Parallel Computations of Finite Element Output Bounds for Conjugate Heat Transfer [*]

Zhong Cheng [†]and Marius Paraschivoiu[‡]

Abstract

This paper investigates the *A Posteriori* finite element bound method applied to a heat transfer problem in a multi-material electronic components array. The temperature field is obtained by solving Poisson equations and Convection-Diffusion equations in different regions of the computational domain. The bound method calculates very sharp lower and upper bounds of the temperature of the hottest component which is assumed to be the engineering output of interest. This paper shows that for this two-dimensional problem the bound method can yields more than an eighty-fold reduction in simulation time over a fine mesh calculation (330,050 d.o.f.) while still maintaining quantitative control over the accuracy of the engineering output of interest. Parallel implementation on a Beowulf cluster is also reported.

1 Introduction

A partial differential equation, or a set of partial differential equations, is generally used to describe the physical behavior of a system that is relevant to thermal, mechanical, fluid, or electromagnetic phenomena. For most engineering problems, however, it is impossible to yield an analytical solution

[*] accepted in Finite Elements in Analysis and Design

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

[‡] Corresponding author, Department of Mechanical and Industrial Engineering, University of Toronto, 5 King's College Road, Toronto, Ontario, Canada M5S 3G8. E-mail: marius@mie.utoronto.ca

for the underlying partial differential equation. Computer simulations offer a relatively fast, flexible and inexpensive approximation of the solution. It is well known that the accuracy of the approximation depends on the discretization size. As the size of the elements decreases, the accuracy increases but the computational cost increases as well.

In practical engineering problems, a designer has to yield an accurate simulation by using limited computer resources. The simulation is often only used to extract characteristic values (termed outputs) of the system under investigation. For example, the heat flux across a boundary, or the values of temperature at some location, is often more relevant in engineering design than the entire temperature field solution. Nevertheless the solution must be accurate to yield an accurate output. The designer is then facing a trade-off between solution accuracy and computational cost. To reconcile the computational cost and accuracy, Paraschivoiu and Patera [13, 11, 14] have developed an *a posteriori* finite element method which calculates upper and lower bounds to the output associated with a very accurate solution by solving cheap local subdomain problems.

To illustrate the cost advantage of the bound method, let us consider a problem of heat transfer in a multi-material devise represented in Fig. 1, where the temperature Θ is fixed on boundary Γ_L, and periodic on boundaries Γ_T and Γ_B, and Γ_R is insulated. The temperature field is described by a set of Poisson equations and one convection-diffusion equation. Heat generated in region Ω_2 is removed by conduction and convection. In a realistic scenario, a designer is interested in the average temperature of the heated element. The output of interest is therefore $s = \int_{\Omega_2} \Theta d\Omega$. To obtain an approximation of this output, a mesh is constructed and a solution of the temperature field is obtained. For simplicity, let us consider a mesh generated by a structured refinement of the computational domain subdivided into triangles. Fig. 2 represents a generic discretization of the computational domain into $N \times N \times 2$ elements. The output calculated on a mesh where $N = 144$ is $s_H = 0.112332$ and the elapsed computation time on a single processor is 174 seconds. The designer verifies the accuracy of this result by computing the output on a finer mesh. If the mesh is refined by a factor two in each direction ($N = 288$) then the output is $s_{H/2} = 0.112471$ and the computation time has increased to 2,563 seconds. However, if the mesh is refined by a factor four in each direction ($N = 576$) then the output is $s_{H/4} = s_h = 0.112526$ and the computation time is 39,083 seconds. While the computation times depend on the solver, here a biconjugate iterative solver with a Jacobi preconditioner, the main idea remains. The computational time increases exponentially as the discretization size decreases. A designer may decide not to calculate the output on the finest mesh however calculations on the two coarser meshes add up to 2,737 seconds. The bound method applied to bounding the output of the finest mesh ($N = 576$) offers a lower bound $s_{LB} = 0.111260$ and an upper bound $s_{UB} = 0.113513$ or presented in terms of uncertainty $s_h = 0.112387 \pm 1.00\%$. The total calculation time, i.e. both bounds, is 470 seconds; approximately three times the cost of the coarse mesh calculations or 1.2% of the fine mesh calculations. This scenario is not generic but is a good illustration of the potential of the bounds. In fact optimization of the programming of the bound calculations should reduce the computation times presented. Furthermore this scenario should be even more convincing when considering problems in higher dimensions as well as discretizations on unstructured meshes.

Machiels, Peraire, and Patera extended the bound method to address other types of partial differential equations including the Helmholtz equation, the Burgers equation and the incompressible Navier-Stokes equations [14, 15, 9]. In all previous work, the Ladevèze and Leguilon procedure is used to calculate the inter-subdomain connectivity. This procedure is very complicated to implement and hard to extend to three space dimensions. The finite element tearing and interconnecting (FETI) method is exploited herein to calculate the inter-subdomain connectivities. Farhat and Roux developed this method to decompose the computational domain for efficiently calculating parallel solutions [6]. The FETI method is based on a domain decomposition approach using Lagrange multipliers to distribute work on different processors. At present, the FETI method is widely used in structural mechanics problems because it offers robust iterative solutions for parallel processing of large-scale finite element systems. Recently, Paraschivoiu reformulated the FETI method for calculating bounds for the convection-diffusion equation in three-dimensional space [12].

In this paper, the approach described in [12] is applied to calculate the bounds for a set of partial differential equations composed of Poisson equations and convection-diffusion equations with different conductivities in a multi-material heat transfer problem. The purpose of this paper is to demonstrate the attributes of the bound method with a numerical example. The accuracy and the computational cost are presented for single processor calculations and multi-processor calculations. This is the first implementation of the bound method for parallel distributed computations. A brief discussion of the implementation and the performance is presented.

This paper is organized as follows. Section 2 contains a review of the bound formulation. Numerical results are reported in Section 3 for calculations on a single processor. In Section 4, the implementation of the method for parallel processing is developed and results are reported. Finally, a Conclusion summarizes the main observations.

2 Bounds formulation

The bound method is based upon the construction of an augmented Lagrangian, in which the objective is a quadratic "energy" reformulation of the desired output, and the constraints are the finite element equilibrium equations and the inter-subdomain continuity requirements. In this section, the bound procedure is briefly reviewed without any proof of the bounding properties. The reader may want to read [11, 10] for further details.

2.1 Problem Statement

Given a two dimensional domain, $\Omega =]0,1[\times]0,1[$, consisting of four regions $\Omega_1 =]\frac{2}{3},1[\times]0,1[$, $\Omega_2 =]\frac{1}{3},\frac{2}{3}[\times]0,\frac{1}{6}[$, $\Omega_3 =]0,\frac{2}{3}[\times]\frac{1}{2},\frac{2}{3}[$, $\Omega_4 =$ other, (Fig. 1), a problem of heat transfer is considered which is defined by the following set of equations:

$$
\begin{aligned}
-\nabla^2\Theta + U \cdot \nabla\Theta &= 0 \quad \text{in} \quad \Omega_1 && (1)\\
-100\ \nabla^2\Theta &= 10 \quad \text{in} \quad \Omega_2 && (2)\\
-100\ \nabla^2\Theta &= 0 \quad \text{in} \quad \Omega_3 && (3)\\
-\nabla^2\Theta &= 0 \quad \text{in} \quad \Omega_4 && (4)
\end{aligned}
$$

where Θ is the temperature solution, and U is the convection coefficient. Continuity of heat flux and temperature is enforced across the different regions.

