
Hae-Won Choi\textsuperscript{a,2}, Marius Paraschivoiu\textsuperscript{b,1}

\textsuperscript{a}Department of Mechanical and Industrial Engineering, University of Toronto, 5 King’s College Road, Toronto, Ontario, Canada M5S 3G8.

\textsuperscript{b}Department of Mechanical and Industrial Engineering, Concordia University, 1455 de Maisonneuve Blvd. West, Montreal, Quebec, Canada H3G 1M8.

Abstract

Large-scale numerical simulations of the flow and associated transport phenomena governed by the Navier-Stokes and Energy equations are routinely calculated in engineering practice. Nevertheless, the uncertainty due to spatial discretization limits the confidence of practitioners in numerical solutions. An approach to provide information about the accuracy of the quantity of interest is proposed herein. The novel \textit{a posteriori} error estimation technique – the bound method – is based on relaxing Lagrange multipliers that enforces continuity between sub-domains. The method provides fast, efficient, asymptotic but reliable lower and upper bounds to the output of underlying partial differential equations (PDEs). Herein, we highlight the method when applied to outputs of the steady incompressible Navier-Stokes and Energy equations. The bound method in this paper follows the directly equilibrated hybrid-flux approach for the flux calculation between sub-domains and uses the Crouzeix-Raviart ($P_2^+ - P_1$) approximation spaces. To improve the effectiveness of the bound method, an adaptive sub-domain refinement strategy leading to sharper bounds is adopted. A convective heat transfer problem in a series of electronic chip devices is investigated. The novelty of this paper is to present bounds using adaptive domain decomposition for outputs associated to a complex three dimensional field solution of the Navier-Stokes and Energy equations.

Key words: Bound method; Adaptive domain decomposition; Navier-Stokes and Energy equations; Directly equilibrated hybrid-flux; Crouzeix-Raviart finite element.

Corresponding author. Tel: 1-514-848-2424 (ext. 3147); Fax: 1-514-848-3175. E-mail address: paraschi@me.concordia.ca.

Present address: Scientific Computing Division (SCD), National Center for Atmospheric Research
1 Introduction

As computer simulations are used in the engineering practice, it is necessary to have tools to evaluate the accuracy of solutions and more precisely the quantities used in design. In this paper we focus on the novel \textit{a-posteriori} error estimation, termed the bound method, which provides a fast, efficient but reliable lower and upper bounds to the quantity of interest (i.e., the output) of the steady incompressible Navier-Stokes and Energy equations. This paper illustrates this technique for a cooling problem in an array of electronic chip devices.

Numerical solutions of Partial Differential Equations (PDEs) are hindered by computational uncertain and expensive sequencings of meshes for predicting design output quantities. To circumvent this high cost and to ensure high fidelity of Computational Fluid Dynamics (CFD) for design, implicit \textit{a-posteriori} error estimation techniques have recently extended to estimate the error associated with the quantity of interest. Earlier \textit{a-posteriori} error estimation techniques, reviewed and summarized in [32,2,4,20], are based on the self-adjoint problems to provide upper bounds of error in global energy or $L_2$ norms, or on the concept of the error in the constitutive relation [19], or on the error indicator of the equilibrium equations [3], or on the unevenness of the gradient [34]. However, recent \textit{a-posteriori} error estimation techniques deal directly with the error with respect to a specific quantities of interest (i.e., target, goal or output): Becker and Rannacher developed the \textit{‘dual-weighted residual’} (DWR) method [5,6] for error control and mesh optimization in computing local quantities of interest; Paraschivoiu, Peraire and Patera developed the \textit{‘bound method’} [27,26] to calculate bounds to an output; and Prudhomme and Oden suggested the \textit{‘goal-oriented’} technique to compute the error to a goal of the analysis [28,24]. Since these quantities – output or goal – refer to a linear functional of the solution obtained from an underlying partial differential equations (PDEs), such techniques have served as powerful numerical tools to estimate and control the quantity of interest of the engineering design. Although these approaches have some similarities, this paper is focused on the bound method.

For the last few years, a number of extensions and contributions have lead to improvements to the bound method [9,10,12,14,15,22,21,25]. One of the main features of the bound method is the domain decomposition which leads to the computational savings. Choi and Paraschivoiu [13] have compared different approaches, i.e., the \textit{flux-free} and \textit{hybrid-flux} approaches. Firstly, the flux-free approach, developed in [22] and recently extended to three space dimensions [13], uses overlapping elements (or nodal patch). It imposes a partition of unity to remove the boundary condition requirement for each patch but makes local calculations five times more expensive. Secondly, the hybrid-flux approach decomposes the global domain into non-overlapping local subdomains, i.e., subdomains used in the hybrid-flux approach are in fact the coarse $H$-mesh tetrahedra. These subdomains are further refined into self-similar tetrahedra which lead to discrete local Neumann subproblems. The inter-subdomain boundary data (i.e. hybrid-fluxes), required to be self-equilibrated, plays a crucial role in decomposing the global computational domain into a multitude of independently decomposed local subdomains. To evaluate the hybrid-flux, various hybrid-flux evaluation techniques have been developed. For two-dimensional problems, the

---

(NCAR), 1850 Table Mesa Dr., Boulder, CO 80305, USA. E-mail address: haewon@ucar.edu.
method by Ladevèze and Leguillon [19] and the flux-splitting method by Ainsworth and Oden [1] can be used. For three-dimensional problems, Paraschivou [25] reformulated the FETI method for the convection-diffusion equation. Most recently Choi and Paraschivou [14] proposed the directly equilibrated hybrid-flux technique which provides significant computational savings for calculating these fluxes.

The bound method for the Navier-Stokes and Energy equations was first developed by Machiels et al. in [21]. It was based on the general ‘asymptotic’ bound procedure for the hybrid-flux approach to address a natural convection problem in two space dimensions. Nevertheless, our paper follows the Lagrangian formulation for the hybrid-flux approach described in [10,14]. Furthermore, the hybrid-flux calculation utilizes the directly equilibrated hybrid-flux approach described in [14]. The bound procedure is based on two-level computations associated with two different meshes: two sets of global calculations are approximated on the coarse $H$-mesh and a multitude of elemental sub-domain problems are performed on the fine $h$-mesh. Note that due to the non-coerciveness of the equations, these bounds are "asymptotically" rigorous [23,22]. To be more precise, the bounds are rigorous only when the coarse $H$-mesh (the sub-domain mesh) is fine enough. To improve the bound gap and to ensure that the method is in the asymptotic region an adaptive sub-domain refinement strategy is used.

The outline of this paper is as follows. A motivation example for the bound method is presented in Section 2. Section 3 gives the governing equations and their weak formulations of the model problem. Section 4 introduces the Lagrangian formulation and the proof of bounding properties for the Navier-Stokes and Energy equations. The numerical procedure for the bound method is detailed in Section 5. In addition, the adaptive sub-domain procedure is briefly summarized there. Section 6 presents the numerical results for the adaptive sub-domain bound method. Finally, Section 7 concludes this paper.

2 Motivation

To motivate the bound method, we consider the following model problem: a series of electronic chip devices as shown in Figure 1. In a realistic design scenario, engineers need to determine an output such as the mass flow rate, the mean temperature or the drag/lift force in a specific region. One then verifies that the output quantities are within an acceptable design target and a specific design condition is accepted or rejected accordingly. In practice, a number of design situations must be tested. For a single design point, no initial information regarding the ‘ideal’ mesh is known. Therefore, many meshes with different discretization sizes are tested to estimate the accuracy of the desired output.

For our model problem, a forced convection due to a pressure driven flow around an array of two electronic chips at Reynolds number $Re = 100$, Péclet number $Pe = 100$ and heat flux $q = \frac{1}{Pr} \frac{\partial \Theta}{\partial n} = 1$ is simulated. The output selected for this example is the mean temperature in the region $\Gamma^O$ such that $s = \frac{1}{\Gamma^O} \int_{\Gamma^O} \Theta dA$. To provide an estimation to the mean temperature in the specific output region, we first apply a standard Galerkin finite element method to simulate the
uid ow and heat transfer, and then to approximate the output. Without prior information about the spatial discretization needed, typically one starts by using a coarse mesh $\mathcal{T}_{H}^{\text{init}}$ for instance the one shown in Figure 2a. Using a finite element method on this mesh, an output of $s_{H}^{\text{init}} = 13.06$ is calculated within 1933 CPU sec. However, there is a lack of certainty about the value obtained. Does this coarse mesh capture the underlying physics? To increase the fidelity of the simulation, the engineer will use a finer mesh. A relatively fine mesh $\mathcal{T}_{H/3}^{\text{init}}$, for instance having 64,800 tetrahedra and 293,279 d.o.f as shown in Figure 2c, predicts the output of $s_{H/3}^{\text{init}} = 13.38$ at the cost of 482,528 CPU sec or $\approx 5.58$ CPU days. Nevertheless, is this mesh fine enough to be considered "almost" exact? It is clear that by using a standard Galerkin finite element method, engineers face a "trade-off" between numerical accuracy and computational cost.

To address this "trade-off" between numerical accuracy and computational cost for large scale three-dimensional applications, designers can use the bound method. The bound method calculates lower and upper bounds to the output but also provides the ability to sharpen the bound gap.
These bounds are calculated through a two-level hierarchical computation at the fraction of the cost of a standard Galerkin finite element method evaluated on the finer mesh. Using the bound method where each tetrahedral of the initial coarse mesh \( T^\text{init}_H \) is a sub-domain, the calculated bounds are \( s^\text{init}_h = 13.49 \pm 2.492 (18.47\%) \). These values represent the upper and lower values bounding the output associated with the mesh \( T^\text{init}_H/3 \). To sharpen the bounds, the sub-domain refinement strategy can be exploited. The bound method calculated on an uniformly refined sub-domain mesh \( T^\text{init}_H/2 \), as shown in Figure 2b, having 19,200 tetrahedra sub domains gives an output as \( s^\text{init}_h = 13.32 \pm 0.807 (6.06\%) \) at the cost of 140,384 CPU sec \( \approx 1.63 \) CPU days. However, using adaptively refined sub-domains \( T^\text{adapt}_H \) having 18,797 tetrahedra, as shown in Figure 2d, the bound method provides bounds to the output of \( s^\text{adapt}_h = 13.38 \pm 0.716 (5.35\%) \) at the cost of 136,426 CPU sec \( \approx 1.578 \) CPU days. Note that the average value of the bounds (13.38) provides an estimate of the output that is the same as the output calculated on the mesh \( T^\text{init}_H/3 \). Hence, the bound method described in this paper can serve as an efficient design tool for engineering applications. Further numerical investigations will be presented in a later section.

