Hindawi Publishing Corporation
Abstract and Applied Analysis
Volume 2013, Article ID 161873, 13 pages
http://dx.doi.org/10.1155/2013/161873

Research Article

A Pressure-Stabilized Lagrange-Galerkin Method in a Parallel Domain Decomposition System

Qinghe Yao and Qingyong Zhu

School of Engineering, Sun Yat-Sen University, 510275 Guangzhou, China

Correspondence should be addressed to Qingyong Zhu; mcszqy@mail.sysu.edu.cn

Received 30 April 2013; Accepted 14 June 2013

Academic Editor: Santantu Saha Ray

A pressure-stabilized Lagrange-Galerkin method is implemented in a parallel domain decomposition system in this work, and the new stabilization strategy is proved to be effective for large Reynolds number and Rayleigh number simulations. The symmetry of the stiffness matrix enables the interface problems of the linear system to be solved by the preconditioned conjugate method, and an incomplete balanced domain preconditioner is applied to the flow-thermal coupled problems. The methodology shows good parallel efficiency and high numerical scalability, and the new solver is validated by comparing with exact solutions and available benchmark results. It occupies less memory than classical product-type solvers; furthermore, it is capable of solving problems of over 30 million degrees of freedom within one day on a PC cluster of 80 cores.

1. Introduction

The Lagrange-Galerkin method raises wide concern about the finite-element simulation of fluid dynamics. Based on the approximation of the material derivative along the trajectory of fluid particle, the method is natural in the simulation to physical phenomena, and it is demonstrated to be unconditionally stable for a wide class of problems [1–5]. A number of researches about the Lagrange-Galerkin method were done in the case of single processor element (PE) (cf. [6–8]); the symmetry of the matrices and good stability of the scheme were reported; using a numerical integration based on a division of each element, Rui and Tabata [9] developed a second scheme for convection-diffusion problem; Massarotti et al. [10] used a second-order characteristic curve method, and a special iteration was used to keep the symmetry of the stiffness matrix. The Lagrange-Galerkin method uses an implicit time discretization, and therefore an element searching algorithm is necessary to implement it. The element searching may become very expensive when the geometry is complicated or the mesh size is very small. Due to its doubtable efficiency and feasibility for complex simulations in the case of single PE, rare research has been done to implement it in parallel, by which the enormous computation power enables us to solve more challenging simulation problems.

The present study is concentrated on improving the solvability of the Lagrange-Galerkin method on large scale and complex problems by domain decompositions. Piecewise linear interpolations are thus employed for velocity, pressure, and temperature; therefore, the so-called inf-sup condition [11] should be satisfied, which is the first difficulty to be overcome in this work. Stabilization methods for incompressible flow problems were reported by many researchers (cf. [12–15]). Park and Sung proposed a stabilization for Rayleigh-Bénard convection by using feedback control [16]; for consistently stabilized finite element methods, Barth et al. classified the stabilization techniques and studied influence of the stabilization parameter in convergence [17]; Bochev et al. stated the requirements on choice of stabilization parameter if time step and mesh are allowed to vary independently [18]. As far as we know, it may not be enough to investigate what stabilization techniques are efficient for nonsteady and nonlinear flow problems approximated by Lagrange-Galerkin methods in a domain decomposition system, where the interface problem can be solved by preconditioned conjugate gradient (PCG) method. In this paper, a pressure-stabilization method, which keeps the symmetry of the linear system and is effective for high Reynolds number and Rayleigh number simulations, is introduced to implement the Lagrange-Galerkin method in a domain decomposition system.

The element searching algorithm in a domain decomposition system using unstructured grids is the second difficulty to implement the Lagrange-Galerkin method in a domain decomposition system (cf. [5, 19]). Minev et al. reported an optimized binary searching algorithm for single PE by storing the necessary data structures in a way similar to the CSR compact storage format; however, the element information data is stored distributedly in the domain decomposition system by the skyline format, and a different way needs to be found to overcome the extra difficulty caused by the parallel computing algorithm. This step is critical, in the sense that it can be very computationally expensive and can thus make the entire algorithm impractical.