Machiels, Maday and Patera [8] first described this transport problem, which is related to cooling an array of electronic chips in an electrical component system. Heat must be removed from the heated element (Ω_2) by the fluid to the higher conductivity material (Ω_3), and finally to the cold wall ($x_1 = 0$). Area Ω_4 is made up of lower conductivity material and no heat is generated in this area. All the chips are arranged side by side in an array so the boundary condition is periodical in the x_2 direction.

The boundary conditions are

$\Theta(x_1, 0) = \Theta(x_1, 1)$, $\frac{\partial\Theta}{\partial x_1}|_{x_1=1} = 0$, $\Theta(0, x_2) = 0$.

In Fig. 1, Pe is the Peclet number defined as $\frac{VL}{\alpha}$, where V is the maximum fluid velocity, L is the characteristic length and α is the thermal diffusivity. Physically, the convection coefficient U is the velocities of the cooling fluids which can be expressed as $U = Pe\ (0,\ 36(x_1 - \frac{2}{3})(1 - x_1))$.

Area Ω_2 has the highest temperature in the domain and a designer may be interested in making sure that its average temperature does not exceed an upper limit.

2.2 Generalities

In the context of the bound method, the output value associated with a fine mesh is delimited by upper and lower bounds. The term output is used to identify the characteristic quantity of interest of a physical system. The particular linear functional investigated here is the average value of the field solution, namely, the mean temperature in Ω_2, $s = \frac{1}{|\Omega_2|}\int_{\Omega_2} \Theta d\Omega$. Many relevant linear functionals can be constructed, including the value at one point or the flux over a boundary segment [9, 11, 14].

The discrete output of interest is

$$s_h = \ell_h^T \Theta_h, \tag{5}$$

where Θ_h is an n-long vector representing the discrete field solution of Equations (7) and (8) and ℓ_h^T is the discrete functional. Note that n is the number of nodes, including boundary nodes, associated with the finite element discretization of Θ. The discretization exploits the finite element vector space X_h consisting of continuous piecewise linear functions on a fine triangular structured mesh $\mathcal{T}_h$. It is assumed that X_h is a very fine approximation space such that Θ_h and s_h are high fidelity approximations of the exact solution Θ and associated output s. The unknown nodal values of Θ_h, i.e. $\tilde{\Theta}_h$, can be obtained by solving the algebraic system

$$\tilde{L_h}\tilde{\Theta}_h = \tilde{f}_h, \tag{6}$$

where $\tilde{L_h}$ is $r \times r$ non-symmetric positive-definite sparse matrix arising from the finite element discretization of the problem, and $\tilde{f_h}$ is a right-hand side r-long vector representing a prescribed force. Clearly, the number of interior degrees-of-freedom r is the number of nodes n minus the Dirichlet nodes. Recall that X_h is a high fidelity approximation space which requires a very large number of degrees-of-freedom, r, and therefore s_h requires huge computational resources. Thus, the goal is to bound s_h instead.

Let's now consider different hierarchical levels of discretizations, of the global domain Ω. To start Ω is decomposed into a coarse structured triangular mesh, $\mathcal{T}_H$. A generic example of such a mesh is presented in Fig. 3. A set of N_k uniform square subdomains $\Omega^{(k)}$ are also constructed. The bold lines in Fig. 4 represent these subdomains. Each subdomain $\Omega^{(k)}$ $\{k = 1, 2, \ldots, N_k\}$ only belongs to one material area $\{\Omega_i\}$, $\{i = 1, 2, 3, 4\}$, whereas one material area can contain several subdomains. Each square subdomain $\Omega^{(k)}$ is further decomposed into an uniform refinement of triangulations $\mathcal{T}_h$ of refinement size R ($R = \frac{H}{h}$) (Fig. 5). The linear finite element spaces associated with these meshes are X_H, $\hat{X}_H$, and $\hat{X}_h$, respectively.

2.3 Bound Calculations Review

The bound method is based on the constrained minimization problem formulated as an extremization of an augmented Lagrangian [12]. A Lagragian is constructed such that the quadratic objective vanishes when the field variable satisfies the equilibrium equation ensuring that the saddle point equals the value of the output. In addition, a domain decomposition of the problem is introduced as a constrain to reduce computation cost. The discrete form of this Lagrangian for a generic discretization δ of the model problem described in Section 2.1 is

$$\begin{aligned}
&\mathcal{L}(\Theta_\delta, \mu_\delta, \lambda_\delta) = \\
&\qquad \sum_{k=1}^{N_k} \left(\Theta_\delta^{(k)^T} A_\delta^{(k)} \Theta_\delta^{(k)} - f_\delta^{(k)^T} \Theta_\delta^{(k)} + \ell_\delta^{(k)^T} \Theta_\delta^{(k)} \right) \qquad (7) \\
&\qquad - \sum_{k=1}^{N_k} \mu_\delta^{(k)^T} \left(L_\delta^{(k)} \Theta_\delta^{(k)} - f_\delta^{(k)} \right) + \lambda_\delta{}^T \sum_{k=1}^{N_k} B_\delta^{(k)} \Theta_\delta^{(k)}.
\end{aligned}$$

The field variable Θ_δ is the solution of Equation (1) through (4) and constraints are enforced using μ_δ, the adjoint, and λ_δ, the inter-subdomain connectivity. For N_k subdomains, $\Theta_\delta = \{\Theta_\delta^{(1)}, \ldots, \Theta_\delta^{(N_k)}\}$ and $\mu = \{\mu_\delta^{(1)}, \ldots, \mu_\delta^{(N_k)}\}$. Here, $A_\delta^{(k)}$ is the finite element discretization of the symmetric part of $L_\delta^{(k)}$ and $B_\delta^{(k)}$ is the sign Boolean matrix which localizes the "jumps" at the interface.

Note that

$$\sum_{k=1}^{N_k} \left({\Theta_\delta^{(k)}}^T A_\delta^{(k)} \Theta_\delta^{(k)} - {f_\delta^{(k)}}^T \Theta_\delta^{(k)} + {\ell_\delta^{(k)}}^T \Theta_\delta^{(k)} \right)$$

represents the objective which is composed of the energy equality and the output. The local equilibrium equations, $L_\delta^{(k)} \Theta_\delta^{(k)} - f_\delta^{(k)} = 0$, and the inter-subdomain continuity condition, $\sum_{k=1}^{N_k} B_\delta^{(k)} \Theta_\delta^{(k)} = 0$, are enforced using the adjoint μ_δ and the inter-subdomain connectivity λ_δ.