3 Finite Element Background

This section describes the model problem and the finite element function spaces and discretizations. To begin, the strong and weak formulations of the steady incompressible Navier-Stokes and Energy equations are presented. Then, an output linear functional related to an engineering application is introduced. Finally, several finite element spaces used in the bound method associated with the governing equations are defined.

3.1 Governing equations

The governing equations for the model problem, i.e., the steady incompressible Navier-Stokes and Energy equations, are given by their non-dimensional forms as follows:

\[
\frac{\partial u_i}{\partial x_i} = 0, \quad (1)
\]

\[
u_j \frac{\partial u_i}{\partial x_j} = - \frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial}{\partial x_j} \left( \frac{\partial u_i}{\partial x_j} \right) + f_i, \quad (2)
\]

\[
u_j \frac{\partial \Theta}{\partial x_j} = \frac{1}{Pe} \frac{\partial}{\partial x_j} \left( \frac{\partial \Theta}{\partial x_j} \right) \quad \text{in } \Omega \quad i, j = 1, 2, 3, \quad (3)
\]

with boundary conditions for the velocity field

\[
u_i = 0, \quad \text{on } \Gamma^D_0, \quad (4)
\]

\[
u_i = u^D, \quad \text{on } \Gamma^D, \quad (5)
\]
\[ u_i|_{\Gamma_s} = u_i|_{\Gamma_e}, \quad \text{on } \Gamma^p, \quad \text{(6)} \]

and for the temperature field

\[ \begin{align*}
\Theta &= 0, \quad \text{on } \Gamma^D_0, \\
\Theta &= g^D, \quad \text{on } \Gamma^D, \\
\Theta|_{\Gamma_s} &= \Theta|_{\Gamma_e}, \quad \text{on } \Gamma^P, \\
\frac{1}{Pe} \frac{\partial \Theta}{\partial n} &= g^N, \quad \text{on } \Gamma^N, \\
\frac{\partial \Theta}{\partial n} &= 0, \quad \text{on } \Gamma^N_0, \quad \text{(11)}
\end{align*} \]

where \( \Omega \) is a bounded domain in \( \mathbb{R}^3 \) with the boundary \( \partial \Omega \) decomposed into a Dirichlet boundary (\( \Gamma^D \)), a Neumann boundary (\( \Gamma^N \)) for the heat flux generated by the electronic chips and a Periodic boundary (\( \Gamma^P \)) at the left and right sides of the computational domain. Here \( \Gamma_s \) and \( \Gamma_e \) denotes the starting and ending periodic surface, respectively. The boundary with the subscript (0) indicates the homogeneous boundary condition (i.e., \( \Gamma^D_0 \) represents the homogeneous Dirichlet condition and \( \Gamma^N_0 \) represents the homogeneous Neumann condition). \( \Gamma^N_0 \) is representing the insulation condition on the channel walls. Note that there is no boundary condition required for the pressure field in this study. Boundary conditions for other problems involving the Navier-Stokes equations are extensively discussed in [16]. Here \((u_i, p, \Theta)\) are the non-dimensional velocity, pressure, and temperature fields, respectively. \( Re \) represents the Reynolds number defined as \( Re = \frac{U_\infty D_h}{\nu} \) where \( U_\infty, D_h, \) and \( \nu \) denote the reference velocity, the hydraulic diameter, and the kinematic viscosity, respectively and \( f_i \) denotes the body force vector applied in the domain. \( Pe \) denotes the Péclet number defined as \( Pe = \frac{U_\infty D_h}{\alpha} \) where \( \alpha \) is the thermal diffusivity. Also the Péclet number \( Pe \) can also be written as \( Pe = Re \cdot Pr \) where \( Pr \) is the Prandtl number. Note that, the temperature field can only be calculated once the velocity field is known.

### 3.2 Variational weak formulation

Function spaces used in the finite element method are introduced as

\[ \mathcal{H}^1_0(\Omega) = \{ w \in \mathcal{H}^1(\Omega) : w|_{\Gamma^D} = 0 \}, \quad \text{(12)} \]
\[ L^2_0(\Omega) = \{ q \in L^2(\Omega) : \int_\Omega q \, d\Omega = 0 \}, \quad \text{(13)} \]

where \( \mathcal{H}^1 \) is the Hilbert space of functions \( v \) such that \( v \in L^2(\Omega) \) and \( \nabla v \in (L^2(\Omega))^2 \), in which \( L^2(\Omega) \) is the space of square-integrable functions. Then spaces of the essential boundary conditions are introduced as

\[ X = \{ v_i \in \mathcal{H}^1_0(\Omega) \}, \quad \text{(14)} \]
\[ X^D = \{ v_i \in H^1(\Omega) : v_i|_{\Gamma_D} = u^D : v_i|_{\Gamma_\gamma} = v_i|_{\Gamma_\gamma} \}, \]  
(15)

\[ X^\Theta = \{ w \in H^1_0(\Omega) \}, \]  
(16)

\[ X^{\Theta,D} = \{ w \in H^1(\Omega) : w|_{\Gamma_D} = g^D : w|_{\Gamma_\gamma} = w|_{\Gamma_\gamma} \}, \]  
(17)

\[ Q = \{ q \in L^2_0(\Omega) \}, \]  
(18)

where \( i = 1, 2, 3 \) in three space dimensions.

Under these definitions, the variational weak formulation of the continuity equation (1), incompressible Navier-Stokes equations (2), and Energy equations (3) are formulated as follows:

Find \( (u_i, p, \Theta) \in X^D \otimes Q \otimes X^{\Theta,D} \) such that:

\[- \int_{\Omega} q \frac{\partial u_i}{\partial x_i} dV = 0, \quad \forall q \in Q, \]  
(19)

\[ \int_{\Omega} \left( \frac{1}{Re} \frac{\partial u_i}{\partial x_j} \frac{\partial v_i}{\partial x_j} + u_j \frac{\partial u_i}{\partial x_j} v_i - p \frac{\partial v_i}{\partial x_i} \right) dV = \int_{\Omega} f_i v_i dV, \quad \forall v_i \in X, \quad i = 1, 2, 3, \]  
(20)

where \( dV \) and \( dA \) denote the differential volume and area elements, respectively. \( f_i \) is the volumetric force and \( g^N \) is the heat flux, which are assumed to be smooth, i.e., \( f_i \in L^2(\Omega) \) and \( g^N \in L^2(\Omega) \).

3.3 Output linear functionals

The output denoted by \( s \) considered herein is the linear functional of the field solution. The output of interest can be of two types, depending on the field variables, namely a ‘flow’ output and/or a ‘temperature’ output.

The flow output represented by \( s^u \) is the mass flow rate in a volume region \( \Omega^O \) such that

\[ s^u = \ell^{O,u}(u) = \frac{1}{|\Omega^O|} \int_{\Omega^O} u \cdot n_d dV, \]  
(22)

where \(|\Omega^O|\) is the volume of the domain \( \Omega^O \) and \( n_d \) is the unit normal vector with respect to \( x_d \)-direction where \( d \) denotes the specific \( x \)-direction. On the other hand, the temperature output, \( s^\Theta \), is defined by the mean temperature in the area region \( \Gamma^O \) and is written as

\[ s^\Theta = \ell^{O,\Theta}(\Theta) = \frac{1}{|\Gamma^O|} \int_{\Gamma^O} \Theta dA, \]  
(23)
where \( |\Gamma^O| \) is the area of specific output region \( \Gamma^O \). Here the output linear functional is assumed to be smooth such as \( \ell^O \in L^2(\Omega) \). Note that a combined output can be expressed by \( s = s^u + s^\Theta \).

### 3.4 Discrete formulation

For simplicity, the matrix representation of the variational weak form using the finite element spaces defined in Appendix A is written as follows:

Find \( (u_\delta, p_\delta, \Theta_\delta) \in X^D_\delta \otimes Q_\delta \otimes X^{\Theta,D}_\delta \) such that

\[
\frac{1}{Re} A_\delta u_\delta + C_\delta (u_\delta) u_\delta - D^T_\delta p_\delta = f_\delta, \quad i = 1, 2, 3, \tag{24}
\]

\[
\sum_{i=1}^{3} D_i u_\delta = 0, \tag{25}
\]

\[
\frac{1}{Pe} A_\delta \Theta_\delta + C_\delta (u_\delta) \Theta_\delta = g_\delta, \tag{26}
\]

where \( \delta \) represents a discretization size of the finite element. Note that Equation (24) denotes the discrete form of the Navier-Stokes equations, Equation (25) is the discrete form of the continuity equation which plays a role as the incompressibility constraint, and Equation (26) represents the discrete form of Energy equation. The matrix \( A_\delta, C_\delta \) and \( D_\delta \) denote the discrete stiffness, convective, and divergence operators, respectively. \( D^T_\delta \) is represented by ‘transpose’ of \( D_\delta \) and \( f_\delta \) represent the forcing terms and the boundary inhomogeneities. The equivalent block-matrix form of Equations (24)-(26) is given by

\[
\begin{bmatrix}
\frac{A_\delta}{Re} + C_\delta (u_\delta) & 0 & 0 & -D^T_1 & 0 \\
0 & \frac{A_\delta}{Re} + C_\delta (u_\delta) & 0 & -D^T_2 & 0 \\
0 & 0 & \frac{A_\delta}{Re} + C_\delta (u_\delta) & -D^T_3 & 0 \\
D_1 & -D_2 & -D_3 & 0 & 0 \\
0 & 0 & 0 & 0 & \frac{A_\delta}{Pe} + C_\delta (u_\delta)
\end{bmatrix}
\begin{bmatrix}
u_1 \\
u_2 \\
u_3 \\
p \\
\Theta
\end{bmatrix}
= \begin{bmatrix}
f_1 \\
f_2 \\
f_3 \\
p \\
g
\end{bmatrix}. \tag{27}
\]

### 4 Lagrangian Formulation

This section describes the Lagrangian formulation for the Navier-Stokes and Energy equations. Bounding properties are obtained by applying the dual max-min (or inf-sup) theory to the Lagrangian constructed below.
4.1 Augmented Lagrangian

The Lagrangian associated with the bounds is presented as two Lagrangians: one related to the Navier-Stokes equations and one related to Energy equation. The Lagrangian is expressed in terms of the stabilization parameter $\kappa$ to permit sharper bounds.