The remainder of this paper is organized as follows: in Section 2, the formula of the governing equation and the pressure-stabilization Lagrange-Galerkin method is described; Section 3 focuses on the parallel implementation of this scheme. Numerical results and comparisons with classical asymmetric product type methods in [20] are shown in Section 4. Conclusions are drawn in Section 5.

2. Formulation

2.1. The Governing Equations. Let Ω be a three-dimensional polyhedral domain with the boundary $\partial\Omega$. The conservation equations of mass and momentum are governed by

$$\begin{gathered}\frac{\partial u}{\partial t} + (u \cdot \nabla) u - 2\nu \nabla \cdot D(u) + \nabla p = f^{\text{buoyancy}} \\ \text{in } \Omega \times (0, \bar{t}), \\ \nabla \cdot u = 0 \quad \text{in } \Omega \times (0, \bar{t}), \\ u = \hat{u} \quad \text{on } \Gamma_1 \times (0, \bar{t}), \\ \sum_{j=0}^{3} \sigma_{ij} n_j = 0 \quad \text{on } \frac{\partial\Omega}{\Gamma_1 \times (0, \bar{t})}, \\ u = u_0 \quad \text{in } \Omega, \text{ at } t = 0,\end{gathered} \tag{1}$$

where $\Gamma_1 \subset \partial\Omega$ and

$$f^{\text{buoyancy}} = \beta (T_r - T) g \tag{2}$$

is the gravity force per unit mass derived on the basis of Boussinesq approximation. g is the gravity [m/s^2], β, T, and T_r are the thermal expansion coefficient [1/K], the temperature [K], and the reference temperature [K], and u, t, ν, and p are velocity vector [m/s], time [s], kinematic viscosity coefficient [m^2/s], and kinematic pressure [m^2/s^2], respectively. σ_{ij} is the stress tensor [N/m^2] defined by

$$\begin{gathered}\sigma_{ij}(u, p) \equiv -p\delta_{ij} + 2\nu D_{ij}(u), \\ D_{ij}(u) \equiv \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right), \quad i, j = 1, 2, 3,\end{gathered} \tag{3}$$

with the Kronecker delta δ_{ij}.

The fluid is assumed to be incompressible according to Boussinesq approximation, and the density is assumed to be constant except in the gravity force term where it depends on temperature according to the indicated linear law; see (2). The energy equation is

$$\begin{gathered}\frac{\partial T}{\partial t} + u \cdot \nabla T - a\Delta T = S \quad \text{in } \Omega \times (0, \bar{t}), \\ T = \hat{T} \quad \text{on } \Gamma_2 \times (0, \bar{t}), \\ a\frac{\partial T}{\partial n} = 0 \quad \text{on } \frac{\partial\Omega}{\Gamma_2 \times (0, \bar{t})}, \\ T = T_0 \quad \text{in } \Omega, \text{ at } t = 0,\end{gathered} \tag{4}$$

where $\Gamma_2 \subset \partial\Omega$, a is the thermal diffusion coefficient [m^2/s], and S is the source term with the unit of [K/s].

2.2. The Lagrange-Galerkin Finite-Element Method. Some preliminaries are arranged for the derivation of a finite element scheme of (1) and (4). Let the subscript h denote the representative length of the triangulation, and let $\mathfrak{T}_h \equiv \{K\}$ denote a triangulation of Ω consisting of tetrahedral elements. Given that g is a vector valued function on Γ_1, the finite element spaces are as follows:

$$\begin{gathered}X_h \equiv \left\{v_h \in C^0(\overline{\Omega})^3;\, v_h|_K \in P_1(K)^3, \forall K \in \mathfrak{T}_h\right\}, \\ M_h \equiv \left\{q_h \in C^0(\overline{\Omega});\, q_h|_K \in P_1(K), \forall K \in \mathfrak{T}_h\right\}, \\ V_h(g) \equiv \left\{v_h \in X_h;\, v_h(P) = g(P), \forall P \in \Gamma_1\right\}, \\ \Theta_h(b) \equiv \left\{\theta_h \in M_h;\, \theta_h(P) = b(P), \forall P \in \Gamma_2\right\}, \\ V_h \equiv V_h(0),\ \Theta_h \equiv \Theta_h(0),\ Q_h = M_h.\end{gathered} \tag{5}$$

Let $(\cdot,\cdot)$ defines the L_2 inner product; the continuous bilinear forms a and b are introduced by

$$\begin{gathered}a(u, v) \equiv 2\nu (D(u), D(v)), \\ b(u, v) \equiv -(\nabla \cdot u, q),\end{gathered} \tag{6}$$

respectively.