The current bound method is based on a two level hierarchical approximations. The first level calculation is a relatively inexpensive approximation to find the candidate Lagrange multipliers, μ_H and λ_H , on a coarse mesh $\mathcal{T}_H$ with an element size $\delta = H$. The second level calculation is on a fine mesh $\mathcal{T}_h$ with an element size h , on which the coarse mesh Lagrange multipliers μ_H and λ_H are linearly interpolated to the fine mesh. It is shown in [11] that $s_h = \mathcal{L}_h(\Theta_h, \mu_h, \lambda_h)$ if the Lagrange multipliers and the field solution Θ_h are calculated on the mesh $\mathcal{T}_h$ and $s_{LB} \equiv \mathcal{L}_h(\hat{\Theta}_h, \mu_H, \lambda_H) \leq s_h$ if the Lagrange multipliers are calculated on any coarse mesh denoted by $\mathcal{T}_H$ and interpolated on a mesh $\mathcal{T}_h$. It is important to realize that s_h is the output that is bounded, not s. To calculate the upper bound, the Lagrangian is modified ($\mathcal{L}_h^{UP}$) such that the output term, ${\ell_\delta^{(k)}}^T \Theta_\delta^{(k)}$, is subtracted instead of being added. The saddle point value of this Lagrangian becomes minus the output value: $-s_h = \mathcal{L}_h^{UP}(\Theta_h^{UP}, \mu_h^{UP}, \lambda_h^{UP})$. Applying the same principles as above leads to $-s_{UP} \equiv \mathcal{L}_h^{UP}(\hat{\Theta}_h^{UP}, \mu_H^{UP}, \lambda_H^{UP}) \leq -s_h$ or simply $s_{UP} \geq s_h$.

The procedure for the upper bound and lower bound is similar. For simplicity, only the lower bound procedure is presented. The four main computational steps are as follows:

- **a) Coarse mesh calculations:** By evoking stationarity of the Lagrangian (7) associated with $\mathcal{T}_H$ without any domain decomposition, the following equations are obtained:

$$\begin{aligned} L_H \Theta_H &= f_H && (8) \\ L_H^T \mu_H &= 2A_H \Theta_H - f_H + \ell_H && (9) \end{aligned}$$

The field solution, Θ_H, and the adjoint, μ_H, are calculated by solving (8) and (9) respectively on this coarse mesh. Now, evoking stationarity of the same Lagrangian on a decomposed problem leads to:

$$
\begin{aligned}
L_H \Theta_H &= f_H && (10)\\
2A_H \Theta_H &= f_H - \ell_H + L_H^T \mu_H - B_H^T \lambda_H && (11)\\
B_H \Theta_H &= 0 && (12)
\end{aligned}
$$

The inter-subdomain connectivity, λ_H, is obtained by constructing and solving a dual problem from (11) and (12). For the bound method the dual interface problem [12] leads to:

$$
\begin{bmatrix} 2F_H & -G_H \\ -G_H^T & 0 \end{bmatrix} \begin{bmatrix} \lambda_H \\ \alpha_H \end{bmatrix} = \begin{bmatrix} 2d_H \\ -e_H \end{bmatrix}, \qquad (13)
$$

where each of these terms is given by

$$
\begin{aligned}
F_H &= \sum_{k=1}^{N_k} B_H^{(k)} A_H^{(k)^+} B_H^{(k)^T}, && (14)\\
G_H &= \left[B_H^{(1)}[1] \;\; \ldots \;\; B_H^{(N_k)}[1] \right], && (15)\\
\alpha_H &= \left[\alpha_H^{(1)} \;\; \ldots \;\; \alpha_H^{(N_k)} \right], && (16)\\
d_H &= \sum_{k=1}^{N_k} B_H^{(k)} A_H^{(k)^+} q_H^{(k)}, && (17)\\
e_H &= \left[[1]^T q_H^{(1)} \right] \;\; \ldots \;\; \left[[1]^T q_H^{(N_k)} \right], && (18)
\end{aligned}
$$

where $A_H^{(k)^+}$ is a generalized inverse of $A_H^{(k)}$ and $q_H^{(k)} = f_H^{(k)} - \ell_H^{(k)} + L_H^{(k)^T} \mu_H^{(k)}$. The constraint $G_H^T \lambda_H = e_H$ ensures that $(q_H^{(k)} - B_H^{(k)^T} \lambda_H) \in$ range $A_H^{(k)}$ for all $k = 1, ..., N_k$.

A preconditioned conjugate gradient algorithm is used to solve the interface problem. A projection operator is exploited to handle the self-equilibrium constraints. The FETI method iterates on λ_H, given an initial λ_H^0 which satisfies the constraint $G_H^T \lambda_H^0 = e_H$. The remainder,

$\Delta\lambda_H = \lambda_H - \lambda_H^0$, satisfies

$$\begin{aligned} P_H F_H \Delta\lambda_H &= P_H(d_H - F_H \lambda_H^0), & (19) \\ G_H^T \Delta\lambda_H &= 0, & (20) \end{aligned}$$

where P_H is an orthogonal projector onto $\mathrm{Ker}(G_H^T)$ defined as

$$P_H = I_H - G_H(G_H^T G_H)^{-1} G_H^T. \tag{21}$$

The FETI algorithm can be viewed as a two-step preconditioned conjugate gradient method to solve the interface problem and can be found abundantly in the literature [1, 2, 5, 7, 3, 16]. This algorithm can be summarized as in [4]:

1. Initialize

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

2. Iterate

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

The FETI iterations residual obeys the following stopping criterion

$$\frac{\|\tilde{F}_H^{-1} w^n\|_2}{\|q_H^n\|_2} \leq \epsilon_g, \tag{22}$$

where ϵ_g is the global FETI tolerance. In this study, ϵ_g is set to 10^{-3}.

- **b) Interpolation:** Both Lagrange multipliers, μ_H and λ_H, are linearly interpolated onto the fine mesh $\mathcal{T}_h$, i.e., $\mu_H \rightarrow \mu_h$; $\lambda_H \rightarrow \lambda_h$.

- **c) Fine mesh calculations:** The only remaining calculations before obtaining the lower bound are decoupled symmetric solves in each subdomain to find $\hat{\Theta}_h^{(k)}$.

$$2A_h^{(k)}\hat{\Theta}_h^{(k)} = f_h^{(k)} - \ell_h^{(k)} + L_h^{(k)^T}\mu_h^{(k)} - B_h^{(k)^T}\lambda_h, \qquad k = 1 \dots N_k. \tag{23}$$

- **d) Bound calculations:** The lower bound of the output of interest, s_{LB}, is calculated from:

$$s_{LB} = \sum_{k=1}^{N_k} \left(\hat{\Theta}_h^{(k)^T} A_h^{(k)} \hat{\Theta}_h^{(k)} - f_h^{(k)^T} \hat{\Theta}_h^{(k)} + \ell_h^{(k)^T} \hat{\Theta}_h^{(k)} \right) - \sum_{k=1}^{N_k} \mu_h^{(k)^T} \left(L_h^{(k)} \hat{\Theta}_h^{(k)} - f_h^{(k)} \right) + {\lambda_h}^T \sum_{k=1}^{N_k} B_h^{(k)} \hat{\Theta}_h^{(k)}, \tag{24}$$

Taking account of (23), the lower bound is simply written as:

$$s_{LB} = -\sum_{k=1}^{N_k} \hat{\Theta}_h^{(k)^T} A_h^{(k)} \hat{\Theta}_h^{(k)} + \sum_{k=1}^{N_k} f_h^{(k)^T} \mu_h^{(k)}, \tag{25}$$

3 Numerical Results

The numerical results will show rigorous, accurate and quantitative bounds for the engineering quantity of interest: the average temperature of the heated element Ω_2. For comparison, the solution on the fine mesh is calculated, from which the corresponding output is evaluated. This output is called the "truth" output because it is assumed to have negligible difference from the exact solution of the PDE. For this problem, the truth mesh is a uniform refinement of the computational domain into 576×576 squares further subdivided into triangles (576×576 × 2 elements). Fig. 6, 7, and 8 show isocontours of temperature (15 isocontours from 0.00 to 0.14) for different Peclet numbers. For $Pe = 0$, or no convection term, the temperature profile is symmetric along $x_2 = \frac{7}{12}$. The convection effect becomes stronger with increasing Peclet number, and the temperature field becomes asymmetric. In Ω_2 and Ω_3, where the conductivity is very high, the temperature is nearly unchanged. As expected, the results show that the average temperature of the heated element Ω_2 decreases as the Peclet number increases.