The Lagrangian for the Navier-Stokes and Energy equations is:

$$
\mathcal{L}^\pm (v_i, q, \mu_i, \lambda_i, t_i, w, \phi, r) \in W_h \otimes Q_h \otimes X_h \otimes V_h \otimes X_h^\Theta \otimes J_h^\Theta
$$

$$
\mathcal{L}^\pm (v_i, q, \mu_i, \lambda_i, t_i, w, \phi, r) = \mathcal{L}^{u, \pm} (v_i, q, \mu_i, \lambda_i, t_i) + \mathcal{L}^{\Theta, \pm} (w, \phi, r), \quad (28)
$$

where the Lagrangian for the Navier-Stokes equations is given by

$$
\mathcal{L}^{u, \pm} (v_i, q, \mu_i, \lambda_i, t_i)\sum_{k=1}^{N_k} \sum_{i=1}^{3} \left[ \kappa \left( T_i^k A^T v_i^k - 2q_i^k T_i^k v_i^k - f_i^k T_i^k v_i^k + v_i^k L^u k_i^k \right) 
- p_i^k T_i^k v_i - q_i^k T_i^k u_i^k \right] + \mathcal{L}^{O, kT} \left( u_i^k + v_i^k \right)
+ \sum_{k=1}^{N_k} \sum_{i=1}^{3} L^u \left( u_i^k + v_i^k \right) T_i^k \left( p_i^k + q_i^k \right) - f_i^k
$$

$$
- \lambda_i^k T_i^k \left( u_i^k + v_i^k \right)
+ \sum_{k=1}^{N_k} t_i^k B^k v_i^k, \quad (29)
$$

and the Lagrangian for the Energy equation is written as

$$
\mathcal{L}^{\Theta, \pm} (w, \phi, r) = \sum_{k=1}^{N_k} \left[ \kappa \left( w_i^k T^A \Theta^k w_i^k - g_i^k T^w w_i^k + w_i^k T^L \Theta^k \Theta^k \right) 
\right]
+ \mathcal{L}^{O, kT} \left( \Theta^k + w_i^k \right)
+ \sum_{k=1}^{N_k} \phi_i^k \left[ L^\Theta, k (\Theta^k + w_i^k) - g_i^k \right]
+ r_i^k \sum_{k=1}^{N_k} B^k w_i^k, \quad (30)
$$

where for $N_k$ subdomains, $v_i = \{ v_{i(1)}^k, \ldots, v_{i(N_k)}^k \}$, $q = \{ q_{(1)}^k, \ldots, q_{(N_k)}^k \}$, $\mu_i = \{ \mu_{i(1)}^k, \ldots, \mu_{i(N_k)}^k \}$, $\lambda = \{ \lambda_{(1)}^k, \ldots, \lambda_{(N_k)}^k \}$, $t_i = \{ t_{i(1)}^k, \ldots, t_{i(N_k)}^k \}$, $w = \{ w_{(1)}^k, \ldots, w_{(N_k)}^k \}$, $\phi = \{ \phi_{(1)}^k, \ldots, \phi_{(N_k)}^k \}$ and $r = \{ r_{(1)}^k, \ldots, r_{(N_k)}^k \}$. The superscript ‘$k$’ denotes each local subdomain, i.e., $1 \leq k \leq N_k$. $A^u k = \frac{1}{\kappa} A$ is a symmetric
matrix associated with the diffusion term of the Navier-Stokes equations, $L^{u,k}$ is the combination of stiffness and convection matrix, i.e., $L^{u,k} = \frac{1}{Re}A^{k} + C^{k}(u)$ where $C^{k}(u)$ is the convection operator, $D^{k}$ represents the divergence matrix, $A'^{k} = \frac{1}{Re}A^{k}$ is the discrete diffusion term in the energy equation $L^{\Theta,k} = \frac{1}{Re}A^{k} + C^{k}(u)$, and $B^{k}$ is the sign Boolean matrix which localizes the “jumps” at the interface. $(u^{k}_{H}, p^{k}_{H}, \Theta^{k}_{H})$ are the coarse mesh velocity, pressure, and temperature vectors, respectively. There are several candidate ‘Lagrange multipliers’, which are Lagrange multipliers for the Navier-Stokes equations, i.e., $(\mu_{i}, \lambda_{i}, t_{i})$ and for the Energy equation, i.e., $(\Phi^{k}, r)$, respectively. Note that, $(\cdot)^{+}$ sign indicates the Lagrangian needed for the lower output bound and $(\cdot)^{-}$ sign denotes the one for the upper output bound. Note that the Navier-Stokes equations cannot be introduced directly as a linear constraint in the Lagrangian. The Navier-Stokes equations are therefore linearized about a coarse mesh solution and this form is used as the constraint. This approach limits the bound to be rigorous only when the size of the course mesh is fine enough to capture the main features of the flow. This is the reason why these bounds are called asymptotic bounds. Nevertheless, in practice it was observed that all the coarse meshes used were fine enough and lead to rigorous bounds.

### 4.2 Proof of the Bounding Properties

Appealing to the dual max-min (or inf-sup) theory [31] applied to the Lagrangian (28) for candidate Lagrange multipliers yields the bounds to the fine mesh output for the coupled Navier-Stokes and energy equations.

The output on the fine mesh can be obtained by solving a constrained minimization problem represented as the saddle of the Lagrangian:

$$\pm s^{u}_{h} = \sup_{(\mu, \lambda, t) \in X_{h} \otimes Q_{h} \otimes F_{h}} \inf_{(v, q) \in W_{h} \otimes Q_{h}} L_{u,\pm}(v, q, \mu, \lambda, t).$$  \hspace{1cm} (31)

By inserting candidate Lagrange multipliers obtained form a coarser mesh into the Lagrangian (29) and appealing the dual max-min (or inf-sup) theory yields the lower and upper bounds such that

$$\left(\pm s^{u}_{h}\right)_{LB} \equiv \inf_{(v, q) \in W_{h} \otimes Q_{h}} L_{u,\pm}(v, q, \hat{\mu}_{h}^{\pm}, \hat{\lambda}_{h}^{\pm}, \hat{r}_{h}^{\pm}) \leq s^{u}_{h},$$

$$s^{u}_{h} \leq -\inf_{(v, q) \in W_{h} \otimes Q_{h}} L_{u,-}(v, q, \hat{\mu}_{h}^{-}, \hat{\lambda}_{h}^{-}, \hat{r}_{h}^{-}) \equiv (s^{u}_{h})_{UB},$$  \hspace{1cm} (32)

which holds for any function group $(\hat{\mu}_{h}^{\pm}, \hat{\lambda}_{h}^{\pm}, \hat{r}_{h}^{\pm}) \in X_{h} \otimes Q_{h} \otimes F_{h}$.

Similarly the bounds for the Energy equations follows that:

$$\pm s^{\Theta}_{h} = \sup_{(\Phi, r) \in X_{h}^{\Theta} \otimes F_{h}^{\Theta}} \inf_{w \in U_{h}^{\Theta}} L_{\Theta,\pm}(w, \Phi, r) \leq \inf_{w \in U_{h}^{\Theta}} L_{\Theta,-}(w, \hat{\Phi}_{h}^{\pm}, \hat{r}_{h}^{\pm}).$$  \hspace{1cm} (33)
Hence lower and upper bounds to the fine mesh output are constructed as follows:

\[
(s_h^\Theta)_{LB} \equiv \inf_{w \in U_h^\Theta} L^{\Theta,+}(w, \hat{\phi}_h^+, \hat{r}_h^+) \leq s_h^\Theta \leq -\inf_{w \in U_h^\Theta} L^{\Theta,-}(w, \hat{\phi}_h^-, \hat{r}_h^-) \equiv (s_h^\Theta)_{UB},
\]

which holds for any function pair \((\hat{\phi}_h^\pm, \hat{r}_h^\pm) \in X_h^\Theta \otimes J_h^\Theta\).

5 Adaptive Bound Method

This section describes the numerical procedure of the bound method applied to the steady incompressible Navier-Stokes and Energy equations. The bounds for the coupled Navier-Stokes and Energy equations are only asymptotically rigorous due to the non-coercivity arisen from Navier-Stokes equations as described in [21,23]. However, in practice all coarse meshes constructed lead to rigorous bounds as found in [14]. For the purpose of allowing straightforward implementation, the bound formulation is presented in a matrix form.

The bound procedure is composed of two-level computations performed on two different meshes namely the coarse \(H\)-mesh and the fine \(h\)-mesh calculations. First the coarse \(H\)-mesh calculation is composed of two sets of global computations and the hybrid-ux evaluation. On the other hand, the fine \(h\)-mesh calculation consists of local Neumann problems on each sub-domain followed by the bounds calculations.

5.1 Global Computations

There are two types of calculations that need to be solved on the entire computational domain: ‘primal’ and ‘dual’. The ‘primal’ problem is solved to obtain the field variables and the ‘dual’ problem is solved to obtain the adjoint variables. These problems are solved on a coarse \(H\)-mesh. To solve the non-linear system of field variables, the Newton’s method is utilized. More information regarding the Newton’s method for the steady incompressible Navier-Stokes equations can be found in [8,17,18,29,33].