Let Δt be the time increment, and let $N_{\bar{t}} \equiv [\bar{t}/\Delta t]$ be the total step number. Let the superscript n denote the time step; a finite element approximation of (1) is described as follows: find $\{(u_h^n, p_h^n)\}_{n=1}^{N_{\bar{t}}} \in V_h(g) \times Q_h$, such that for $(v_h, q_h) \in V_h \times Q_h$,

$$\begin{gathered}\left(\frac{u_h^n - u_h^{n-1} \circ X_1\left(u_h^{n-1}, \Delta t\right)}{\Delta t}, v_h\right) + a\left(u_h^n, v_h\right) \\ + b\left(v_h, p_h^n\right) = \left(f^n, v_h\right), \\ b\left(u_h^n, q_h\right) = 0,\end{gathered} \tag{7}$$

where $X_1(\cdot,\cdot)$ denotes a first-order approximation of a particle's position [5], and the notation $\circ$ denotes the composition of functions.

For the purpose of large scale computation, a piecewise equal-order interpolation for velocity and pressure is used, as can be seen from (5). Pressure stabilization is thus needed to keep the necessary link between V_h and Q_h. A penalty Galerkin least-squares (GLS) stabilization method for pressure is proved in [12] to hold the same asymptotic error estimates as the method of Hughes et al. [21] and it is computationally cheap. For P1/P1 elements, the stabilization is reduced to

$$\sum_{K\in\mathfrak{I}_h} \delta_K h_K^2 (\nabla p_h^n, -\nabla q_h)_K, \tag{8}$$

which does no modification to the momentum equation because of vanishing of the second-order derivate term. Here, h_K denotes the maximum diameter of an element K. Unlike [6, 12], where a constant δ (>0) is used as the stabilization parameter, an element-wise stabilization parameter

$$\delta_K = \begin{cases} \alpha, \\ \text{for } \log_{10}\left[\text{Max}\left\{\left\|\nabla p_h^{n-1}\right\|_2\right\}_{i=1}^4\right] \le 1, \\ \alpha \times \log_{10}\left[\text{Max}\left\{\left\|\nabla p_h^{n-1}\right\|_2\right\}_{i=1}^4\right], \\ \qquad\qquad\qquad \text{otherwise} \end{cases} \tag{9}$$

is used in this work, where ∇p_h^{n-1} is gradient of the FEM approximated pressure at t^{n-1} and i is the number of the nodal point in a tetrahedral element. Since α is very important to balance the accuracy and convergence of the scheme, it is discussed in Section 4.1. The localized stabilization parameter is designed to be adaptive to the pressure gradient, and thus it has a better control on the pressure field.

By adding (8) to (7), a pressure-stabilized FEM scheme for Navier-Stokes problems is achieved. The nonsteady iteration loops for solving (1) and (4) and then reads the following.

Step 1. Compute the particle's coordinates by

$$X_1\left(u_h^{n-1}, \Delta t\right) \equiv x - u_h^{n-1}\Delta t, \tag{10}$$

and search the element holding the particle at t^{n-1}.

Step 2. Find T_h^n by

$$\left(\frac{T_h^n - T_h^{n-1} \circ X_1\left(u_h^{n-1}, \Delta t\right)}{\Delta t}, \theta_h\right) + \left(a\nabla T_h^n, \nabla\theta_h\right) = \left(S^n, \theta_h\right). \tag{11}$$

