1H}^{(1)} \ \ldots \ R_H^{(N_k)^T} q_{1H}^{(N_k)}\right], \end{aligned} \qquad (40)$$

where $K_H^{(k)^+}$ is a generalized inverse of $K_H^{(k)}$. The constraints $G_H^T \lambda_{0H} = b_{0H}$ and $G_H^T \lambda_{1H} = b_{1H}$ ensure that $\left(-\ell_H^{(k)} - L_H^{(k)^T} \psi_H{}^{(k)}\right) \in$ range $A_H^{(k)}$ and $\left(f_H^{(k)} - L_H^{(k)} \Theta_H^{(k)}\right) \in$ range $A_H^{(k)}$ for all $k = 1, ..., N_k$.

The FETI method iterates on λ_{0H} and λ_{1H}, given an initial λ_{0H}^0 and λ_{1H}^0 which satisfy the constraints $G_H^T \lambda_{0H} = b_{0H}$ and $G_H^T \lambda_{1H} = b_{1H}$. Note that in equations (34) and (35) only the right hand side is different.

The FETI algorithm can be regarded as a two-step preconditioned conjugate gradient method to solve the interface problem and can be summarized as in [9, 3]:

1. Initialize

$$\begin{aligned} \hat{\lambda}_H^0 &= G_H (G_H^T G_H)^{-1} b_H \\ w^0 &= P_H^T (d_H - F_H \hat{\lambda}_H^0) \end{aligned}$$

2. Iterate $n = 0, 1, ...$ until convergence

$$\begin{aligned} y^n &= P_H \tilde{F}_H^{-1} w^n \\ p^n &= y^n - \sum_{i=0}^{n-1} \frac{y^{nT} F_H p^i}{p^{iT} F_H p^i} p^i \\ \eta^n &= \frac{y^{nT} w^n}{p^{nT} F_H p^n} \\ \hat{\lambda}_H^{n+1} &= \hat{\lambda}_H^n + \eta^n p^n \\ w^{n+1} &= w^n - \eta^n P_H^T F_H p^n \end{aligned}$$

where P_H is a projection operator and $\tilde{F}_H^{-1}$ is a preconditioner.

2.7 Local Neumann Problems

Both Lagrangian candidate multipliers, i.e., ψ_H and $\lambda_H^{\pm}$ (more precisely, λ_{0H} or λ_{1H}), and the field variable Θ_H are linearly interpolated onto the fine mesh $\mathcal{T}_h$, i.e., $\psi_H \to \psi_h$, $\lambda_H^\pm \to \lambda_h^\pm$ (precisely, $\lambda_{0H} \to \lambda_{0h}$ or $\lambda_{1H} \to \lambda_{1h}$), and $\Theta_H \to \hat{\Theta}_h$.

The minimizers of $\mathcal{L}^\pm(v, \psi_h^\pm, \lambda_h^\pm)$, which is $\hat{e}_h^\pm \in \hat{X}_h$, will satisfy the following equations:

$$\begin{aligned} 2\kappa\, A_h^{(k)} (\hat{e}_h^\pm)^{(k)} &= \kappa \left(f_h^{(k)} - L_h^{(k)} \hat{\Theta}_h^{(k)} \right) \mp \ell_h^{(k)} - {L_h^{(k)}}^T \psi_h^{\pm(k)} - {B_h^{(k)}}^T \lambda_h^\pm, \\ & \quad k = 1, ..., N_k. \end{aligned} \tag{41}$$

Alternatively, in terms of $\hat{e}_{0h}$ and $\hat{e}_{1h}$

$$2\, A_h^{(k)} \hat{e}_{0h}^{(k)} = -\ell_h^{(k)} - {L_h^{(k)}}^T {\psi_h}^{(k)} - {B_h^{(k)}}^T \lambda_{0h}, \tag{42}$$

$$\begin{aligned} 2\, A_h^{(k)} \hat{e}_{1h}^{(k)} &= f_h^{(k)} - L_h^{(k)} \hat{\Theta}_h^{(k)} - {B_h^{(k)}}^T \lambda_{1h}, \\ & \quad k = 1, ..., N_k, \end{aligned} \tag{43}$$

where $\hat{e}_h^\pm = \pm\frac{1}{\kappa}\hat{e}_{0h} + \hat{e}_{1h}$.

2.8 Bound Calculation

The lower and upper bounds can be derived from the Lagrangian in (15) or its equivalent simplified form

$$(s_h)_{UB} = s_H + \kappa \sum_{k=1}^{N_k} (\hat{e}_h^-)^{(k)^T} A_h^{(k)} (\hat{e}_h^-)^{(k)}, \tag{44}$$

$$(s_h)_{LB} = s_H - \kappa \sum_{k=1}^{N_k} (\hat{e}_h^+)^{(k)^T} A_h^{(k)} (\hat{e}_h^+)^{(k)}. \tag{45}$$

Alternatively, the bounds are

$$\begin{aligned} (s_h)_{UB} &= s_H - 2 \sum_{k=1}^{N_k} (\hat{e}_{0h}^{(k)})^T A_h^{(k)} \hat{e}_{1h}^{(k)} \\ & \quad + \frac{1}{\kappa} \sum_{k=1}^{N_k} (\hat{e}_{0h}^{(k)})^T A_h^{(k)} \hat{e}_{0h}^{(k)} + \kappa \sum_{k=1}^{N_k} (\hat{e}_{1h}^{(k)})^T A_h^{(k)} \hat{e}_{1h}^{(k)}, \end{aligned} \tag{46}$$

$$\begin{aligned} (s_h)_{LB} &= s_H - 2 \sum_{k=1}^{N_k} (\hat{e}_{0h}^{(k)})^T A_h^{(k)} \hat{e}_{1h}^{(k)} \\ & \quad - \frac{1}{\kappa} \sum_{k=1}^{N_k} (\hat{e}_{0h}^{(k)})^T A_h^{(k)} \hat{e}_{0h}^{(k)} - \kappa \sum_{k=1}^{N_k} (\hat{e}_{1h}^{(k)})^T A_h^{(k)} \hat{e}_{1h}^{(k)}. \end{aligned} \tag{47}$$

The half bound gap becomes:

$$\triangle(\mathcal{T}_H,\kappa)=\frac{1}{\kappa}\sum_{k=1}^{N_k}(\hat{e}_{0h}^{(k)})^T A_h^{(k)}\hat{e}_{0h}^{(k)}+\kappa\sum_{k=1}^{N_k}(\hat{e}_{1h}^{(k)})^T A_h^{(k)}\hat{e}_{1h}^{(k)}. \tag{48}$$

The output predictor is rewritten in matrix form as

$$(s_h)_{pre}(\mathcal{T}_H)=s_H-2\sum_{k=1}^{N_k}(\hat{e}_{0h}^{(k)})^T A_h^{(k)}\hat{e}_{1h}^{(k)}. \tag{49}$$

Note that, equation (49) is independent of the stabilization parameter κ. The actual calculation of this parameter is presented next.