5.1.1 The field variable calculation:

For the primal problem, field variables, i.e., velocity, pressure, and temperature fields, are calculated on the coarse \(H\)-mesh. The block-matrix form of the primal problem is summarized as
follows: Find \((u_{ih}, p_{ih}, \Theta_{ih}) \in X_{ih}^D \otimes Q_{ih} \otimes X_{ih}^{\Theta D}\) such that

\[
\begin{bmatrix}
A_H^u + C_H(u_H) & 0 & 0 & -D_{1H}^T & 0 \\
0 & A_H^u + C_H(u_H) & 0 & -D_{2H}^T & 0 \\
0 & 0 & A_H^u + C_H(u_H) - D_{3H}^T & 0 & 0 \\
-D_{1H} & -D_{2H} & -D_{3H} & 0 & 0
\end{bmatrix}
\begin{bmatrix}
u_{1H} \\
u_{2H} \\
u_{3H} \\
p_H
\end{bmatrix}
= \begin{bmatrix}
f_{1H} \\
f_{2H} \\
f_{3H} \\
g_H
\end{bmatrix},
\]

(35)

where \(A_H^u = \frac{1}{Re}A_H\) and \(A_H^{\Theta} = \frac{1}{Pe}A_H\) are the global diffusive matrix for the Navier-Stokes and Energy equations, respectively, \(C_H(u_H)\) is the global convection matrix, \(f_{ih}\) denotes the global velocity force vector, and \(g_{ih}\) is the global temperature force vector. Then the ‘velocity residual’ and the ‘temperature residual’ of the primal equation (35) are defined by

\[
\mathcal{R}_{u_{ih}}^{ps} = \sum_{T_H \in \mathcal{T}_H} \mathcal{R}_{u_{T_H}}^{ps},
\]

(36)

\[
\mathcal{R}_{\Theta_{ih}}^{ps} = \sum_{T_H \in \mathcal{T}_H} \mathcal{R}_{\Theta_{T_H}}^{ps},
\]

(37)

where

\[
\mathcal{R}_{u_{T_H}}^{ps} = f_{iT_H} - (A_{T_H}^u + C_{T_H}(u_{T_H})) u_{iT_H} + D_{iT_H}^T p_{iT_H},
\]

(38)

\[
\mathcal{R}_{\Theta_{T_H}}^{ps} = g_{iT_H} - (A_{T_H}^{\Theta} + C_{T_H}(u_{T_H})) \Theta_{iT_H},
\]

(39)

of which \(f_{iT_H}\) and \(g_{iT_H}\) are the combination of the force term and the inhomogeneous Neumann boundary terms for velocity and temperature fields, respectively.

The solution strategy for the Navier-Stokes system, the mixed penalty method is utilized for a slightly modified system such as: Find \((u_{ih}, p_{ih}) \in X_{ih}^D \otimes Q_{ih}\) such that

\[
\begin{bmatrix}
A_H^u + C_H(u_H) & 0 & 0 & -D_{1H}^T & 0 \\
0 & A_H^u + C_H(u_H) & 0 & -D_{2H}^T & 0 \\
0 & 0 & A_H^u + C_H(u_H) - D_{3H}^T & 0 & 0 \\
-D_{1H} & -D_{2H} & -D_{3H} & -\epsilon I & 0
\end{bmatrix}
\begin{bmatrix}
u_{1H} \\
u_{2H} \\
u_{3H} \\
p_H
\end{bmatrix}
= \begin{bmatrix}
f_{1H} \\
f_{2H} \\
f_{3H} \\
g_H
\end{bmatrix},
\]

(40)

where \(\epsilon\) is the penalty parameter which can be any small value of \(\epsilon > 0\) and \(I\) is the identity matrix. The non-linear nature of the system (40) requires an iterative scheme such as Picard method or Newton’s method. At each iteration, the resulting system raised from Equation (40) is linear, non-symmetric, and positive-definite so that the general iterative method such as Generalized Minimal
Residual (GMRES) or Bi-Conjugate Gradient (BiCG) method can be used. Once the velocity field is known from Equation (40), the temperature field can be easily evaluated by solving
\[
\left( A_H^\Theta + C_H(u_H) \right) \Theta_H = g_H, \tag{41}
\]
of which system is non-symmetric and linear so that a BiCG solver can be used.

5.1.2 The adjoint calculation:

For the dual problem, adjoints, i.e., adjoint velocity, pressure, and temperature fields, are also evaluated on the coarse $H$-mesh. The block-matrix form of the dual problem is given as follows: Find $(\psi_H, \lambda_H, \Phi_H) \in X_H \otimes Q_H \otimes X_H^\Theta$ such that
\[
\begin{bmatrix}
A_H^u + C_H^T(u_H) & 0 & 0 & -D_{1H}^T & 0 \\
0 & A_H^u + C_H^T(u_H) & 0 & -D_{2H}^T & 0 \\
0 & 0 & A_H^u + C_H^T(u_H) & -D_{3H}^T & 0 \\
-D_{1H} & -D_{2H} & -D_{3H} & 0 & 0 \\
0 & 0 & 0 & 0 & A_H^\Theta + C_H^T(u_H)
\end{bmatrix}
\begin{bmatrix}
\psi_1_H \\
\psi_2_H \\
\psi_3_H \\
\lambda_H \\
\Phi_H
\end{bmatrix}
= \begin{bmatrix}
-f_{1H}^O \\
-f_{2H}^O \\
-f_{3H}^O \\
-f_{pH}^O \\
-g_{H}^O
\end{bmatrix}, \tag{42}
\]
where $C_H^T(u_H)$ is the transpose of global convection matrix and $f_{1H}^O, f_{2H}^O, f_{3H}^O, f_{pH}^O, g_{H}^O$ denote the global velocity, pressure, and temperature output functionals, respectively. Then the ‘adjoint velocity residual’ and ‘adjoint temperature residual’ of the dual equation (42) are defined as
\[
\begin{align*}
R_{\psi_{1H}} &= \sum_{T_H \in T_H} R_{\psi_{1H}}^{(d)} \\
R_{\lambda_{1H}} &= \sum_{T_H \in T_H} R_{\lambda_{1H}}^{(d)} \\
\end{align*}
\]
where
\[
R_{\psi_{1H}}^{(d)} = -f_{1H}^O - (A_H^u + C_H^T(u_H)) \psi_{1H} + D_{1H}^T \lambda_{1H}, \tag{45}
\]
\[
R_{\lambda_{1H}}^{(d)} = -g_{H}^O - (A_H^\Theta + C_H^T(u_H)) \Phi_{1H}, \tag{46}
\]
of which $f_{1H}^O$ and $g_{H}^O$ are combination of the output functional and the inhomogeneous Neumann boundary terms for adjoint velocity and temperature fields, respectively.
The mixed penalty method is also used for solving Equation (42) as follows: Find \((\psi_l, \lambda_l) \in X_H \otimes Q_H\) such that
\[
\begin{bmatrix}
A_H + C^T_H(u_H) & 0 & 0 & -D^T_{1H} \\
0 & A_H + C^T_H(u_H) & 0 & -D^T_{2H} \\
0 & 0 & A_H + C^T_H(u_H) & -D^T_{3H} \\
-D_{1H} & -D_{2H} & -D_{3H} & -\varepsilon I
\end{bmatrix}
\begin{bmatrix}
\Psi_1 \\
\Psi_2 \\
\Psi_3 \\
\lambda
\end{bmatrix}
= \begin{bmatrix}
-f^O_{1H} \\
-f^O_{2H} \\
-f^O_{3H} \\
-f^O_{\rho H}
\end{bmatrix},
\tag{47}
\] 
where the system is linear, non-symmetric, and positive definite so that a GMRES or a BiCG solver can be used. Note that the adjoint calculation is much less expensive than the field variable calculation since it is linear. Once the adjoint velocity field is solved from Equation (47), the adjoint temperature field can be easily evaluated by solving
\[
\left(A^\Theta_H + C^T_H(u_H)\right) \Phi_H = -g^O_H,
\tag{48}
\]
This system is linear and non-symmetric so that a BiCG solver can be used.

5.2 Hybrid-Flux Calculation

The next step for the bounds calculation is the construction of the sub-domains. Each tetrahedron of the coarse mesh is transformed into a sub-domain. These sub-domains will be refined subsequently and will serve as the domain for the local Neumann problems. Therefore, the Lagrange multipliers that enforce continuity between sub-domains first need to be calculated. In this work this Lagrange multiplier is called the hybrid-flux.

The hybrid-flux is calculated by a novel hybrid-flux technique termed the directly equilibrated hybrid-flux approach [14]. The hybrid flux only connects faces, that is, each sub-domain tetrahedral has three hybrid-flux unknowns on each face. This approach is essential for interpolating the hybrid flux on the face of the finer sub-domain discretization.

The coarse \(H\)-mesh hybrid-flux calculation is written as follows:

- The hybrid-flux for the primal problem (35):
  
  Find \(\{t^{vr}_{l_H}, t^{\Theta vr}_{l_H}\} \in \mathcal{F}_H \otimes \mathcal{F}^\Theta_H\) such that
  \[
  B^T_H t^{vr}_{l_H} = R^u_{l_H}, \quad i = 1, 2, 3, \tag{49}
  \]
  \[
  B^T_H t^{\Theta vr}_{l_H} = R^\Theta_{l_H}. \tag{50}
  \]

- The hybrid-flux for the dual problem (42):
  
  Find \(\{t^{fu}_{l_H}, t^{\Theta fu}_{l_H}\} \in \mathcal{F}_H \otimes \mathcal{F}^\Theta_H\) such that
The boolean matrix $B_H$ on both sides. A standard conjugate gradient solver then solves the system. It is proven in [14] that the hybrid-flux obtained with this approach is equilibrated. Note that in this paper the hybrid-flux calculation is only required for the velocity and the adjoint velocity contributions due to the divergence-free property of Crouzeix-Raviart finite element. The pressure and the adjoint pressure contributions are not needed to be considered due to the discontinuous pressure approximation.