Step 3. Find (u_h^n, p_h^n) by

$$\begin{aligned} &\left(\frac{u_h^n - u_h^{n-1} \circ X_1\left(u_h^{n-1}, \Delta t\right)}{\Delta t}, v_h\right) + a_0\left(u_h^n, v_h\right) \\ &\quad + b\left(v_h, p_h^n\right) + b\left(u_h^n, q_h\right) \\ &\quad + \sum_{K\in\mathfrak{I}_h} \delta_K h_K^2 (\nabla p_h^n, -\nabla q_h)_K \\ &= \left(f^n, v_h\right) + \left(\beta\left(T_r - T_h^n\right) g, v_h\right). \end{aligned} \tag{12}$$

Step 4. Compute the relative error by a $H^1 \times L^2 \times H^1$ norm defined by

$$\begin{aligned} \left\|(u, p, T)\right\|_{H^1\times L^2\times H^1} \equiv{}& \frac{1}{\sqrt{\text{Re}}}\|u\|_{H^1(\Omega)^3} \\ &+ \left\|p\right\|_{L^2} + \|T\|_{H^1(\Omega)}, \end{aligned} \tag{13}$$

where Re denotes the Reynolds number, and set

$$\begin{aligned} \text{diff} ={}& \frac{\left\|\left(u^n, p^n, T^n\right) - \left(u^{n-1}, p^{n-1}, T^{n-1}\right)\right\|_{H^1\times L^2\times H^1}}{\left\|\left(u^{n-1}, p^{n-1}, T^{n-1}\right)\right\|_{H^1\times L^2\times H^1}} \\ &\le \text{Err}_{\text{NS}} \end{aligned} \tag{14}$$

as the steady-state criterion; if (14) is satisfied or the number of loops reaches the maximum, then stop the iteration; otherwise, repeat Steps 1–3.

As can be seen from Steps 2 and 3, both (1) and (4) are approximated by the Lagrange-Galerkin method, and the searching algorithm only needs to be performed once in a nonsteady loop. It can also be seen that the solver is also flexible, and it can solve pure Navier-Stokes problems by setting the body force in (2) to external force and omitting Step 2.

3. Implementation

3.1. A Parallel Domain Decomposition System. To begin with the parallel domain decomposition method, the domain decomposition is introduced briefly as follows. The whole domain is decomposed into a number of *subdomains* without overlapping, and the solution of each *subdomain* is superimposed on the equation of the inner boundary of the *subdomains*. By static condensation, the linear system

$$K\overline{u} = \overline{f} \tag{15}$$

is written as

$$\begin{bmatrix} K_{II}^{(1)} & 0 & \cdots & 0 & K_{IB}^{(1)} R_B^{(1)} \\ 0 & \ddots & & \vdots & \vdots \\ \vdots & & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & K_{II}^{(N)} & K_{IB}^{(N)} R_B^{(N)} \\ R_B^{(1)T} K_{IB}^{(1)T} & \cdots & \cdots & R_B^{(N)T} K_{IB}^{(N)T} & K_{BB} \end{bmatrix} \begin{bmatrix} \overline{u}_I^{(1)} \\ \vdots \\ \vdots \\ \overline{u}_I^{(N)} \\ \overline{u}_B \end{bmatrix} = \begin{bmatrix} \overline{f}_I^{(1)} \\ \vdots \\ \vdots \\ \overline{f}_I^{(N)} \\ \overline{f}_B \end{bmatrix}, \tag{16}$$

where K is the stiffness matrix, $\overline{u}$ denotes the unknowns (u and p), and $\overline{f}$ is the force vector. R is the restriction operator consists of 0-1 matrix. The superscripts (N) means the

Nth *subdomain*, and subscript I and B relate to the element of the inner boundary, and interface boundary respectively.

From (16), it can be observed that the interface problems

$$\begin{aligned}\sum_{i=1}^{N} R_B^{(i)^T} \left(K_{BB}^{(i)} - K_{IB}^{(i)^T} K_{II}^{(i)^{-1}} K_{IB}^{(i)}\right) R_B^{(i)} \overline{u}_B \\ = \sum_{i=1}^{N} R_B^{(i)^T} \left(\overline{f}_B^{(i)} - K_{IB}^{(i)^T} K_{II}^{(i)^{-1}} \overline{f}_I^{(i)}\right)\end{aligned} \quad (17)$$

and the inner problems

$$\overline{u}_I^{(i)} = K_{II}^{(i)^{-1}} \left(\overline{f}_I^{(i)} - K_{IB}^{(i)^T} R_B^{(i)} \overline{u}_B\right), \quad i = 1, \ldots, N \quad (18)$$

can be solved separately [22]. In this work, the interface problems are solved first iteratively, and the inner problems are then solved by substituting $\overline{u}_B$ in to (18).