2.9 Optimal Bounds

The optimal stabilization parameter κ^* is formulated in matrix form as

$$\kappa^*=\sqrt{\frac{\sum_{k=1}^{N_k}\left(\hat{e}_{0h}^{(k)}\right)^T A_h^{(k)}\hat{e}_{0h}^{(k)}}{\sum_{k=1}^{N_k}\left(\hat{e}_{1h}^{(k)}\right)^T A_h^{(k)}\hat{e}_{1h}^{(k)}}}. \tag{50}$$

Note that, the contribution to the optimal stabilization parameter κ^* is evaluated for each subdomain. Then, the optimal bounds are rewritten as

$$\begin{aligned}(s_h)_{UB} &= s_H-2\sum_{k=1}^{N_k}(\hat{e}_{0h}^{(k)})^T A_h^{(k)}\hat{e}_{1h}^{(k)}\\ &\quad+\frac{1}{\kappa^*}\sum_{k=1}^{N_k}(\hat{e}_{0h}^{(k)})^T A_h^{(k)}\hat{e}_{0h}^{(k)}+\kappa^*\sum_{k=1}^{N_k}(\hat{e}_{1h}^{(k)})^T A_h^{(k)}\hat{e}_{1h}^{(k)}, \qquad (51)\\ (s_h)_{LB} &= s_H-2\sum_{k=1}^{N_k}(\hat{e}_{0h}^{(k)})^T A_h^{(k)}\hat{e}_{1h}^{(k)}\\ &\quad-\frac{1}{\kappa^*}\sum_{k=1}^{N_k}(\hat{e}_{0h}^{(k)})^T A_h^{(k)}\hat{e}_{0h}^{(k)}-\kappa^*\sum_{k=1}^{N_k}(\hat{e}_{1h}^{(k)})^T A_h^{(k)}\hat{e}_{1h}^{(k)}, \qquad (52)\end{aligned}$$

and the associated bound gap becomes

$$\triangle(\mathcal{T}_H,\kappa^*)=\frac{1}{\kappa^*}\sum_{k=1}^{N_k}(\hat{e}_{0h}^{(k)})^T A_h^{(k)}\hat{e}_{0h}^{(k)}+\kappa^*\sum_{k=1}^{N_k}(\hat{e}_{1h}^{(k)})^T A_h^{(k)}\hat{e}_{1h}^{(k)}. \tag{53}$$

3 Adaptive Mesh Refinement

In practice, an engineer will be interested in bounds with a specific bound gap. Improving the sharpness of the bound gap is essential to the bound method. The bounds can be improved by uniformly refining the coarse mesh. Nevertheless, this approach is far from optimal. It was shown in the previous section that the bound gap is related to the sum of local quantities. In this section, the procedure of adaptive mesh refinement based on decreasing the value of the local contribution to the bound gap is presented. This technique is validated on a heat transfer problem in a rectangular duct.

3.1 Local Indicators

The adaptive mesh refinement procedure can be implemented on any pair of meshes, $\mathcal{T}_H$ and $\mathcal{T}_h$, which satisfy the requirement that $\mathcal{T}_h$ is a refinement of $\mathcal{T}_H$. Since the cost of computing the bounds is essentially a function of the number of elements K_H in $\mathcal{T}_H$, it is desirable to construct a mesh with the minimum number of tetrahedra that maximize the bound accuracy (minimize the bound gaps). In this section an adaptive algorithm for generating such optimized grids proposed in [14, 6] is extended to three dimensional spaces.

Recall that the half bound gap can be expressed as a sum of elemental contributions, i.e.,

$$\triangle(\mathcal{T}_H) = \sum_{T_H \in \mathcal{T}_H} \triangle_{T_H}(\mathcal{T}_H), \tag{54}$$

where

$$\triangle_{T_H}(\mathcal{T}_H) = \frac{1}{2}\left[\sum_{k=1}^{N_k} (\hat{e}_h^-)^{(k)} A_h^{(k)} (\hat{e}_h^-)^{(k)} + \sum_{k=1}^{N_k} (\hat{e}_h^+)^{(k)} A_h^{(k)} (\hat{e}_h^+)^{(k)}\right]. \tag{55}$$

Note that $\triangle_{T_H}(\mathcal{T}_H)$ is non-negative and can thus be directly interpreted as the contribution to the bound gap from element T_H.

If the optimal stabilization parameter κ^* is used in the bound calculation, the local indicators should be formulated as

$$\triangle(\mathcal{T}_H) = \sum_{T_H \in \mathcal{T}_H} \triangle_{T_H}(\mathcal{T}_H, \kappa^*), \tag{56}$$

where

$$\begin{aligned}\triangle_{T_H}(\mathcal{T}_H,\kappa^*) &= \frac{1}{\kappa^*}\sum_{k=1}^{N_k}\hat{e}_{0h}^{(k)}A_h^{(k)}\hat{e}_{0h}^{(k)}+\kappa^*\sum_{k=1}^{N_k}\hat{e}_{1h}^{(k)}A_h^{(k)}\hat{e}_{1h}^{(k)},\\ &= \sqrt{\frac{\sum_{k=1}^{N_k}\hat{e}_{1h}^{(k)}A_h^{(k)}\hat{e}_{1h}^{(k)}}{\sum_{k=1}^{N_k}\hat{e}_{0h}^{(k)}A_h^{(k)}\hat{e}_{0h}^{(k)}}}\sum_{k=1}^{N_k}\hat{e}_{0h}^{(k)}A_h^{(k)}\hat{e}_{0h}^{(k)}\\ &\quad+\sqrt{\frac{\sum_{k=1}^{N_k}\hat{e}_{0h}^{(k)}A_h^{(k)}\hat{e}_{0h}^{(k)}}{\sum_{k=1}^{N_k}\hat{e}_{1h}^{(k)}A_h^{(k)}\hat{e}_{1h}^{(k)}}}\sum_{k=1}^{N_k}\hat{e}_{1h}^{(k)}A_h^{(k)}\hat{e}_{1h}^{(k)}.\end{aligned} \tag{57}$$

An adaptive mesh refinement strategy inspired by [14, 6] is described. Starting from an initial grid $\mathcal{T}_H^0$, local bound gaps $\{\triangle(\mathcal{T}_H^n),\ n=1,2,3,...\}$ are calculated. Here n denotes the n^{th} refinement such that each tetrahedron $\mathcal{T}_H^n$ is a refinement of the preceding tetrahedron $\mathcal{T}_H^{n-1}$. This approach does not guarantee that $\triangle(\mathcal{T}_H^n) \leq \triangle(\mathcal{T}_H^{n-1})$ for any particular n, but it does ensure that, for a sufficiently large n, $\triangle(\mathcal{T}_H^n) \leq \triangle^{target}$, where $\triangle^{target} > 0$ is a specified positive gap target.