5.3 Local Neumann Problems

The local Neumann subproblems for the bound method is evaluated independently for each sub-domain ‘$k$’ as summarized in following subsections.

5.3.1 Interpolation:

Variables approximated in the coarse $H$-mesh are required to be interpolated onto the fine $h$-mesh. The field variables $(u_H, p_H, \Theta_H)$ and the adjoint variables $(\psi_H, \lambda_H, \Phi_H)$ are interpolated by using a ‘volume’ interpolation where as the hybrid-fluxes $(t_H^{\psi/du}, t_H^{\Phi/du})$ are interpolated by exploiting a ‘face’ interpolation.

\[
\hat{I}_H^u u_H \bigg|_{T_H} \rightarrow u_h \bigg|_{T_H}; \quad (53)
\]
\[
\hat{I}_H^\psi \psi_H \bigg|_{T_H} \rightarrow \psi_h \bigg|_{T_H}; \quad (54)
\]
\[
\hat{I}_H^p p_H \bigg|_{T_H} \rightarrow p_h \bigg|_{T_H}; \quad (55)
\]
\[
\hat{I}_H^\lambda \lambda_H \bigg|_{T_H} \rightarrow \lambda_h \bigg|_{T_H}; \quad (56)
\]
\[
\hat{I}_H^\Theta \Theta_H \bigg|_{T_H} \rightarrow \Theta_h \bigg|_{T_H}; \quad (57)
\]
\[
\hat{I}_H^\Phi \Phi_H \bigg|_{T_H} \rightarrow \Phi_h \bigg|_{T_H}; \quad (58)
\]
\[
\hat{I}_H^{t^{\psi/du}} t_H^{\psi/du} \bigg|_{\gamma_H} \rightarrow t_h^{\psi/du} \bigg|_{\gamma_H}; \quad (59)
\]
\[
\hat{I}_H^{t^{\Phi/du}} t_H^{\Phi/du} \bigg|_{\gamma_H} \rightarrow t_h^{\Phi/du} \bigg|_{\gamma_H}. \quad (60)
\]

5.3.2 Incompressible local projections:

Then the fine $h$-mesh local Neumann subproblems require that the local projections of $(u_H^{k_i}, p_H^{k_i})$ and $(\psi_H^{k_i}, \lambda_H^{k_i})$ on the fine $h$-mesh respect the incompressibility constraints for each sub-domain. A set of incompressible local projections are evaluated as follows:
The incompressible local projection for the primal problem is:

\[ \text{Find } (\delta_{i_h}^{u,k}, \delta_{h}^{p,k}) \in V_h \otimes Y_h \text{ such that} \]

\[
\begin{bmatrix}
A_h^k & 0 & 0 & -D_h^{k T} \\
0 & A_h^k & 0 & -D_h^{k T} \\
0 & 0 & A_h^k & -D_h^{k T} \\
-D_h^{1} - D_h^{2} - D_h^{3} & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta_{i_h}^{u,k} \\
\delta_{2_h}^{u,k} \\
\delta_{3_h}^{u,k} \\
\delta_{i_h}^{p,k}
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
\sum_{i=1}^{3} D_h^{k} u_{i_h}^{k}
\end{bmatrix}.
\]

(61)

The incompressible local projection for dual problem is:

\[ \text{Find } (\delta_{i_h}^{\psi,k}, \delta_{h}^{\lambda,k}) \in V_h \otimes Y_h \text{ such that} \]

\[
\begin{bmatrix}
A_h^k & 0 & 0 & -D_h^{k T} \\
0 & A_h^k & 0 & -D_h^{k T} \\
0 & 0 & A_h^k & -D_h^{k T} \\
-D_h^{1} - D_h^{2} - D_h^{3} & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta_{i_h}^{\psi,k} \\
\delta_{2_h}^{\psi,k} \\
\delta_{3_h}^{\psi,k} \\
\delta_{i_h}^{\lambda,k}
\end{bmatrix} =
\begin{bmatrix}
0 \\
0 \\
0 \\
\sum_{i=1}^{3} D_h^{k} \psi_{i_h}^{k}
\end{bmatrix}.
\]

(62)

The interpolated field and adjoint variables, i.e., \((u_{i_h}^{k}, p_{h}^{k})\) and \((\psi_{i_h}^{k}, \lambda_{h}^{k})\), respectively, are modified as follows:

- For the **primal** subproblem:

\[
\tilde{u}_{i_h}^{k} = u_{i_h}^{k} + \delta_{i_h}^{u,k}, \quad i = 1, 2, 3, \quad (63)
\]

\[
\tilde{p}_{h}^{k} = p_{h}^{k} + \delta_{h}^{p,k}. \quad (64)
\]

- For the **dual** subproblem:

\[
\tilde{\psi}_{i_h}^{k} = \psi_{i_h}^{k} + \delta_{i_h}^{\psi,k}, \quad i = 1, 2, 3, \quad (65)
\]

\[
\tilde{\lambda}_{h}^{k} = \lambda_{h}^{k} + \delta_{h}^{\lambda,k}. \quad (66)
\]

### 5.3.3 Reconstructed error calculations:

The modified field variables \((\tilde{u}_{i_h}^{k}, \tilde{p}_{h}^{k}, \Theta_{h})\), the modified adjoint variables \((\tilde{\psi}_{i_h}^{k}, \tilde{\lambda}_{h}^{k}, \Phi_{h})\), and interpolated hybrid-flux \((\tilde{t}_{i_h}^{h}, \tilde{\Theta}_{h})\) are now used in the local Neumann subproblems. Introducing the local discontinuous fine mesh errors for field variables as \((\tilde{e}_{i_h}^{ur,k}, \tilde{e}_{h}^{ur,k}, \tilde{e}_{h}^{\Theta,ur,k})\) and similarly for adjoints as \((\tilde{e}_{i_h}^{au,k}, \tilde{e}_{h}^{au,k}, \tilde{e}_{h}^{\Theta,au,k})\), the solution of these problems lead to obtaining local error for each sub-domain 'k' as follows:
• The local primal Neumann subproblem:
Find \((e_{i_h}^{\rho,k}, e_{h}^{\rho,k}, e_{h}^{\Theta_p,k}) \in W_h \otimes Y_h \otimes V_h^\Theta\) such that

\[
\begin{bmatrix}
A_h^{u,k} & 0 & 0 & -D_h^{1,k} T & 0 \\
0 & A_h^{\Sigma_1^k} & 0 & -D_h^{2,k} T & 0 \\
0 & 0 & A_h^{\Sigma_2^k} & -D_h^{3,k} T & 0 \\
-D_h^{1,k} - D_h^{2,k} - D_h^{3,k} & 0 & 0 & 0 & A_h^{\Theta_p,k}
\end{bmatrix}
\begin{bmatrix}
e_{i_h}^{\rho,k} \\
e_{h}^{\rho,k} \\
e_{h}^{\Theta_p,k}
\end{bmatrix}
=
\begin{bmatrix}
R_{i_h}^{u,k} \\
R_{h}^{\rho,k} \\
R_{h}^{\Theta_p,k}
\end{bmatrix}
- \begin{bmatrix}
B_h^{k} t_{i_h}^{\rho,k} \\
B_h^{k} t_{i_h}^{\rho,k} \\
B_h^{k} t_{i_h}^{\rho,k}
\end{bmatrix} .
\] (67)

where

\[
R_{i_h}^{\rho,k} = f_{i_h}^{u,k} - \left( A_h^{u,k} + C_h^T (\tilde{u}_h) \right) \tilde{u}_h^{k} + D_h^{k} T \bar{p}_h^{k}, \quad i = 1, 2, 3,
\] (68)

\[
R_{h}^{\Theta_p,k} = g_h^{k} - \left( A_h^{\Theta_p,k} + C_h^T (\tilde{u}_h) \right) \Theta_h^k.
\] (69)

• The local dual Neumann subproblem:
Find \((e_{i_h}^{\rho,k}, e_{h}^{\rho,k}, e_{h}^{\Theta_p,k}) \in W_h \otimes Y_h \otimes V_h^\Theta\) such that

\[
\begin{bmatrix}
A_h^{u,k} & 0 & 0 & -D_h^{1,k} T & 0 \\
0 & A_h^{\Sigma_1^k} & 0 & -D_h^{2,k} T & 0 \\
0 & 0 & A_h^{\Sigma_2^k} & -D_h^{3,k} T & 0 \\
-D_h^{1,k} - D_h^{2,k} - D_h^{3,k} & 0 & 0 & 0 & A_h^{\Theta_p,k}
\end{bmatrix}
\begin{bmatrix}
e_{i_h}^{\rho,k} \\
e_{h}^{\rho,k} \\
e_{h}^{\Theta_p,k}
\end{bmatrix}
=
\begin{bmatrix}
R_{i_h}^{u,k} \\
R_{h}^{\rho,k} \\
R_{h}^{\Theta_p,k}
\end{bmatrix}
- \begin{bmatrix}
B_h^{k} t_{i_h}^{\rho,k} \\
B_h^{k} t_{i_h}^{\rho,k} \\
B_h^{k} t_{i_h}^{\rho,k}
\end{bmatrix} .
\] (70)

where

\[
R_{i_h}^{\rho,k} = -f_{i_h}^{O_k} - \left( A_h^{u,k} + C_h^T (\tilde{u}_h) \right) \tilde{u}_h^{k} + D_h^{k} T \bar{p}_h^{k}, \quad i = 1, 2, 3,
\] (71)

\[
R_{h}^{\Theta_p,k} = -g_h^{O_k} - \left( A_h^{\Theta_p,k} + C_h^T (\tilde{u}_h) \right) \Phi_h^k.
\] (72)