The convergence of the bound is investigated by calculating the bounds on different coarse meshes. The global computational domain is divided into $6{\times}6 = 36$ subdomains with a different refinement in each subdomain. Calculations are also carried out for Peclet numbers of 0, 50 and 100. Fig. 9, 10 and 11 show the upper and lower bounds vs. the ratio of coarse mesh dimension to the fine mesh, namely, $R = \frac{H}{h}$. Recall that the fine mesh size h is fixed in all simulations as well as the number of subdomains. The upper and lower bounds are normalized as $s_{UP} = \frac{s_{UB}}{s_h}$ and $s_{LOW} = \frac{s_{LB}}{s_h}$ respectively, where s_h is the output on the "truth" mesh. These charts show that the bound method performs better for small Peclet numbers. The bound gap, i.e. the upper bound minus the lower bound, is also highly dependent on the coarse mesh. A finer coarse mesh leads to sharper bounds. This convergence has been shown to decrease quadratically with the size of the coarse mesh, H [10].

Table 1 shows both the memory and the simulation time (CPU) associated to the different coarse mesh sizes, as well as the corresponding half bound gap. The CPU time decreases drastically as a coarser mesh is exploited. A relatively good approximation of the "truth" output can be obtained with less than 2% of the CPU time and with 15% of the memory if the bound method is performed on a coarse mesh of size $36 \times (24 \times 24)$ instead of a general finite element calculation on the fine mesh. The method provides 100% certainty that the output value, as would be computed on the finest mesh, is between the lower bound $s_{LB} = 0.11189$ and the upper bound $s_{UB} = 0.11381$ or presented in terms of uncertainty $s_h = 0.11285 \pm 0.85\%$ for the most difficult case, $Pe = 100$.

The above results were obtained on a Pentium III 800 MHz with SunOS 5.8. The Jacobian preconditioned Biconjugate iterative solver was used to solve the underlying nonsymmetric equations. Conjugate Gradient was used to solve symmetric linear equations.

The efficiency of the bound method can be further improved by optimizing the domain decomposition. For analysis the global computational domain is divided into $6{\times}6 = 36$ subdomains, $12{\times}6 = 72$ subdomains, or $12{\times}12 = 144$ subdomains with a different refinement in each subdomain. Table 2 reflects the computation time with respect to the number of subdomains. For a coarse mesh of 144×144 subdivided into 144 subdomains,

Resources vs. Refinement				Half bound gap (%)		
Mesh	Refinement	Memory	CPU	Pe=0	Pe=50	Pe=100
576 × 576	Truth mesh	296M	39000s	0.00	0.00	0.00
36×(48×48)	2	131M	6400s	0.13	0.20	0.31
36×(32×32)	3	75M	2207s	0.28	0.41	0.55
36×(24×24)	4	42M	1332s	0.45	0.64	0.85
36×(16×16)	6	28M	1097s	0.85	1.22	1.75
36×(12×12)	8	24M	1053s	1.33	1.96	3.07
36 × (8×8)	12	24M	1049s	2.56	4.19	7.78

Table 1: Computational resources used by the bound method and corresponding half bound gap

Mesh vs. CPU (sec.)				Half bound gap (%)		
Subdomains × Local Mesh	Coarse	Fine	Total	Pe=0	Pe=50	Pe=100
(6 × 6) × (24 × 24)	308	1024	1332	0.45	0.64	0.85
(12 × 6) × (12 × 24)	294	361	655	0.50	0.70	0.97
(12 × 12) × (12 × 12)	295	185	470	0.50	0.69	1.00

Table 2: Computation time and half bound gap for different domain decompositions

the bound calculations cost only 470 seconds of CPU time. With the domain decomposed into $6 \times 6 = 36$ subdomains the CPU time increases to 1,330 seconds. Decomposing into more subdomains reduces the CPU time required for the fine mesh calculations since the subdomain problems become smaller while the coarse mesh calculations remain the same except for the FETI calculations. As a result, the total CPU time of the bound method also decreases. Also note that the fine mesh calculations become less expensive than the coarse mesh calculations.

4 Parallel Calculations

In the previous section it was shown that the bound method offers a significant computational advantage over the general finite element method on a single processor. In this section, a discussion of its computational advantage

Proc.	Elapsed Time (sec) $36 \times (24 \times 24)$		
#	Coarse Cal.	Fine Cal.	Total
1	292	1040	1332
2	158	558	716
3	114	352	466
4	91	266	357
6	70	178	248
9	71	116	187
12	80	88	168

Table 3: Performance results for increasing number of processors

on multi-processors is opened. Parallel results reported here were obtained on a Beowulf cluster of Pentium III 800MHz/756M memory computers, connected with a 10/100 Mbits/sec HUB, running on SunOS 5.8 operating system. The communication uses the Message Passing Interface (MPI) available through MPICH library.

Implementation of the bound method requires parallelization of two main tasks. The first task is the coarse mesh calculations. It involves calculation of the temperature solution, the adjoint solution and the inter-subdomain connectivity. The second task is the fine mesh calculations which are decoupled problems and are of intrinsic parallel nature. Therefore, subdomain problems are calculated separately on different processors at the same time. Communication between processors is very small, translating into large parallel efficiency. Table 4 shows that the fine mesh calculation can reach efficiencies as high as 99% for the case of 36 subdomains distributed on 9 processors. Generally, the bound method achieves high efficiencies for this task even on a Beowulf cluster. This characteristics indicates the high potential of extending the bound method to parallel computations.