In order to identify the elements in $\mathcal{T}_H^{n-1}$ that need to be refined, first the largest elemental contribution $\triangle_{max}^{n-1}$ to the bound gap $\triangle(\mathcal{T}_H^{n-1})$ can be calculated as

$$\triangle_{max}^{n-1} = \max_{T_H\in\mathcal{T}_H^{n-1}} \triangle_{T_H}(\mathcal{T}_H^{n-1}), \tag{58}$$

and then all elements $T_H \in \mathcal{T}_H^{n-1}$ for which

$$\triangle_{T_H}(\mathcal{T}_H^{n-1}) \geq \beta\,\triangle_{max}^{n-1}, \tag{59}$$

can be selected for refinement. The parameter β controls the cutoff, i.e., identifying elements to be refined at each adaptive cycle ($0 < \beta < 1$). At present, this parameter is specified a *priori*, and is independent of n.

The adaptive mesh refinement process is terminated when the accuracy goal is achieved. In the applications that follow, the accuracy measure is the relative (half) bound gap θ^n, given by

$$\theta^n = \frac{\triangle(\mathcal{T}_H^n)}{(s_h)_{pre}(\mathcal{T}_H^n)}. \tag{60}$$

The adaptive process is thus halted when $\theta^n \leq \theta^{obj}$ where θ^{obj} is the prescribed accuracy. The adaptive process is also stopped when reaching a maximum number of iterations.

	$\mathcal{T}_H^0$	$\mathcal{T}_H^2$	$\mathcal{T}_H^4$	$\mathcal{T}_H^5$	$\mathcal{T}_{H/2}^0$
# of elements	3,000	5,613	12,082	21,445	24,000
# of nodes	756	1,308	2,671	4,501	4,961
$(s_h)_{UB}$	1.2365	1.2812	1.3220	1.3203	1.2559
s_H	0.7055	0.8996	1.0171	1.0622	0.9570
$(s_h)_{LB}$	0.6062	0.8166	0.9423	0.9969	0.9087
$(s_h)_{pre}$ $\pm\triangle(\mathcal{T}_H)$	0.9213 $\pm$ 0.3151	1.0489 $\pm$ 0.2323	1.1321 $\pm$ 0.1898	1.1586 $\pm$ 0.1617	1.0823 $\pm$ 0.1736
θ (%)	34.2	22.1	16.8	13.9	16.0
max $\triangle_{T_H}$	4×10^{-3}	8.6×10^{-4}	2.4×10^{-4}	1.1×10^{-4}	9×10^{-4}

Table 1: Optimal bounds for uniform and adaptive refinement of a structured initial mesh.

3.2 Adaptivity Procedure

The adaptive mesh refinement procedure is presented in this section. The four steps required for the entire procedure are:

1. Calculate on an initial "coarse" mesh $\mathcal{T}_H^0$ the corresponding bound gap $\triangle(\mathcal{T}_H^0)$. Calculate the half bound gap $\triangle_{T_H}$ for each element and find the largest elemental contribution $\triangle_{max}^0$ to the bound gap $\triangle(\mathcal{T}_H^0)$. Tag elements which have a large contribution to the bound gap based on equation (59).

2. Refine the tagged elements from Step 1.

3. Calculate bounds, bound gap, and local indicators using the new coarse mesh.

4. Repeat Step 2 to Step 3 until the desired bound gap is achieved.

4 Numerical Results

In this section, the convection-diffusion equation is investigated for a rectangular duct flow where the thermal diffusivity is 0.1 (i.e., $\alpha = 0.1$). The average temperature over Γ^O is the desired output quantity. A comparison between the uniform and adaptive refinement of a structured initial mesh is reported.

All computations are performed on a single processor with a Pentinum III 933 MHz CPU having 1536M memory. For this problem, all the bound calculations are carried out with a refinement of 4 ($R = 4$). Obviously in practice the refinement should be larger. When $R = 4$, each local subdomain has 35 nodes and 64 tetrahedral elements. In the adaptive process, elements are selected for refinement based on equation (59) for $\beta = 0.4$ for all cycles. The global FETI tolerance is set to 10^{-3} for this case.

Recall that the goal of the adaptive technique is to decrease the bound gap and distribute the local bound gap error more uniformly over the entire domain. Results of the optimal bound method using uniform and adaptive refinement of a structured initial mesh are summarized in Table 1. Table 1 shows that a relative (half) bound gap of 16.8% is achieved with 12,082 elements ($\mathcal{T}_H^4$), while the optimal uniform refinement provides 16.0% with twice the number of elements ($\mathcal{T}_{H/2}^0$ has 24,000 elements). The output $s_h = 1.1692$ on a fine "truth" mesh consisting of $270,641$ degrees-of-freedom (nodes) and $1,536,000$ tetrahedral elements. Note that, the output predictor $(s_h)_{pre}$ for the final adapted mesh $\mathcal{T}_H^5$ equals 1.1586 which is very close to the fine "truth" output ($s_h = 1.1692$). The uniformly refined mesh $\mathcal{T}_{H/2}^0$ leads to an output predictor $(s_h)_{pre}$ equal to 1.0823 which is significantly less accurate.

Table 2 reports the number of elements (%) by half bound gap $\triangle(T_H)$ intervals for an uniform and an adaptive refinement of a structured initial mesh with optimal stabilization parameter κ^*. As more adaptive refinement cycles are performed, the maximum value of the half bound gap $\triangle(T_H)$ is decreased and the range of the bound gap is more uniformly distributed over the entire domain. It is observed that the final adapted mesh $\mathcal{T}_H^5$ (21,445 elements) has 12 elements (0.1%) that have values of bound gap greater than 10^{-4} while the uniformly refined mesh $\mathcal{T}_{H/2}^0$ (24,000 elements) has 317 elements (1.3%). Note that, these 1.3% elements of the uniformly refined mesh $\mathcal{T}_{H/2}^0$ cause the increase of bound gap (bound error).