The Lagrange-Galerkin method keeps the symmetry of the stiffness matrix, and the GLS pressure-stabilization term in (8) also produces a symmetric matrix; therefore, K is symmetric in (15), and a PCG method is employed to get the u_I in (18), and to avoid drawback of the classical domain decomposition method, such as Neumann-Neumann and diagonal-scaling, a balanced domain decomposition preconditioner is used to prevent the growing of condition number with the number of *subdomains*. An identity matrix is chosen as the coarse matrix, and the coarse problem is solved incompletely by omitting the *fill-ins* in some sensitive places during the Cholesky factorization. By using this inexact balanced domain decomposition preconditioning, the coarse matrix is sparser and thus easier to be solved; therefore, the new solver is expected to have better solvability on large scale computation models.

3.2. The Lagrange-Galerkin Method in Parallel. The element searching algorithm requires a global-wise element information to determine the position of one particle in the previous time step. However, in the parallel domain decomposition system, the whole domain is split into several *parts* one processor element (PE) works only on the current *part,* and it does not contain any element information of other *parts.* Each *part* is further divided into many *subdomains,* and the domain decomposition is performed by the PE in charge of the *part*. This parallelity causes a computational difficulty: for each time step, the particle is not limited within one *part*; therefore, exchanging the data between different PEs is necessary, which demands the PEs to communicate in the *subdomain*-wise computation.

In order to know the position of a particle at t^{n-1}, a neighbour elements list is created at the beginning of the analysis. Based on the information of neighbour elements and the coordinates calculated by (10), it is possible to find the element holding this particle at t^{n-1}. A 2-dimensional searching algorithm is present as follows (λ_i is the barycentric coordinates, and $ne(\lambda_i)$ is the neighbour element; see Figure 1):

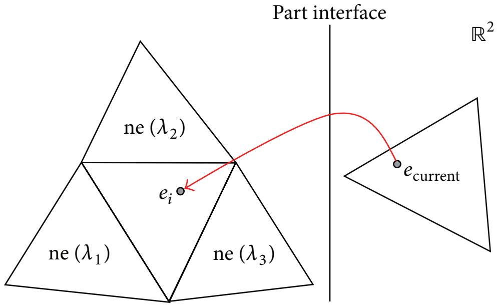


Figure 1: A searching algorithm.

```
(1) initialize: $e_0 = e_{\text{current}}$;
(2) iterate $i = 0, 1, \ldots,$ Maxloops;
    If $\lambda_1, \lambda_2, \lambda_3 > 0$, return $e_i$;
    else if $ne(\text{Min}\{\lambda_1, \lambda_2, \lambda_3\}) \neq$ boundary
        $e_{i+1} = ne(\text{Min}\{\lambda_1, \lambda_2, \lambda_3\})$;
    else break;
(3) return $e_i$.
```