The parallelization of the coarse mesh calculations uses a standard parallel iterative solver library developed by Sandia National Laboratories for solving linear sparse matrix systems, AZTEC [17]. The distributed modified sparse row (DMSR) techniques recommended by AZTEC is used to distribute the global matrices to each processor. Regarding the FETI method, it has been proven to be an efficient parallel semi-iterative solver, however optimization of the method for the problem addressed herein was not performed. Let us

Proc.	Efficiency (%) $36 \times (24 \times 24)$			
#	Coarse Cal.	Fine Cal.	Total	Speed up
2	92.4	93.2	93.0	1.86
3	85.4	98.2	95.2	2.86
4	80.2	97.5	93.2	3.73
6	69.5	97.3	89.3	5.35
9	45.7	99.1	79.1	7.11
12	30.4	98.3	66.0	7.92

Table 4: Parallel efficiency and speed up

analyse the computational performance of the FETI algorithm. Observe that F_H contains the sum of the inverse of each subdomain problem which leads to easy parallelization. The term $F_H p^n$ is a matrix-vector operation that is performed at every iteration and can be distributed on different processors. Note that on the coarse mesh, the size of all matrices $A_H^{(k)}$ is small. The cost associated with the calculation of the interface problem also depends on the coarse problem arising at each FETI iteration. This problem is of the form

$$(G_H^T G_H)x = b, \tag{26}$$

where x and b denote the generic left-hand and right-hand side vectors. This problem is global and not easily parallelized. However, the number of degrees-of-freedom of the system in Equation (26) is equal to the total number of subdomains, in this case, either $6 \times 6 = 36$, $12 \times 6 = 72$ or $12 \times 12 = 144$. This leads to a very small system. Parallel efficiency of the FETI method is theoretically and numerically demonstrated in the literature [5, 4]. An "in house" parallel implementation is used for the computations presented in this paper. A more efficient parallel implementation is nevertheless possible and will be developed in the near future.

Under the hardware environment mentioned above, the parallel efficiency obtained from the coarse mesh calculation is satisfactory for a small number of computers (less than 4 processors), but it becomes poor when more processors are involved. For the coarse mesh calculations the efficiency drops from 92% on 2 processors to 30% on 12 processors, affecting the overall efficiency which decrease from 93% to 66%, respectively. Since the message passing between processors for these calculations is huge, upgrading the hub to a high speed switch can greatly reduce the communication time between processors, therefore improving the parallel efficiency. Obviously the size of the coarse mesh is also too small for achieving high parallel efficiencies. For three-dimensional spaces and for large-scale industrial applications, the coarse mesh will be bigger and higher parallel efficiencies are expected.

5 Conclusion

The accuracy of numerical solutions is directly related to the discretization size of the mesh. An over-refined mesh requires extremely large computer memory and long computation time whereas the over-coarsened mesh does not provide enough accuracy. The method presented herein bounds the output on the fine mesh by carrying out calculations on a course mesh so as to save computational resources. Single processor computations show that the bound method is very fast, i.e. a few percent of the computational cost required on a fine mesh. The computation time can be further reduces on multi-processors, however, low parallel efficiency of the coarse problem must be addressed.

Acknowledgments

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC).

References

[1] C. Farhat. A Lagrange multiplier based on divide and conquer finite element algorithm. *J. Comput. System Engrg.*, 2:149–156, 1991.

[2] C. Farhat, P.-S. Chen, and J. Mandel. A scalable lagrange multiplier based domain decomposition method for implicit time-dependent problems. *Int. J. Numer. Methods. Engrg.*, 38:3831–3854, 1995.

[3] C. Farhat, P.-S. Chen, F. Risler, and F.-X. Roux. A unified framework for accelerating the convergence of iterative substructuring methods with Lagrange multipliers. *Int. J. Numer. Meth. Engrg.*, 42:257–288, 1998.

[4] C. Farhat, K. Pierson, and M. Lesoinne. The second generation FETI methods and their application to parallel solution of the large-scale linear and geometrically non-linear structural analysis problems. *Comput. Methods Appl. Mech. Engrg.*, 184:333–374, 2000.

[5] C. Farhat and F.-X. Roux. A method of finite tearing and interconnecting and its parallel solution algorithm. *Comput. Methods. Appl. Mech. Engrg.*, 32:1205–1227, 1991.

[6] C. Farhat and F.-X. Roux. An unconventional domain decomposition method for efficient parallel solution of large-scale finite element systems. *SIAM J. Sci. Stat. Comput.*, 13(1):379–396, 1992.

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

[8] L. Machiels, M. Maday, and A.T. Patera. Output bounds for reduced-order approximations of elliptic partial differential equations. *Comput. Methods Appl. Mech. Engrg.*, 190:3413–3426, 2001.

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

[10] Y. Maday and A.T. Patera. Numerical analysis of *a posteriori* finite element bounds for linear-functional outputs. *Math. Models and Meth. in Appl. Sci.*, 10 (5):785–799, 2000.

[11] M. Paraschivoiu. *A Posteriori Finite Element Bounds for Linear–Functional Outputs of Coersive Partial Differential Equations and of the Stokes Problem.* PhD thesis, Massachusetts Institute of Technology, 1997.

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

[13] M. Paraschivoiu and A. T. Patera. A hierarchical duality approach to bounds for the outputs of partial differential equations. *Comp. Meth. Appl. Mech. Engrg.*, 158:389–407, 1998.

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

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

[16] D. Rixen. *Substructuring and dual methods in structural analysis.* PhD thesis, University of Liège, Belgium, 1997.

[17] R. S. Tuminaro, M. Heroux, S. A. Hutchinson, and J. N. Shadid. Official aztec user's guide: Version 2.1. Technical report, Sandia National Laboratories, 1999.

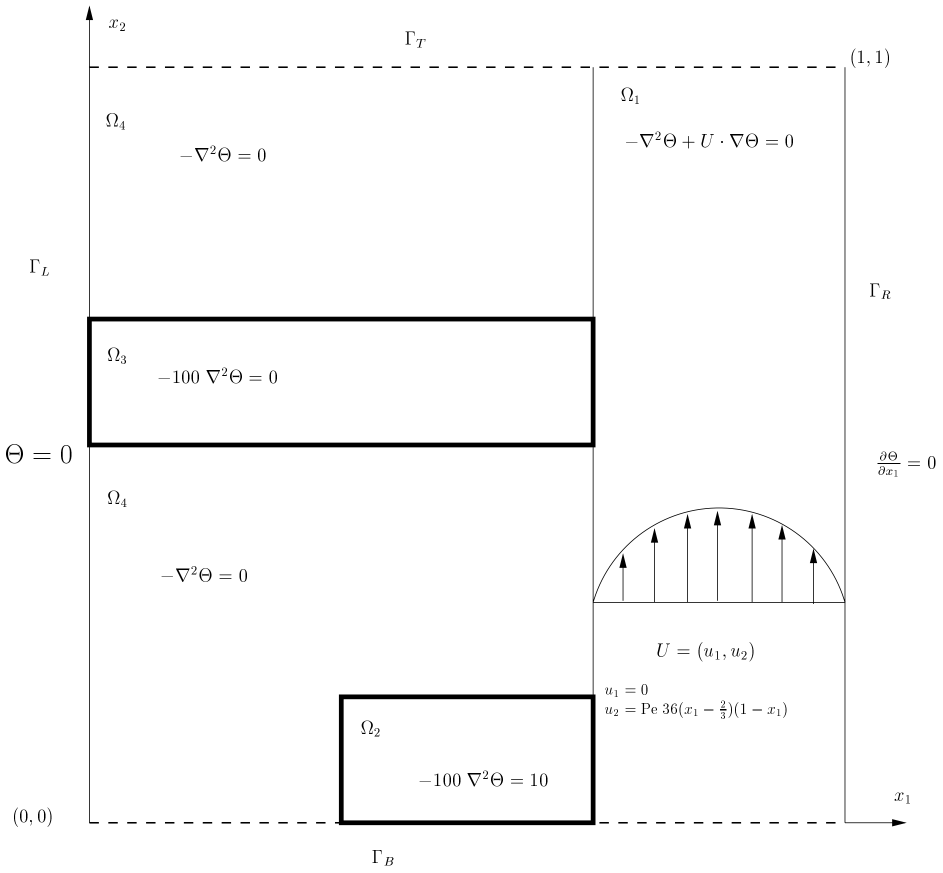

Figure 1: Diagram of each chip.