Figure 2 illustrates the surface of coarse meshes generated using uniform and adaptive refinement of a structured initial mesh. Note that, as more adaptive refinement cycles are performed, the finer grids are concentrated on Γ^O (i.e., $]0,1[\times]0,1[$ at $z = 2$) of the output s (see the adaptively refined meshes in Figure 2). Figure 3 shows the isocontours of a slice of the mesh at

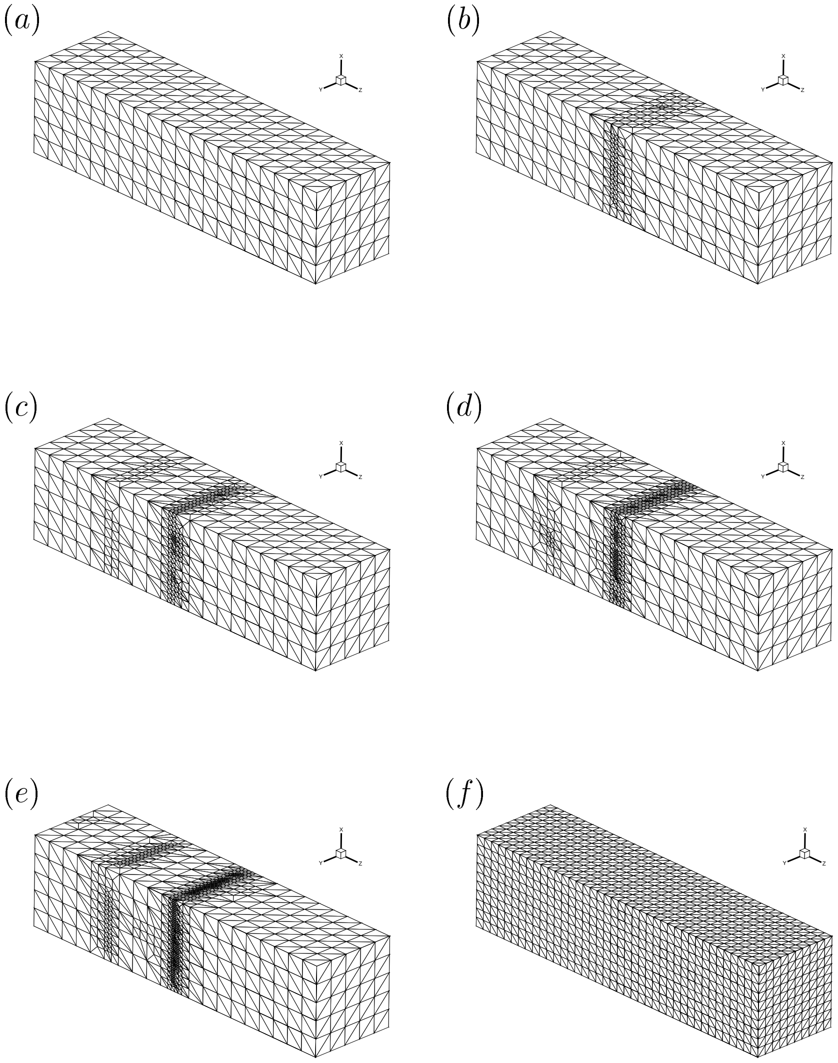

Figure 2: Uniformly and adaptively refined meshes with optimal bound method: (a) $\mathcal{T}_H^0$ has 3,000 elements, (b) $\mathcal{T}_H^2$ has 5,613 elements, (c) $\mathcal{T}_H^3$ has 8,402 elements, (d) $\mathcal{T}_H^4$ has 12,082 elements, (e) $\mathcal{T}_H^5$ has 21,445 elements, and (f) $\mathcal{T}_{H/2}^0$ has 24,000 elements.