5.4 Bounds Construction

The bounds values are obtained by summing the local contribution of each sub-domain. The lower and upper output bounds, i.e., \((s_h)_{LB}\) and \((s_h)_{UB}\), are expressed in terms of the output predictor \((s_h)_{pr}\) and the half bound-gap \(\Delta H\) such as:

\[
(s_h)_{LB} = (s_h)_{pr} - \Delta H, \quad (s_h)_{UB} = (s_h)_{pr} + \Delta H,
\] (73) (74)
\[ (s_h)_{pre} = (s_h^u)_{pre} + (s_h^\Theta)_{pre}, \]  
\[ \Delta H = \Delta H^u + \Delta H^\Theta, \]  
\[ (s_h^u)_{pre} = \bar{s} - 2 \sum_{k=1}^{N_h} \sum_{i=1}^{3} \begin{bmatrix} \bar{e}^u_{ih}, A_h^u, \bar{e}_{ih} \end{bmatrix}, \]  
\[ \bar{s} = \bar{s} + \sum_{k=1}^{N_h} \sum_{i=1}^{3} \begin{bmatrix} \bar{\theta}^T_{ih}, (A_h^u k + C_h^u k) \bar{u}^u_{ih} - D_h^u k T k^T p_h^k - f_h^k) - \bar{k}_h k^T D_h^u k^T \bar{u}^u_{ih} \end{bmatrix}, \]  
\[ \Delta H^u = \sum_{k=1}^{N_h} \Delta H^u k, \]  
and of which for the ‘flow’ field

\[ (s_h^\Theta)_{pre} = s_H - 2 \sum_{k=1}^{N_h} \begin{bmatrix} \bar{e}^\Theta_{ih}, A_h^\Theta, \bar{e}_{ih} \end{bmatrix}, \]  
\[ s_H = \sum_{k=1}^{N_h} s_H^O k \begin{bmatrix} \Theta^k_h \end{bmatrix}, \]  
\[ \Delta H^\Theta = \sum_{k=1}^{N_h} \Delta H^\Theta k, \]  

Both \((s_h)_{pre}\) and \(\Delta H\) can also be expressed alternatively as follows:

\[ (s_h)_{pre} = \frac{1}{2} [(s_h)_{UB} + (s_h)_{LB}], \]  
\[ \Delta H = \frac{1}{2} [(s_h)_{UB} - (s_h)_{LB}], \]  

where the output predictor (86) is the average value for the lower and upper output bounds and the half bound-gap (87) defines the half-gap between the lower and upper output bounds. Note
that the output predictor is independent of the stabilization parameter \( \kappa \) and the elemental decomposition \( \Delta_H \) can serve as local indicators in adaptive strategies [11–13]. The accuracy for the output bound method can be measured by the relative half bound-gap \( \theta \) defined by

\[
\theta = \frac{\Delta_H}{(s_h)_{ge}}
\]

which is the combination of the half bound-gap and the output predictor.

5.5 Optimal Bounds

The stabilization parameter \( \kappa \) may be chosen to minimize \( \Delta_H \). The parameter arises by writing all variables as linear functions in \( \kappa \) and then by deriving output bounds as a function of \( \kappa \). This procedure does not change the bounding properties and output bounds remain rigorous in an asymptotic region. An optimal stabilization parameter \( \kappa^* \) can be evaluated by utilizing minimization property of the half bound-gap. Note that, in the case of \( \kappa = 1 \), the usual non-optimal output bounds are obtained. Then the optimal stabilization parameters, i.e., \( \kappa^{u,*} \) and \( \kappa^{\Theta,*} \), can be formulated as follows:

\[
\kappa^{u,*} = \frac{\sum_{i=1}^{3} e_{iu,k}^T A_h e_{iu,k}}{\sum_{i=1}^{3} e_{iu,k}^T A_h e_{iu,k}},
\]

\[
\kappa^{\Theta,*} = \frac{e_{\Theta,iu,k}^T A_h e_{\Theta,iu,k}}{e_{\Theta,iu,k}^T A_h e_{\Theta,iu,k}}, \quad k = 1, \ldots, N_k,
\]

which are derived from minimization property of the half bound-gap such that

\[
\frac{\partial \Delta_H}{\partial \kappa} \bigg|_{\kappa = \kappa^*} = 0,
\]

where \( \Delta_H \) and \( \kappa^* \) can be applied for \( (\cdot)^u \) and \( (\cdot)^{\Theta} \), respectively.

5.6 Adaptivity

The adaptive procedure summarized in this section follows directly from [11–13] but is novel in the sense that in this work it is applied to the Navier-Stokes and Energy equations in three space dimensions. To minimize the half bound-gap (or improve the sharpness) of the bound method, the adaptive sub-domain refinement strategy can be applied. Starting from an initial mesh \( T^0_H \), the half bound-gap \( \Delta_H \) and local indicators \( \Delta^k_{ijH} = \Delta^u_{ijH} + \Delta^\Theta_{ijH} \) are calculated. At each refinement
cycle, each tetrahedron $T^n_H$ is a refinement based on the local indicator. Note that if the output is only a functional of the temperature, then only the contribution of the temperature error to the local indicators considered. Similarly, for the velocity output, only the velocity local contribution is considered. Here $n$ denotes the $n^\text{th}$ refinement. To identify the sub-domains $T^n_{H}^{-1}$ that need to be refined, first the largest elemental contribution $\Delta^{n-1}_{\text{max}}$ to the bound-gap $\Delta_H \left( T^n_{H}^{-1} \right)$ can be found such that

$$\Delta^{n-1}_{\text{max}} = \max_{T^n_H \in T^n_{H}^{-1}} \Delta_{T^n_H} \left( T^n_{H}^{-1} \right),$$

and then all tetrahedra $T^n_H \in T^n_{H}^{-1}$ for which

$$\Delta_{T^n_H} \left( T^n_{H}^{-1} \right) \geq \beta \Delta^{n-1}_{\text{max}},$$

can be selected for refinements. The parameter $\beta$ controls the cutoff, i.e., identifying elements to be refined at each adaptive cycle ($0 < \beta < 1$). The adaptive process can be stopped when $\theta^n \leq \theta^\text{obj}$ where $\theta^\text{obj}$ is the prescribed accuracy.

The adaptive sub-domain refinement strategy is summarized as follows:

1. Calculate the corresponding bound-gap $\Delta_H$ for each tetrahedron sub-domain $T^n_H$; evaluate the local indicators $\Delta^k_{T^n_H}$; find the maximum sub-domain contribution $\Delta^{n}_{\text{max}}$ among the local indicators; tag sub-domains that satisfied Equation (93);
2. Refine the tagged sub-domains identified in Step 1;
3. Repeat Step 1 and Step 2 until the desired accuracy is achieved or the maximum number of cycles is reached.

6 Results

To validate the proposed technique in terms of bound sharpness and computational cost, the motivation example in Section 2 is revisited. The output of interest considered herein is the mean temperature in the specific area region $\Gamma^O$ and hence $s = s^\Theta$ and $s^H = 0$. All bound calculations are performed with a refinement parameter $R = 4$. Furthermore, all solvers used in this paper are based on LASPack sparse solver [30]. All computations are carried on a dual processor AMD Athlon 1900+ 1.6 GHz CPU computer having 1.5 Gb RAM memory, running Linux.

For this test case, the flow is assumed to be driven by a constant pressure gradient which acts as its driving force. For this problem, $x = (x_1, x_2, x_3)$ and has corresponding unit vectors $\hat{x}_1, \hat{x}_2, \hat{x}_3$. The computational domain shown in Figure 1 is denoted as $\Omega$ and is bounded by $[0,3] \times [0,1] \times [0,2]$ with two electronic chips located in $[0,3/4] \times [1/3,2/3] \times [0,2/3]$ and $[2,7/3] \times [1/3,2/3] \times [0,2/3]$, respectively. The outer boundaries consist of six surfaces, i.e., $\Gamma_1 = 0 \times [0,1] \times [0,2], \Gamma_2 = 3 \times [0,1] \times [0,2], \Gamma_3 = 0 \times [0,3] \times [0,2/3], \Gamma_4 = 3 \times [1 \times 0] \times [0,2/3], \Gamma_5 = 0 \times [3] \times [0,1] \times [0,2], \Gamma_6 = 3 \times [1 \times 0] \times [0,2/3]$, and $[1 \times 0] \times [0,2/3] \times [1 \times 0] \times [0,2/3]$, and
Fig. 3. Sequence of the adaptively refined sub-domain meshes: (a) initial mesh $\mathcal{T}_H$ having 2400 tetrahedra; (b) uniformly refined mesh $\mathcal{T}_{H/2}$ having 19,200 tetrahedra; (c) adaptively refined mesh $\mathcal{T}^3_H$ having 5537 tetrahedra; (d) adaptively refined mesh $\mathcal{T}^4_H$ having 8721 tetrahedra; (e) adaptively refined mesh $\mathcal{T}^5_H$ having 14,195 tetrahedra; (f) adaptively refined mesh $\mathcal{T}^6_H$ having 18,797 tetrahedra.

$\Gamma_6 = [0, 3] \times [0, 1] \times \left[\frac{2}{3}, 1\right], \Gamma_1 = [0, 1] \times \left[\frac{2}{3}, 1\right], \Gamma_2 = [\frac{2}{3}, 2] \times \left[\frac{2}{3}, 1\right], \Gamma_3 = \left[\frac{2}{3}, 1\right] \times \left[\frac{2}{3}, 2\right], \Gamma_4 = \left[\frac{2}{3}, 1\right] \times [0, 1], \Gamma_5 = \left[\frac{2}{3}, 1\right] \times [0, 1], \Gamma_6 = \left[\frac{2}{3}, 1\right] \times [0, 1]$, where $\Gamma_1$ and $\Gamma_2$ are the periodic boundaries and $\Gamma_3$ to $\Gamma_6$ are the homogeneous Dirichlet boundaries. Note that boundary surfaces of the electronic chips have non-slip velocity and a given heat-flux (inhomogenous Neumann). The velocity and pressure fluctuation fields are periodic in the $\hat{x}_1$ direction. The flow is driven by a pressure gradient which induces the forcing terms $f_1 = \frac{100}{Re}$ and $f_2 = f_3 = 0$ in Equation (20). The pressure solution refers to the fluctuations with respect to this pressure gradient. The heat flux boundary condition, $g^N = 1$, contributes to the temperature force vector in the right hand side of Equation (21). Reynolds number and Péclet number considered in this case are $Re = 100$ and $Pe = 100$, respectively. The mesh adaptation cut-off parameter $\beta = 0.4$ is selected.