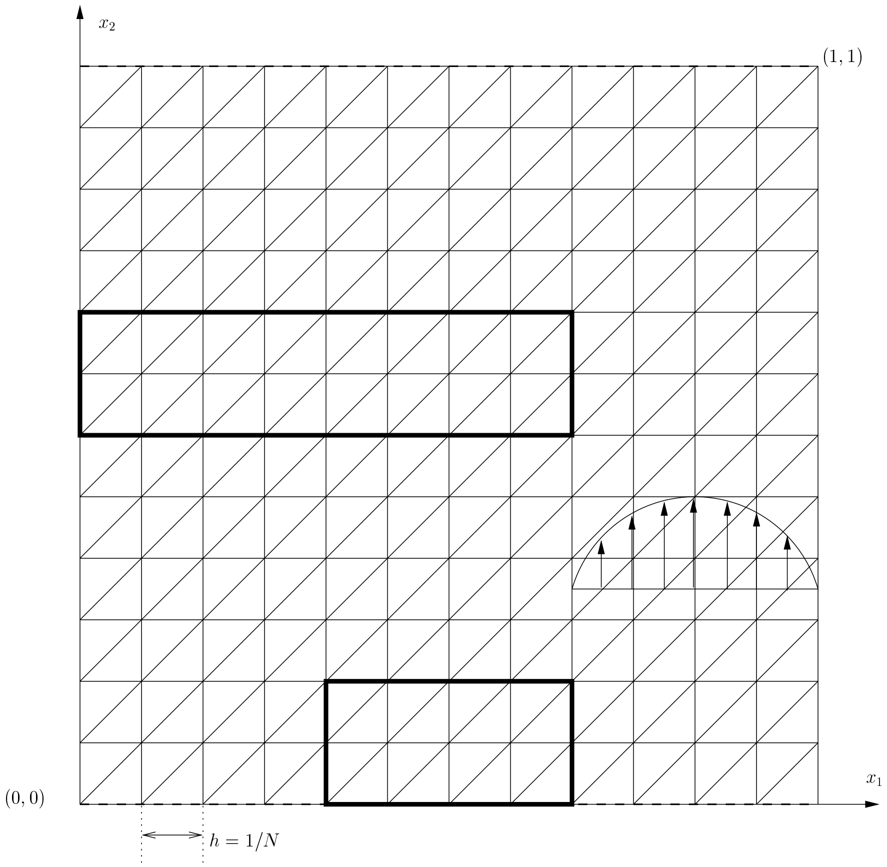

Figure 2: Generic $N \times N \times 2$ mesh.

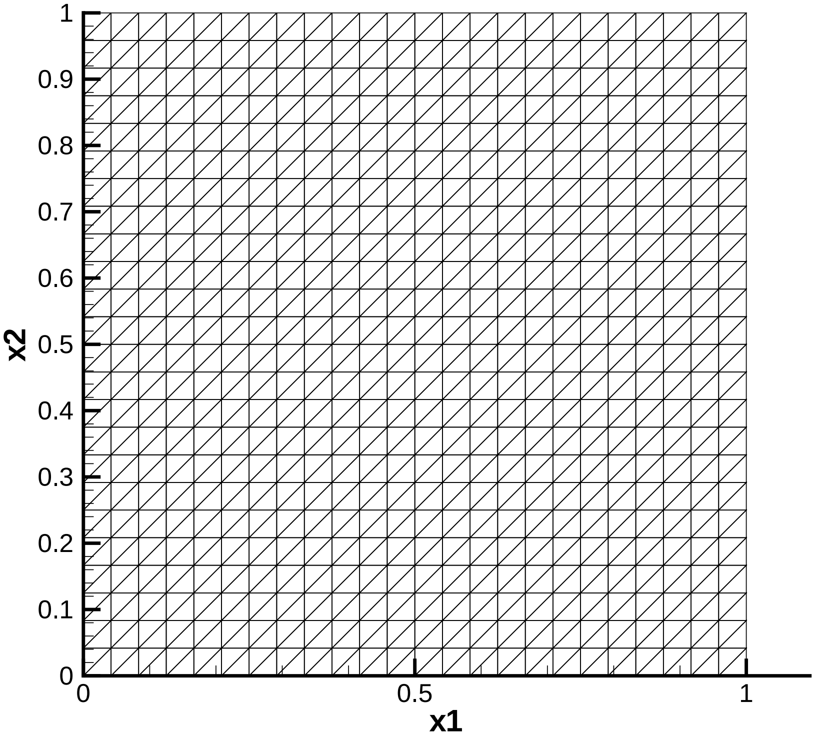

Figure 3: Coarse mesh before domain decomposition.

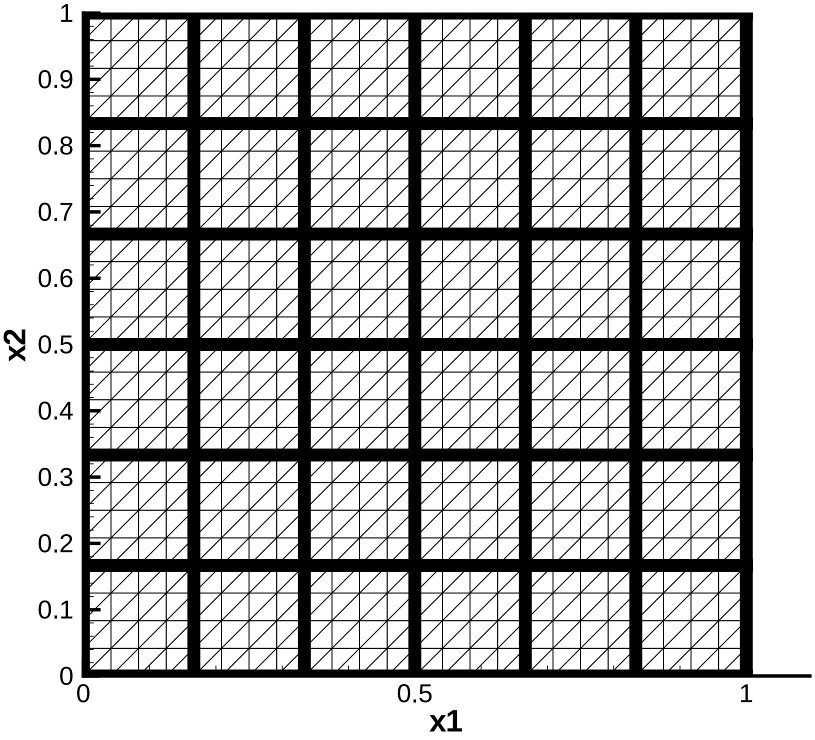

Figure 4: Domain decomposition of the coarse mesh: N_k uniform square subdomains $\Omega^{(k)}$.