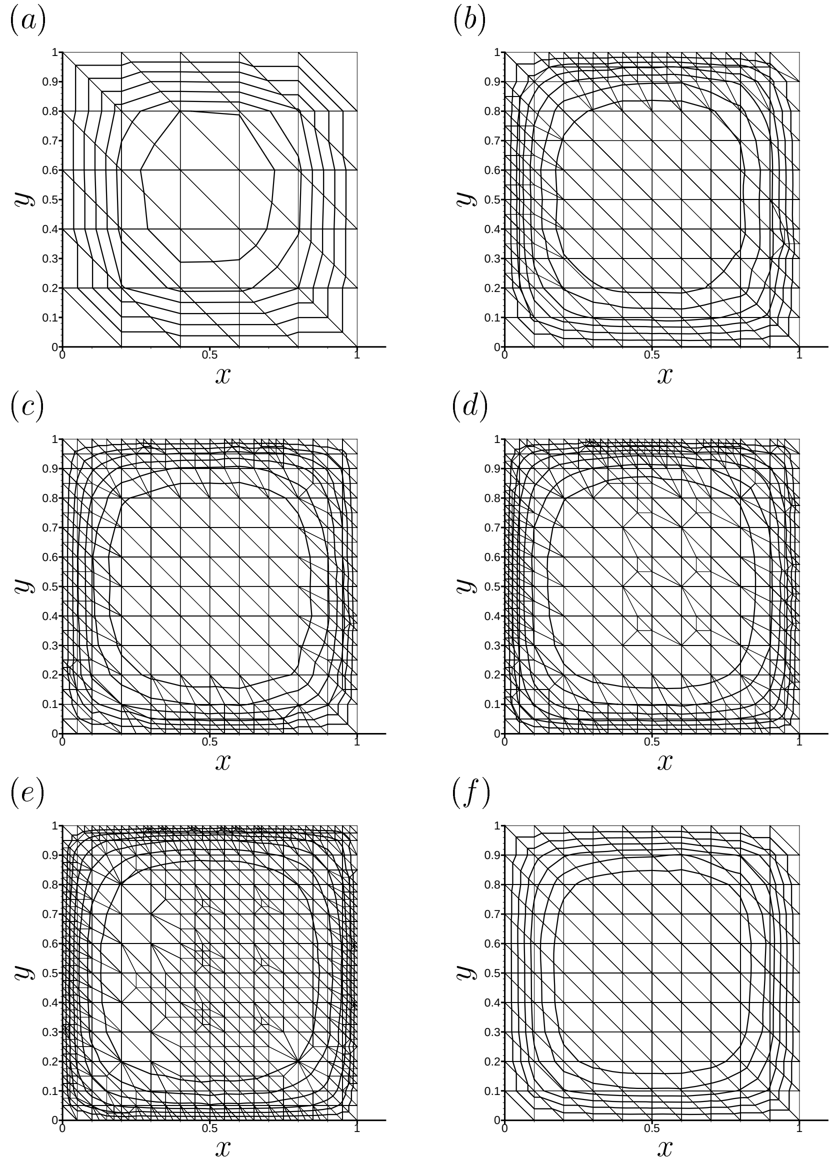

Figure 3: Meshes and Isocontours (0 to 0.6 at intervals of 0.1) of temperature for uniformly and adaptively refined meshes with optimal bound method (slice at z=2.0): (a) $\mathcal{T}_H^0$ has 3,000 elements, (b) $\mathcal{T}_H^2$ has 5,613 elements, (c) $\mathcal{T}_H^3$ has 8,402 elements, (d) $\mathcal{T}_H^4$ has 12,082 elements, (e) $\mathcal{T}_H^5$ has 21,445 elements, and (f) $\mathcal{T}_{H/2}^0$ has 24,000 elements.

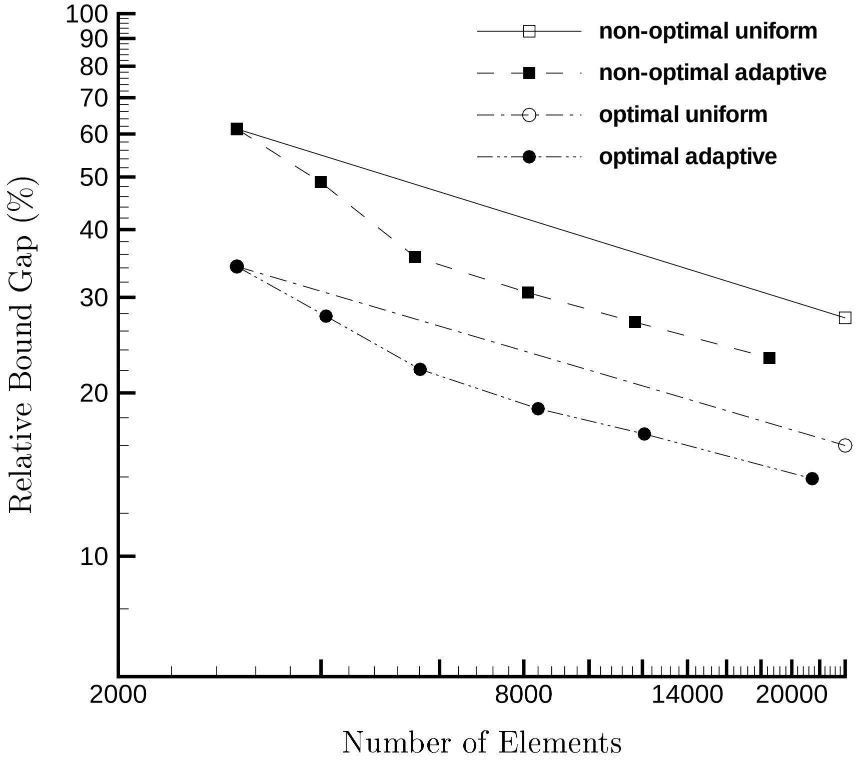

Figure 4: Relative bound gap θ as a function of the number of elements in log-log scale for uniform and adaptive refinement (using the coarse H-mesh hybrid flux calculations).

	$\mathcal{T}_H^0$	$\mathcal{T}_H^5$	$\mathcal{T}_{H/2}^0$
# of elements	3,000	21,445	24,000
$\triangle(T_H) \geq 10^{-3}$	72 (2.4)	0 (0)	0 (0)
$5 \times 10^{-4} \leq \triangle(T_H) < 10^{-3}$	78 (2.6)	0 (0)	34 (0.1)
$10^{-4} \leq \triangle(T_H) < 5 \times 10^{-4}$	338 (11.3)	12 (0.1)	283 (1.2)
$5 \times 10^{-5} \leq \triangle(T_H) < 10^{-4}$	276 (9.2)	307 (1.4)	358 (1.5)
$10^{-5} \leq \triangle(T_H) < 5 \times 10^{-5}$	859 (28.6)	4,268 (19.9)	1,627 (6.8)
$5 \times 10^{-6} \leq \triangle(T_H) < 10^{-5}$	117 (5.9)	3,861 (18.0)	1,273 (5.3)
$10^{-6} \leq \triangle(T_H) < 5 \times 10^{-6}$	164 (5.5)	9,705 (45,2)	6,887 (28.7)
$\triangle(T_H) < 10^{-6}$	1,036 (34.5)	3,292 (15.4)	13,538 (56.4)

Table 2: Number of elements (%) by half bound gap $\triangle(T_H)$ intervals for uniform and adaptive refinement of a structured initial mesh.

$z = 2$ for uniform and adaptive refinements of a structured initial mesh. The value of isocontours ranges from 0 to 0.6 at intervals of 0.1. The isocontours of temperature Θ_H are plotted just for analysis. Recall that the mesh showed in this figure is used to define subdomains where decoupled problems are solved. The temperature solution on this mesh as well as the adjoint field solution are interpolated onto the fine mesh, therefore a more accurate solution on this mesh leads to shaper bounds. As the refinement is performed, the isocontours become smoother. Final isocontours on mesh $\mathcal{T}_H^5$ (21,445 elements) catch the boundary layer near the wall more precisely than that of uniformly refined mesh $\mathcal{T}_{H/2}^0$ (24,000 elements). After four refinements, the method identifies the mesh in the middle region where there is not much refinement required and thereby the sharpness of bounds or relative bound gap is not improved significantly. At this point, the gap is mainly due to the interpolation error of the hybrid flux. This effect is analyzed in the Section 5.

Figure 4 shows the relative bound gap θ with and without the stabilization parameter κ^* as a function of the number of elements over Γ^O using uniform and adaptive refinements of a structured initial mesh. It can be observed from Figure 4 that the adaptive refinement of a structured initial mesh using optimal bound method performs best in terms of relative bound gap than any other method. Because of the interpolation error of hybrid flux on the fine h-mesh, the relative (half) bound gap θ is still large. In the next section the influence of the hybrid flux on the sharpness of bounds and relative (half) bound gap is investigated by calculating the hybrid flux directly on the fine h-mesh.