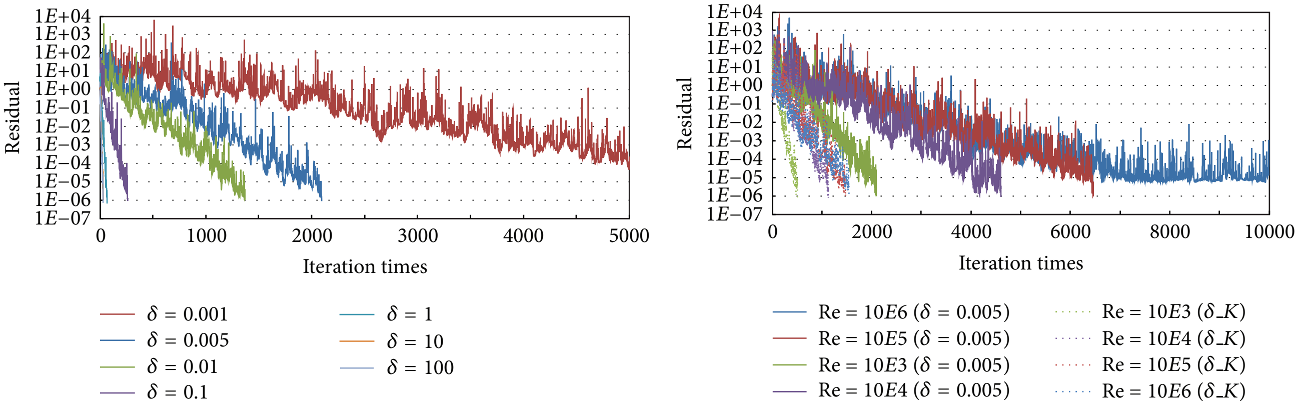
The request of the old solutions, which is the u_h^{n-1} in (10), is relatively trivial when using single PEs or simply solving the problem parallel using symmetric multiprocessing; however, in the domain decomposition system, the particle is not limited within one *part*; it may pass the interface of different *parts*, as can be seen from Figure 1. Because one PE only has the elements information that belongs to the current *part*, communications between PEs are necessary. However, the number of total elements in one *subdomain* may be different, which means that some point to point communication techniques, such as MPI_Send/MPI_Recv or MPI_Sendrecv in MPICH, cannot be used in element wise computation. In the previous research [23], a global variable to store all the old solutions is constructed. This method maintains a high computation speed but costs a huge memory usage. To reduce the memory consumption, a request-response system is used in this work. In the computation, the searching algorithm is performed first, and the element that contains the current particle in the previous time step is thus known; therefore, the PE to get u_h^{n-1} from is also known. However, as the sender does not know which PE requires message from itself, the receiver has to send its request to the sender first; after the request is detected, the sender sends the message to the receiver. The procedure is as follows:

(1) by scanning all the particles in the current *subdomain*, an array including all the data that is needed by the current PE is sent to all the other PEs.

(2) All PEs check if there is any request to itself. If it exists, PEs will prepare an array of the needed data and send it.

(3) The current PE receives the data sent by other PEs.

Data transferred by MPI communication should be packaged properly to avoid the overflow of MPI buffer in case



(a)

(b)

Figure 2: Convergence of (a) different constant δ at $\mathrm{Re} = 10^3$; (b) different Re for $\delta = 0.005$ and localized δ_K.

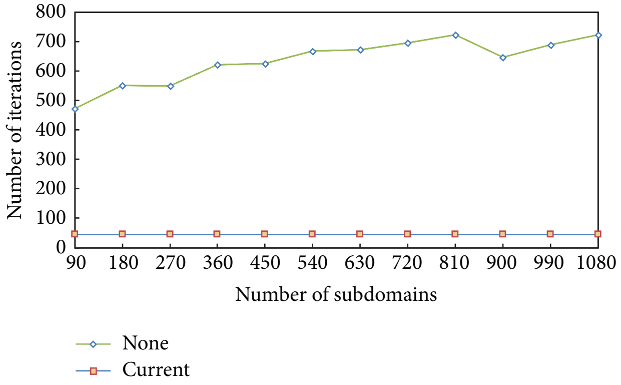


Figure 3: Numerical scalability.

of large-scale computation. Nonblocking communication is employed, and as the 3 steps are performed subsequently, thus the computation time and communication time will be overlapped.

4. Numerical Results and Discussion

The parallel efficiency of new solver is firstly evaluated in this section, and to validate the scheme, exact solutions and available benchmark results classical computational models are compared. The CG convergence is judged by Euclidian norm with a tolerance of 10^{-6}, and for nonsteady iteration, $\mathrm{Err}_{\mathrm{NS}} = 10^{-4}$ is set as the criterion, using the $H^1 \times L^2 \times H^1$ norm defined in (13). For pure Navier-Stokes problems, a similar $H^1 \times L^2$ norm, which is related to velocity and pressure, is employed to judge the steady state.