Figure 3 displays uniformly and adaptively refined sub-domains used in the bound method. Recall that these sub-domains are also the coarse mesh used for global calculations. Typically our method starts calculations from an initial coarse mesh $\mathcal{T}_H$ as shown in Figure 3a. To sharpen the bound gap one can simply use an uniformly refined mesh $\mathcal{T}_{H/2}$ as shown in Figure 3b. The bound method is also appropriate for an adaptive refinement strategies since the method naturally provides local error information (i.e., local half bound-gaps) which can be used for local error indicators for mesh refinements. Figures 3c to 3f show the sequence of adaptively refined sub-domains generated. Earlier analysis and comparison with the fine mesh output [14,10], showed that at low Reynolds number the bound values are always rigorous even on the coarsest possible sub-domain mesh. In this work, the number of sub-domains is increased at each adaptive cycle, therefore, the bounds of the fine mesh output should be strict bounds. The proposed adaptive
technique not only captures the error of physical solutions, but also identifies the error of output quantities in engineering design context, which can be observed from adaptively generated sub-domain mesh in Figure 3. As more adaptive refinement cycles are performed, more tetrahedra sub-domains concentrate near the output surface $\Gamma^O$ as well as around the electronic chips where heat-fluxes are non zero and at corner singularities. Note that the separated flow behind the second chip is not contributing to the output of interest and is therefore that region is less refined.

![Image of mesh and temperature contours](attachment:image.png)

Fig. 4. Illustrations of the surface mesh, temperature contours, and uniform velocity vector for the adapted mesh $\mathcal{T}_H^8$: (a) mesh slice at $x_3 = \frac{1}{3}$, i.e., center horizontal plane (top); (b) temperature contours, ranged from 0 to 18 at unit intervals, of the surface mesh (middle); (c) uniform velocity vector of the surface mesh (bottom).

Furthermore, the adaptively refined mesh properly captures features of multi-physics as well as the interested output for the array of electronic chips. The solution on this subdomain mesh, as shown at the center horizontal plane in Figure 4, can also be used to analyze the results in addition to the bounds to the output.

Table 1 summarizes the optimal bounding values of the output $s$ for the adaptive bound method. As the number of tetrahedra increases adaptively, the half bound-gap $\Delta_H$ and the relative bound-gap $\Theta$ become sharper. Clearly, the adaptive refinement strategy gives sharper bound-gaps than the uniform refinement strategy with much less tetrahedral sub-domains, i.e., the adaptively refined mesh $\mathcal{T}_H^7$ has 14,195 tetrahedra and gives a bound-gap that is slightly sharper than the uniformly
Table 1
Bounding values for the adaptive bound method.

<table>
<thead>
<tr>
<th></th>
<th>$\sigma^0_H$</th>
<th>$\sigma^3_H$</th>
<th>$\sigma^5_H$</th>
<th>$\sigma^7_H$</th>
<th>$\sigma^8_H$</th>
<th>$\sigma^0_{H/2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Elements</td>
<td>2400</td>
<td>5337</td>
<td>8721</td>
<td>14,195</td>
<td>18,797</td>
<td>19,200</td>
</tr>
<tr>
<td># of Nodes</td>
<td>11,823</td>
<td>25,225</td>
<td>40,679</td>
<td>65,541</td>
<td>86,547</td>
<td>88,783</td>
</tr>
<tr>
<td>$(s_h)_{UB}$</td>
<td>15.98</td>
<td>14.49</td>
<td>14.36</td>
<td>14.19</td>
<td>14.09</td>
<td>14.13</td>
</tr>
<tr>
<td>$(s_h)_{LB}$</td>
<td>11.00</td>
<td>12.28</td>
<td>12.43</td>
<td>12.59</td>
<td>12.66</td>
<td>12.52</td>
</tr>
<tr>
<td>$(s_h)_{pre}$</td>
<td>13.49</td>
<td>13.39</td>
<td>13.39</td>
<td>13.38</td>
<td>13.38</td>
<td>13.32</td>
</tr>
<tr>
<td>$\pm \Delta_H$</td>
<td>$\pm 2.492$</td>
<td>$\pm 1.106$</td>
<td>$\pm 0.963$</td>
<td>$\pm 0.799$</td>
<td>$\pm 0.716$</td>
<td>$\pm 0.807$</td>
</tr>
<tr>
<td>$\theta(%)$</td>
<td>18.47</td>
<td>8.26</td>
<td>7.19</td>
<td>5.97</td>
<td>5.35</td>
<td>6.06</td>
</tr>
</tbody>
</table>

Table 2
CPU time (s) breakdown for the adaptive bound method.

<table>
<thead>
<tr>
<th></th>
<th>$\sigma^0_H$</th>
<th>$\sigma^3_H$</th>
<th>$\sigma^5_H$</th>
<th>$\sigma^7_H$</th>
<th>$\sigma^8_H$</th>
<th>$\sigma^0_{H/2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Elements</td>
<td>2400</td>
<td>5337</td>
<td>8721</td>
<td>14,195</td>
<td>18,797</td>
<td>19,200</td>
</tr>
<tr>
<td># of Nodes</td>
<td>11,823</td>
<td>25,225</td>
<td>40,679</td>
<td>65,541</td>
<td>86,547</td>
<td>88,783</td>
</tr>
<tr>
<td>Navier-Stokes problem</td>
<td>1892</td>
<td>7460</td>
<td>17,175</td>
<td>42,165</td>
<td>66,436</td>
<td>75,709</td>
</tr>
<tr>
<td>Energy problem</td>
<td>24</td>
<td>110</td>
<td>257</td>
<td>639</td>
<td>1073</td>
<td>1093</td>
</tr>
<tr>
<td>Hybrid-flux calculation</td>
<td>17</td>
<td>49</td>
<td>97</td>
<td>196</td>
<td>301</td>
<td>296</td>
</tr>
<tr>
<td>Coarse $H$-mesh CPU</td>
<td>1933</td>
<td>7619</td>
<td>17,529</td>
<td>43,000</td>
<td>67,810</td>
<td>77,098</td>
</tr>
<tr>
<td>Interpolation</td>
<td>17</td>
<td>72</td>
<td>177</td>
<td>422</td>
<td>721</td>
<td>753</td>
</tr>
<tr>
<td>Neumann subproblems</td>
<td>7957</td>
<td>19,055</td>
<td>31,957</td>
<td>51,173</td>
<td>67,895</td>
<td>62,533</td>
</tr>
<tr>
<td>Fine $h$-mesh CPU</td>
<td>7974</td>
<td>19,127</td>
<td>32,134</td>
<td>51,595</td>
<td>68,616</td>
<td>63,286</td>
</tr>
<tr>
<td>Total CPU</td>
<td>9907</td>
<td>26,746</td>
<td>48,663</td>
<td>94,595</td>
<td>136,426</td>
<td>140,384</td>
</tr>
</tbody>
</table>

Table 1 reports the CPU (s) cost. The total CPU cost is divided into two subcategories; namely the coarse $H$-mesh calculation and the fine $h$-mesh calculation where both computational costs increases as the number of sub-domains increases. Among the coarse $H$-mesh CPU, the cost for Navier-Stokes calculation is the most expensive and it becomes significantly dominant for larger number of tetrahedra sub-domains. The CPU cost of the fine $h$-mesh calculation takes 3.4 times more than the coarse $H$-mesh calculation for the initial tetrahedron $\mathcal{T}_H$. Nevertheless, the former becomes less expensive than the latter for the larger number of tetrahedra. This is due to high cost arising for the Navier-Stokes calculation. The cost involved in the hybrid-flux calculations
is negligible compared to others thanks to the directly equilibrated approach [14]. In addition the cost of the interpolation does not account for a significant part in the fine $h$-mesh calculation.

7 Conclusion

The adaptive sub-domain procedure applied to the bound method based on the directly equilibrated hybrid-flux approach is developed for the steady incompressible Navier-Stokes and Energy equations in complex three space dimensions domains to address a convective heat transfer problem in a series of electronic chip devices. The proposed technique provides asymptotic sharp bounds to the mean temperature in a specific region. For our model problem, this technique also provides the same estimate of the fine mesh output at 28% of the cost of calculating it with a standard finite element method. The meshes obtained from the adaptively refined sub-domains show that both the multi-physics of the problem as well as the output of interest are captured.

Acknowledgement

The work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC). The authors would also like to acknowledge the CLUMEQ Supercomputer Centre for providing computer resources.

References


A Finite element spaces

Two-level tetrahedral discretizations of the computational domain $\Omega$ are required: the coarse mesh $H$-mesh, $T_H$, consisting of sub-domains $T_H$ and the fine $h$-mesh, $T_h$, consisting of elements $T_h$ such that

$$\Omega = \bigcup_{T_\delta \in T_\delta} T_\delta,$$

(A.1)

where $\delta = H$ and $\delta = h$ denote the coarse mesh and fine mesh discretizations, respectively. The $h$-mesh tetrahedron $T_h$ is a refinement of $T_H$ which can be considered as tetrahedral sub-domains, i.e.,

$$T_H = \bigcup_{T_h \in K_{T_H}} T_h,$$

(A.2)
where $\mathbf{K}_{\gamma_H}$ is the set of $h$-mesh tetrahedral elements that include all $T_H$. Then, the $h$-mesh face $\mathcal{E}_h$ is a refinement of $\mathcal{E}_H$ which is considered as triangular sub-domains, i.e.,

$$
\overline{\gamma}_H = \bigcup_{\gamma_h \in \mathbf{K}_{\gamma_H}} \overline{\gamma}_h,
$$

(A.3)

where $\mathbf{K}_{\gamma_H}$ is the set of $h$-mesh triangular elements that include all $\gamma_H$. A uniform $R$-refinement indicates that the $h$-mesh consists of $\mathbb{R}^3$ tetrahedra $T_h$ sub-refinements of each $T_H$. The reference finite element discretizations are based on the $\mathbf{P}_2^+ - \mathbf{P}_1$ basis and are known as the three-dimensional Crouzeix-Raviart finite element spaces [7]. Recall that the pressure field has $4 - \mathbf{P}_1$ nodes where as the velocity and temperature fields contain $15 - \mathbf{P}_2^+$ nodes.