5 Analysis of the Hybrid Flux Calculations

In this section, the hybrid flux contribution to the bound gap is investigated by calculating the hybrid flux with a good accuracy directly on the fine mesh without interpolation. Previously the hybrid flux was calculated on a coarse H-mesh and then interpolated onto the fine h-mesh, i.e. $\lambda_H \to \lambda_h$. The H-mesh hybrid flux calculations have a computational cost advantage–which means it is inexpensive to approximate the hybrid flux on the fine h-mesh.

The influence of the fine "truth" h-mesh hybrid flux calculations on several results of bound method is explored. The adaptively refined meshes starting from a structured initial mesh are employed for this investigation.

Table 3 reports optimal bounds for adaptive refinement of a structured initial mesh. Table 3 shows a reduction of the relative (half) bound gap from 28.7% to 7.2% by using optimal bound method with adaptive refinement. The final adapted mesh $\mathcal{T}_H^4$ (17,189 elements) gives 7.2% relative bound gap θ. Recall that the relative bound gap θ with the coarse H-mesh hybrid flux calculations achieved only 13.9% on the final adapted mesh $\mathcal{T}_H^4$ (21,445 elements). Note that, the direct calculations of hybrid flux on the fine h-mesh provide almost 7% (6.7 %) reduction in terms of relative (half) bound gap. Nevertheless it is important to point out that $(s_h)_{pre}$ is not necessarily improved. Note that $(s_h)_{pre} = 1.1096$ on a mesh with 17,189 elements and that previously, for the H-mesh hybrid flux calculations, $(s_h)_{pre} = 1.1321$ on a mesh with 12,082 elements.

Table 4 reports the number of elements (%) by bound gap $\triangle(T_H)$ intervals with optimal stabilization parameter κ^* using uniformly refined meshes. The final adapted mesh $\mathcal{T}_H^4$ (17,189 elements) has only 9 elements (0.1%) which have values of local bound gap error greater than 5×10^{-5} Recall that for the previous coarse mesh hybrid flux calculations the final adapted mesh $\mathcal{T}_H^5$ (21,445 elements) has 319 elements (1.5%) which have values of local bound gap error greater than 5×10^{-5}. As the adaptive refinement sequence proceeds, the maximum value of half bound gap $\triangle(T_H)$ is decreased and

	$\mathcal{T}_H^0$	$\mathcal{T}_H^1$	$\mathcal{T}_H^2$	$\mathcal{T}_H^3$	$\mathcal{T}_H^4$
# of elements	3,000	4,499	8,371	12,697	17,189
# of node	756	1,072	1,898	2,803	3,755
$(s_h)_{UB}$	1.1805	1.1823	1.1950	1.1844	1,1892
s_H	0.7055	0.8856	0.9993	1.0280	1.0547
$(s_h)_{LB}$	0.6532	0.8423	0.9611	0.9995	1.0301
$(s_h)_{pre}$	0.9168	1.0123	1.0781	1.0920	1.1096
$\pm\triangle(\mathcal{T}_H)$	$\pm$ 0.2637	$\pm$ 0.1700	$\pm$ 0.1170	$\pm$ 0.0924	$\pm$ 0.0795
θ (%)	28.7	16.8	10.8	8.5	7.2
max $\triangle_{T_H}$	2.9×10^{-3}	6.2×10^{-4}	2.3×10^{-4}	1×10^{-4}	7×10^{-5}

Table 3: Optimal bounds for adaptive refinement of a structured initial mesh.

local bound gap error is more uniformly distributed over the entire domain.

Figure 5 shows adaptively refined meshes. The domain Γ^O (i.e., $]0,1[\times]0,1[$ at $x_3 = 2$) is identified and refined as the adaptive refinement cycle proceeds. Isocontours of temperature (from 0 to 0.6 at intervals) in domain Γ^O are shown in Figure 6 to illustrate how the solution on the coarse mesh is improved. Recall that this solution in interpolated on the fine mesh to calculate the bounds. As the refinement cycle progresses, the isocontours become smoother especially near the wall. Final isocontours for the adapted mesh $\mathcal{T}_H^4$ (17,189 elements) catch the boundary layer near the wall more precisely. As it shown in Figure 5(d) and Figure 5(e), the refined mesh pattern is much more symmetric than the pattern of refined meshes in Figure 3. This indicates the method with fine mesh hybrid flux calculations is much more accurate than with coarse mesh hybrid flux calculations.

The relative bound gap θ as a function of the number of elements for adaptive refinement is reported in Figure 7. Figure 7 shows the difference between the coarse mesh and the fine mesh hybrid flux calculations. It is obvious that a large contribution to the bound gap is associated with the hybrid flux calculation. This indicates that more work in this area may lead to sharper bound gap. Since the computation of the hybrid flux is directly performed on the fine h-mesh, the sharpness of bound gap is improved and the relative (half) bound gap under 10% (7.2%) is achieved in less than 4 cy-

	$\mathcal{T}_H^0$	$\mathcal{T}_H^1$	$\mathcal{T}_H^3$	$\mathcal{T}_H^4$
# of elements	3,000	8,371	12,697	17,189
$\triangle(T_H) \geq 10^{-3}$	74 (2.5)	0 (0)	0 (0)	0 (0)
$5 \times 10^{-4} \leq \triangle(T_H) < 10^{-3}$	60 (2.0)	0 (0)	0 (0)	0 (0)
$10^{-4} \leq \triangle(T_H) < 5 \times 10^{-4}$	259 (8.6)	146 (1.7)	1 (0)	0 (0)
$5 \times 10^{-5} \leq \triangle(T_H) < 10^{-5}$	243 (8.1)	398 (4.7)	148 (1.2)	9 (0.1)
$10^{-5} \leq \triangle(T_H) < 5 \times 10^{-4}$	872 (29.1)	2,452 (29.3)	2,637 (20.8)	2,359 (13.7)
$5 \times 10^{-6} \leq \triangle(T_H) < 10^{-5}$	288 (9.6)	1,740 (20.8)	2,199 (17.3)	2,242 (13.0)
$10^{-6} \leq \triangle(T_H) < 5 \times 10^{-6}$	187 (6.2)	2,405 (28.7)	5,192 (40.9)	7,724 (44.9)
$\triangle(T_H) < 10^{-6}$	1,017 (33.9)	1,230 (14.7)	2,520 (19.8)	4,855 (28.2)

Table 4: Number of elements (%) by half bound gap $\triangle(T_H)$ for adaptively refined meshes.

cles. Nevertheless there is a higher computational cost associated with these calculations. It is investigated here only for analysis. Obviously the sharpness of the bounds can also be improved by performing additional refinement cycles.