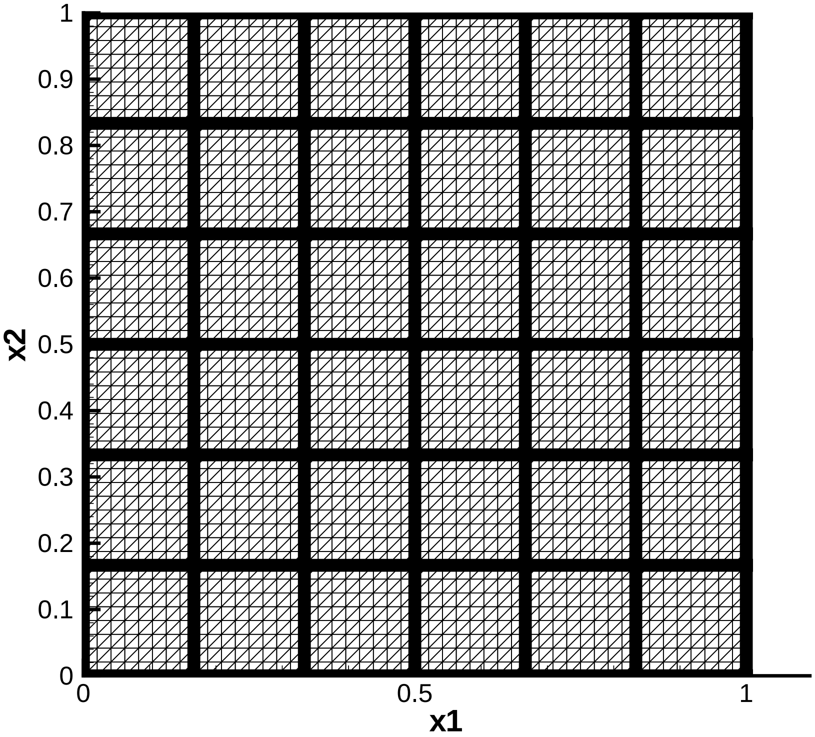

Figure 5: Domain Decomposition of the fine mesh.

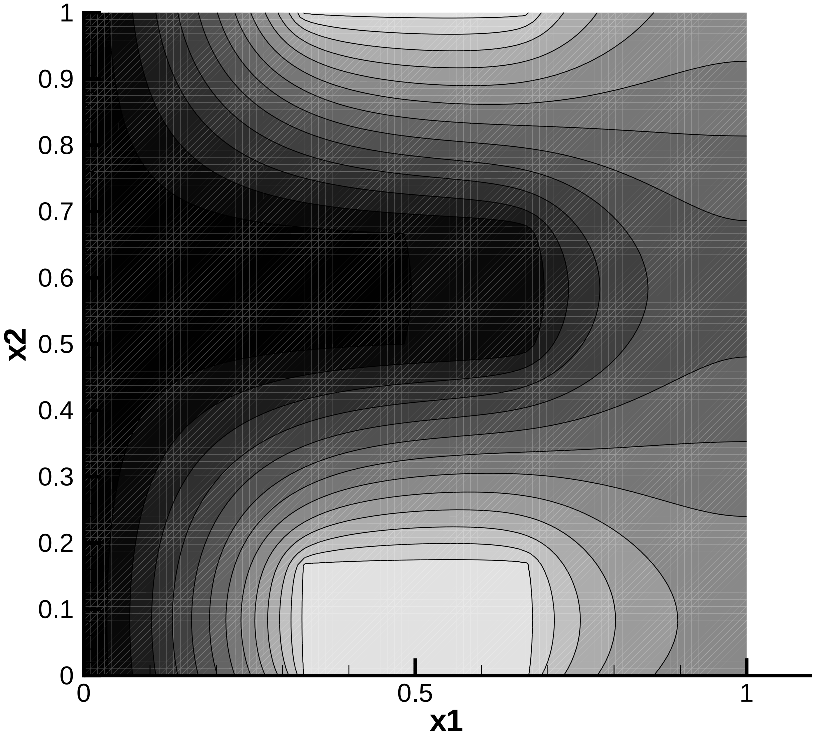

Figure 6: Temperature field solution for $Pe = 0$.

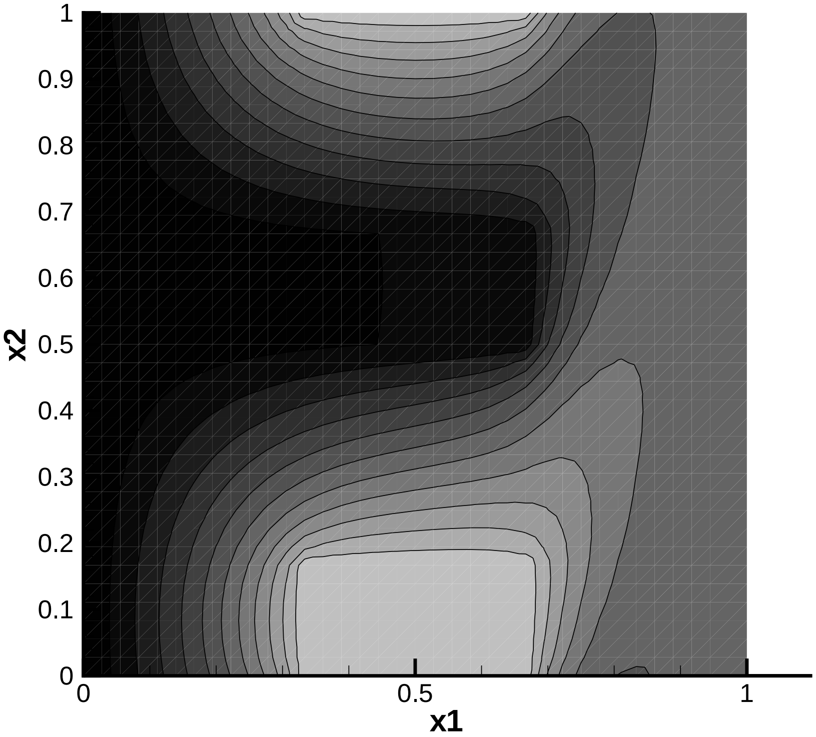

Figure 7: Temperature field solution for $Pe = 50$.

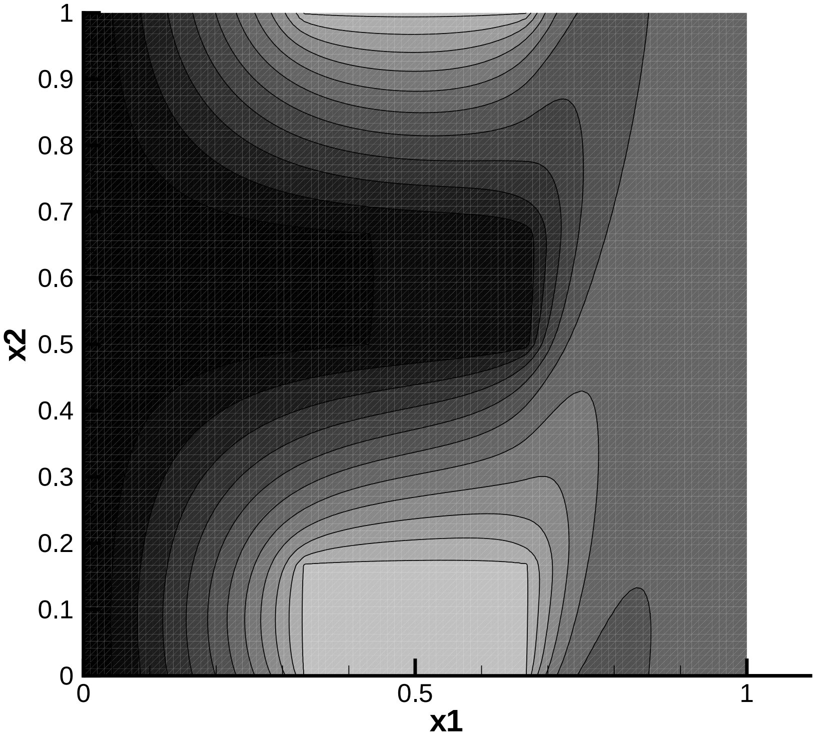

Figure 8: Temperature field solution for $Pe = 100$.