Regular piecewise continuous finite element subspaces for the velocity field of the coarse and fine meshes are given by

$$
X_H = \{ v_i \in X : v_i|_{T_H} \in \mathbf{P}_2^+(T_H), \ \forall T_H \in \mathcal{T}_H \},
$$

(A.4)

$$
X_h = \{ v_i \in X : v_i|_{T_h} \in \mathbf{P}_2^+(T_h), \ \forall T_h \in \mathcal{T}_h \},
$$

(A.5)

and for the temperature field

$$
X^\Theta_H = \{ w \in Y : w|_{T_H} \in \mathbf{P}_2^+(T_H), \ \forall T_H \in \mathcal{T}_H \},
$$

(A.6)

$$
X^\Theta_h = \{ w \in Y : w|_{T_h} \in \mathbf{P}_2^+(T_h), \ \forall T_h \in \mathcal{T}_h \},
$$

(A.7)

where

$$
\mathbf{P}_2^+(T_\delta) = \{ \mathbf{P}_2(T_\delta) \oplus \mathbf{B}_3 \oplus \mathbf{B}_4, \ \forall T_\delta \in \mathcal{T}_\delta \},
$$

(A.8)

which denotes the space of quadratic polynomials enhanced by the cubic facial ‘bubble’ function (i.e., $\mathbf{B}_3 = \{ \xi_1 \xi_2 \xi_3, \xi_1 \xi_2 \xi_4, \xi_1 \xi_3 \xi_4, \xi_2 \xi_3 \xi_4 \}$) over $\gamma_\delta$ which represents the element face space and the fourth-order volumetric ‘bubble’ function (i.e., $\mathbf{B}_4 = \xi_1 \xi_2 \xi_3 \xi_4$) over $T_\delta$. Furthermore, the finite element subspaces for the pressure field of the coarse and fine meshes are as follows:

$$
Q_H = \{ q \in Q : q|_{T_H} \in \mathbf{P}_1(T_H), \ \forall T_H \in \mathcal{T}_H \},
$$

(A.9)

$$
Q_h = \{ q \in Q : q|_{T_h} \in \mathbf{P}_1(T_h), \ \forall T_h \in \mathcal{T}_h \}.
$$

(A.10)

The sub-domain ‘local velocity’ spaces which are $U_H(T_H)$ and $U_h(T_H)$ are introduced as follows:

$$
U_H(T_H) = \{ v_i|_{T_H} \in \mathbf{P}_2^+(T_H), \ \forall T_H \in \mathcal{T}_H \},
$$

(A.11)

$$
U_h(T_H) = \{ v_i|_{T_h} \in \mathbf{P}_2^+(T_h), \ \forall T_h \in \mathbf{K}_{\gamma_H} \} \cap \mathcal{H}^1(T_H), \ \forall T_H \in \mathcal{T}_H,
$$

(A.12)

and the sub-domain ‘local temperature’ spaces are given by
\[ U^0_H(T_H) = \{ w|_{T_H} \in P_2^+(T_H), \forall T_H \in T_H \}, \] (A.13)
\[ U^0_h(T_H) = \{ w|_{T_h} \in P_2^+(T_h), \forall T_h \in K_{T_h} \} \cap H^1(T_H), \forall T_H \in T_h, \] (A.14)

where \( K_{T_h} \) denotes the set of \( h \)-mesh elements contained in \( T_H \). Similarly the sub-domain ‘incompressibility’ constraint on the velocity on behalf of the discontinuous pressure approximation. The corresponding global representations. Note that, associated global representations of ‘discontinuous temperature’ spaces are given as:

\[ Y_H(T_H) = \{ q|_{T_H} \in P_1(T_H), \forall T_H \in T_H \}, \] (A.15)
\[ Y_h(T_H) = \{ q|_{T_h} \in P_1(T_h), \forall T_h \in K_{T_h} \} \cap L^2(T_H), \forall T_H \in T_h. \] (A.16)

The local function spaces including the ‘incompressibility constraint’ is defined by

\[ Z_H(T_H) = \{ v_i \in U_H(T_H) : (\nabla v_i, \nabla q)|_{T_H} = 0, \forall q \in Y_H(T_H) \}, \] (A.17)
\[ Z_h(T_H) = \{ v_i \in U_h(T_H) : (\nabla v_i, \nabla q)|_{T_H} = 0, \forall q \in Y_h(T_H) \}. \] (A.18)

Then the associated global representations of ‘discontinuous velocity’ spaces are as follows:

for the case without the ‘incompressibility’ constraint:

\[ V_H = \{ v_i \in L^2(\Omega) : v_i|_{T_H} \in U_H(T_H), \forall T_H \in T_H \}, \] (A.19)
\[ V_h = \{ v_i \in L^2(\Omega) : v_i|_{T_h} \in U_h(T_H), \forall T_H \in T_h \}. \] (A.20)

for the case with the ‘incompressibility’ constraint:

\[ W_H = \{ v_i \in L^2(\Omega) : v_i|_{T_H} \in Z_H(T_H), \forall T_H \in T_H \}, \] (A.21)
\[ W_h = \{ v_i \in L^2(\Omega) : v_i|_{T_h} \in Z_h(T_H), \forall T_H \in T_h \}. \] (A.22)

In fact, \( U^\theta_H(T_H) \) and \( Z^\theta_H(T_H) \) are Neumann spaces over each \( T_H \), for which \( V^\theta_H \) and \( W^\theta_H \) are the corresponding global representations. Note that, \( Z^\theta_H(T_H) \) imposes the necessary global incompressibility constraint on the velocity on behalf of the discontinuous pressure approximation. The associated global representations of ‘discontinuous temperature’ spaces are given as:

\[ V^\theta_H = \{ w \in L^2(\Omega) : w|_{T_H} \in U^\theta_H(T_H), \forall T_H \in T_H \}, \] (A.23)
\[ V^\theta_h = \{ w \in L^2(\Omega) : w|_{T_h} \in U^\theta_h(T_H), \forall T_H \in T_h \}. \] (A.24)

Let \( E(T_{H}) \) and \( E(T_{h}) \) denote the set of ‘open faces’ in the tetrahedron \( T_H \) and \( T_h \). Then the spaces of velocity functions over the element faces \( \gamma_H \) and \( \gamma_h \) are introduced as follows:

\[ J_H = \{ v_i|_{\gamma_H} \in P_2(\gamma_H) \oplus B_3, \forall \gamma_H \in E(T_{H}), v_i|_{\Gamma^W} = 0 \}, \] (A.25)
\[ J_h = \{ v_i|_{\gamma_h} \in P_2(\gamma_H) \oplus B_3, \forall \gamma_H \in E(T_{h}) \cap E(T_{H}), v_i|_{\Gamma^W} = 0 \}. \] (A.26)
The spaces of temperature functions over the element faces $\gamma_H$ and $\gamma_h$ are given by:

$$
\mathcal{F}_H^\Theta = \{ w | w_{\gamma_H} \in P_2(\gamma_H) \oplus B_3, \forall w_{\gamma_H} \in E(\mathcal{T}_H), w|_{\Gamma^W} = 0 \}, \quad (A.27)
$$

$$
\mathcal{F}_h^\Theta = \{ w | w_{\gamma_h} \in P_2(\gamma_h) \oplus B_3, \forall w_{\gamma_h} \in E(\mathcal{T}_h) \cap E(\mathcal{T}_H), w|_{\Gamma^W} = 0 \}. \quad (A.28)
$$

It follows that $\mathcal{F}_H \subset \mathcal{F}_h$ and $\mathcal{F}_H^\Theta \subset \mathcal{F}_h^\Theta$, the functions in these spaces can be discontinuous.

Finally, the interpolation operators for the velocity, temperature, and pressure fields are defined as $\hat{I}_u^h$, $\hat{I}_Q^h$, and $\hat{I}_p^h$, respectively. These operators interpolate the functions from the coarse mesh to the discontinuous sub-domain fine mesh tetrahedron such that

$$
(\hat{I}_u^h v) |_{T_H} (v^u) = v |_{T_H} (v^u), \quad (A.29)
$$

$$
(\hat{I}_Q^h w) |_{T_H} (v^\Theta) = w |_{T_H} (v^\Theta), \quad (A.30)
$$

$$
(\hat{I}_p^h q) |_{T_H} (v^p) = q |_{T_H} (v^p), \quad (A.31)
$$

where $v^u$ is any of the tetrahedral nodes for the velocity reference element, $v^\Theta$ is any of the tetrahedral nodes for the temperature reference element, and $v^p$ is any of the tetrahedral nodes for the pressure reference element.

The hybrid-flux interpolation operators on faces of tetrahedral elements are denoted as $\hat{I}_u^f$ and $\hat{I}_Q^f$. They interpolate the functions from the coarse mesh to the discontinuous sub-domain fine mesh face triangulation such that

$$
(\hat{I}_u^f v) |_{T_H} (v^f) = v |_{T_H} (v^f), \quad (A.32)
$$

$$
(\hat{I}_Q^f w) |_{T_H} (v^\Theta) = w |_{T_H} (v^\Theta), \quad (A.33)
$$

where $v^f$ and $v^\Theta$ are any of the triangular nodes for the velocity and temperature, respectively on a face $\gamma_h$. Note that the hybrid fluxes are only calculated between two shared faces, and are therefore only interpolated between the faces.