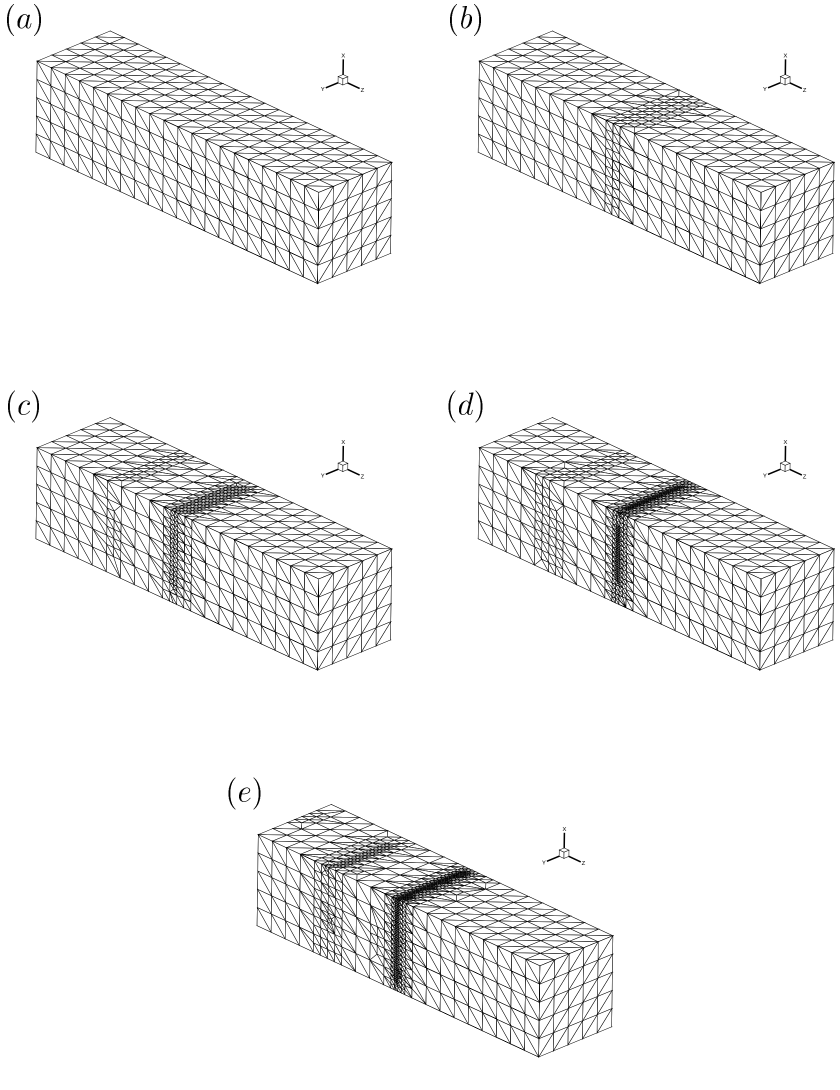

Figure 5: Adaptively refined meshes: (a) $\mathcal{T}_H^0$ has 3,000 elements, (b) $\mathcal{T}_H^1$ has 4,499 elements, (c) $\mathcal{T}_H^2$ has 8,371 elements, (d) $\mathcal{T}_H^3$ has 12,697 elements, and (e) $\mathcal{T}_H^4$ has 17,189 elements.

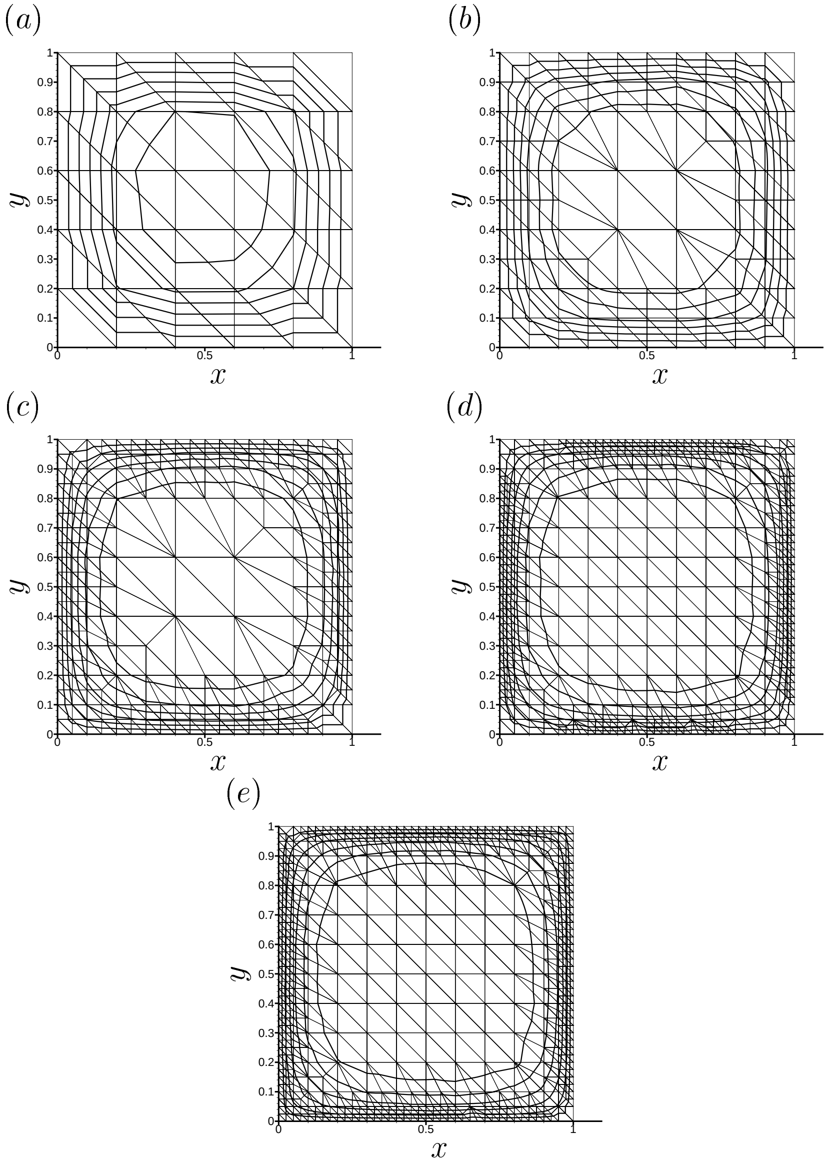

Figure 6: Meshes and Isocontours (0 to 0.6 at intervals of 0.1) of temperature for adaptively refined meshes with optimal bound method (slice at z=2.0): (a) $\mathcal{T}_H^0$ has 3,000 elements, (b) $\mathcal{T}_H^1$ has 4,499 elements, (c) $\mathcal{T}_H^2$ has 8,371 elements, (d) $\mathcal{T}_H^3$ has 12,697 elements, and (e) $\mathcal{T}_H^4$ has 17,189 elements.