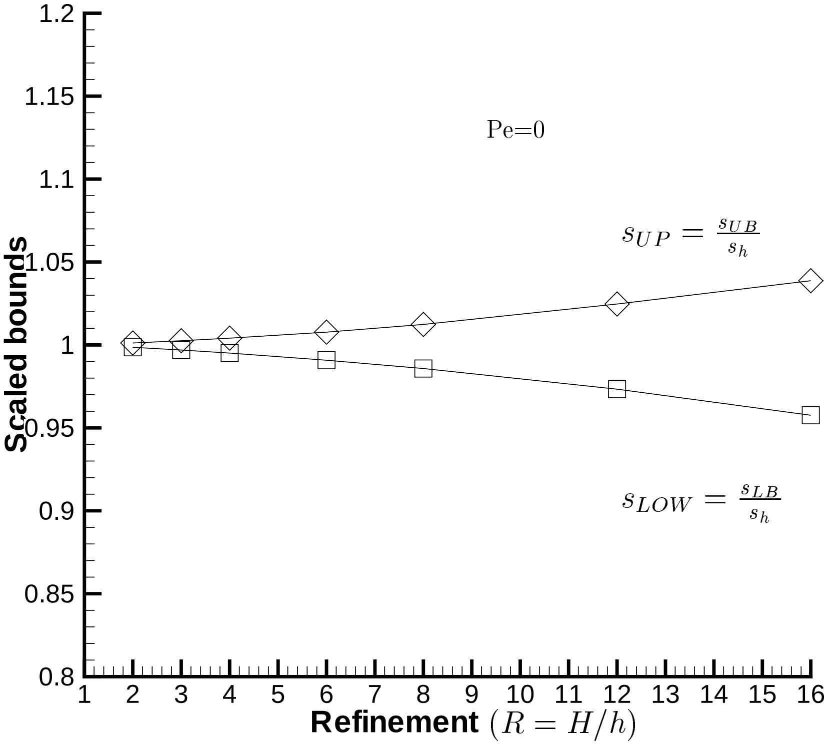

Figure 9: Convergence of the scaled bounds for a decomposition into 6×6 subdomains; $Pe = 0$.

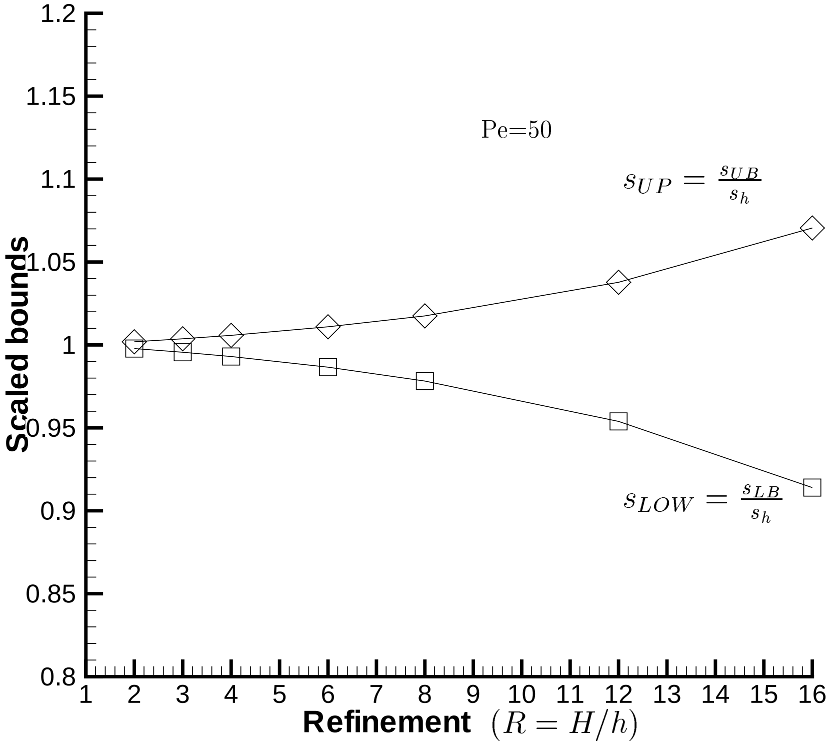

Figure 10: Convergence of the scaled bounds for a decomposition into 6×6 subdomains; $Pe = 50$.

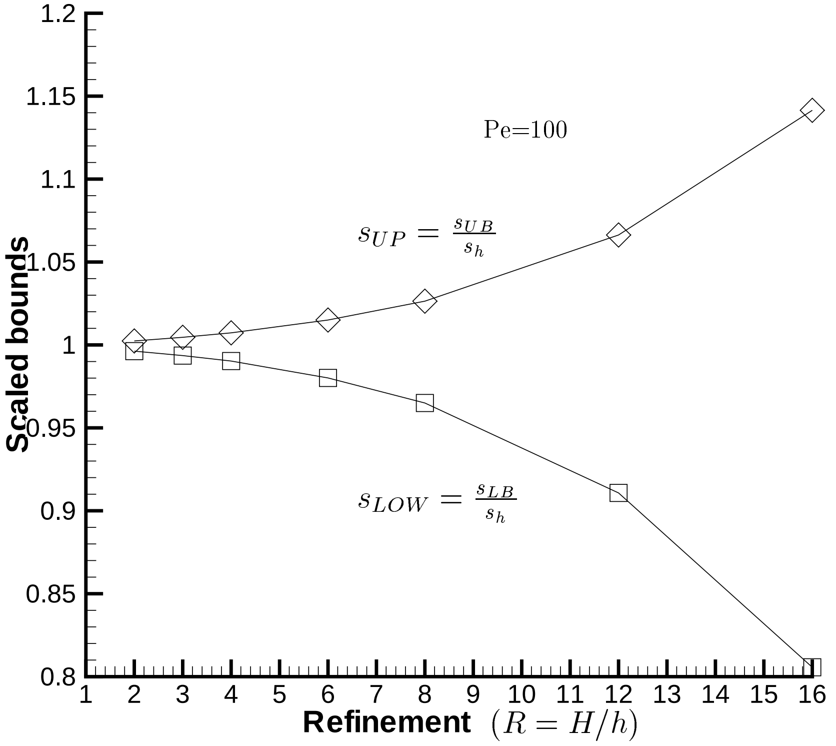

Figure 11: Convergence of the scaled bounds for a decomposition into 6 × 6 subdomains; $Pe = 100$.