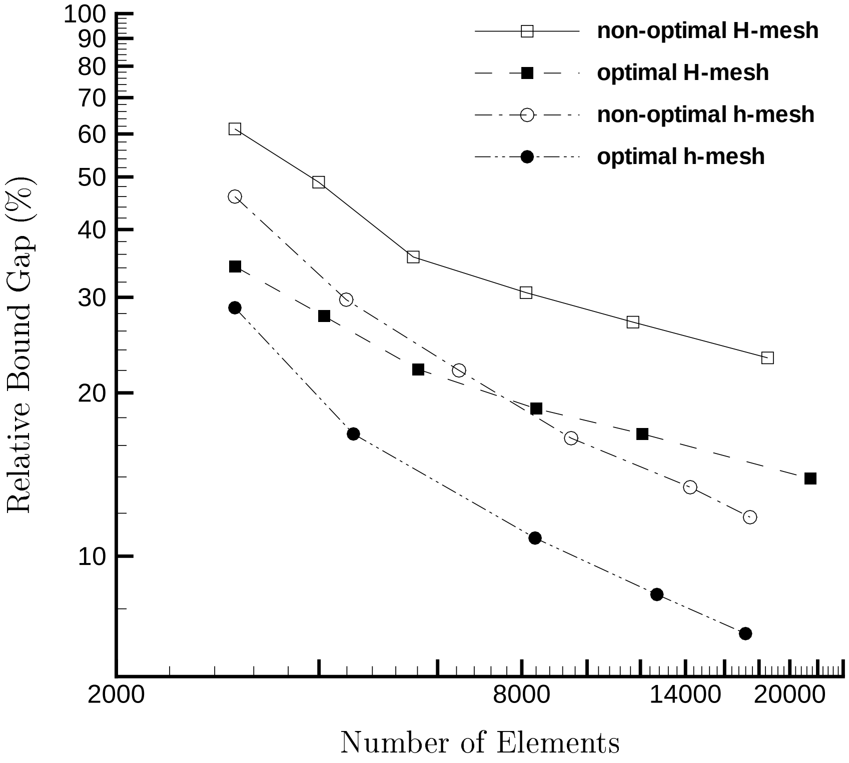

Figure 7: Relative bound gap θ as a function of the number of elements in log-log scale for adaptive refinement: Comparison between coarse H-mesh and fine h-mesh hybrid flux calculations.

References

[1] M. Ainsworth and J. T. Oden. A unified approach to a posteriori error estimation using element residual methods. *Numer. Math.*, 65:23–50, 1993.

[2] H.-W. Choi. *A Posteriori* finite element bounds with adaptive mesh refinement: Application to outputs of the three dimensional Convection–Diffusion equation. Master's thesis, University of Toronto, October 2001.

[3] C. Farhat and F.-X. Roux. A method of finite tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Methods Engineering*, 32:1205–1227, 1991.

[4] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. *Computational Mechanics Advances*, 2(1):1–124, 1994.

[5] P. Ladevèze and D. Leguillon. Error estimation procedures in the finite element method and applications. *SIAM J. Numer. Anal.*, 20:485–509, 1983.

[6] L. Machiels, A. T. Patera, and J. Peraire. *A Posteriori* finite element output bounds for the incompressible Navier–Stokes equations; application to a natural convection problem. *Journal of Computational Physics*, 172:401–425, 2001.

[7] Y. Maday, A. T. Patera, and J. Peraire. A general formulation for a posteriori bounds for output functionals of partial differential equations; application to the eigenvalue problem. *C. R. Acad. Sci.-Math*, 328:823–828, 1999.

[8] M. Paraschivoiu. *A Posteriori finite element bounds for linear–functional outputs of Coericive Partial Differential Equations and of the Stokes Problems.* PhD thesis, Department of Mechanical Engineering, Massachusetts Institute of Technology, October 1997.

[9] M. Paraschivoiu. *A Posteriori* finite element output bounds in three space dimensions using the FETI method. *Comput. Methods Appl. Mech. and Engrg.*, 190:6629–6640, 2001.

[10] M. Paraschivoiu and A. T. Patera. A hierarchical duality approach to bounds for the outputs of Partial Differential Equations. *Comput. Methods Appl. Mech. and Engrg.*, 158:389–407, 1998.

[11] M. Paraschivoiu and A. T. Patera. *A Posteriori* finite element bounds for linear–functional outputs of the Stokes Problem. *Int. J. Numer. Methods Fluids*, 32:823–849, 2000.

[12] M. Paraschivoiu, A. T. Patera, and J. Peraire. *A Posteriori* finite element bounds for linear–functional outputs of elliptic Partial Differential Equations. *Comput. Methods Appl. Mech. and Engrg.*, 150:289–312, 1997.

[13] M. Paraschivoiu, A.T. Patera, Y. Maday, and J. Peraire. Fast bounds for Partial Differential Equation outputs. In J. Borggaard, J. A. Burns, E. M. Cliff, and S. Schreck, editors, *Computational Methods for Optimal Design and Control*, pages 323–360, Birkhäuser, 1998.

[14] A. T. Patera and J. Peraire. Bounds for linear-functional outputs of Coercive Partial Differential equation: local indicators and adaptive refinement. In P. Ladeveze and J. T. Oden, editors, *Proceedings of the Workshop On New Advances in Adaptive Computational Methods in Mechanics*, Cachan, 17-19 September 1997. Elsevier.

[15] A. T. Patera and J. Peraire. Asymptotic *a posteriori* finite element bounds for the outpus of noncoercive problems: the Helmholtz and Burgers equation. *Comput. Methods Appl. Mech. and Engrg.*, 171:77–86, 1999.

[16] N. A. Pierce and M. B. Giles. Adjoint recovery of superconvergent functionals from PDE approximations. *SIAM Review*, 42(2):247–264, 2000.

[17] R. Rannacher. Adaptive Galerkin finite element methods for partial differential equations. *J. Comp. Appl. Math.*, 128:205–233, 2001.