4.1. Efficiency Test. The BDD serious preconditioners were employed in this work; they are very efficient, and their iteration numbers are about 1âĄĎ10 of the normal domain decomposition preconditioners (cf. [23]). The inexact preconditioner mentioned in Section 3 also shows good convergence and is more suitable for large scale computations [24]. It was set as the default preconditioner for all the following computations of this research.

The penalty methods are not consistent since the substitution of an exact solution into the discrete equations (12) leaves a residual that is proportional to the penalty parameter (cf. [17]); therefore, δ_K should be determined carefully. Numerical experiments of a lid-driven cavity flow were tested, and the mesh size was 62 × 62 × 62. The total degrees of freedom (DOF) are 1,000,188, and the results are given by Figure 2. For the purpose of higher accuracy, δ_K is expected to be small; however, the convergence turns worse when δ_K goes small, as can be seen from Figure 2(a). In Figure 2(b), a constant $\delta = 0.005$ is used for different Reynolds numbers, and no convergence is achieved within 10000 PCG loops for $\mathrm{Re} = 10^6$; and the comparison shows that the δ_K performs much better than a constant $\delta = 0.005$ when α is set to 0.005.

The parallel efficiency is assessed firstly by freezing the mesh size of test problem and refining the domain decomposition by decreasing the subdomain size and therefore increasing the number of subdomains; the comparison of the numerical scalability of the current scheme with and without the preconditioner is assessed by Figure 3. It can be seen that with the preconditioning technique, the iterative procedure

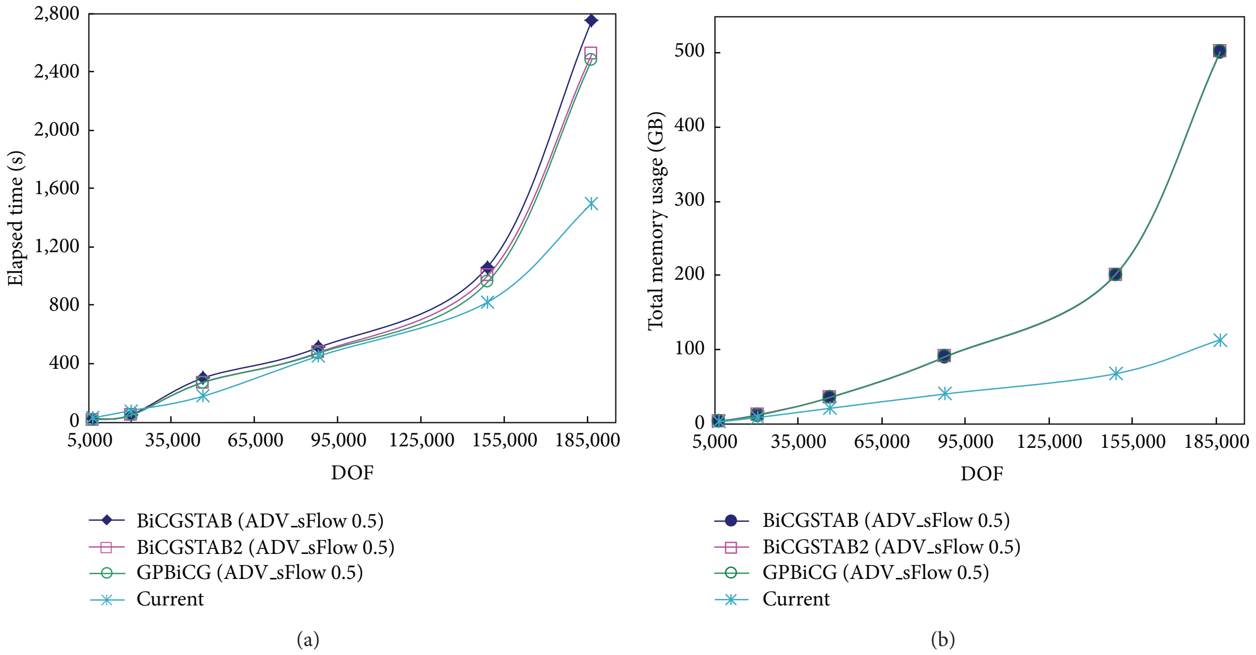


Figure 4: Time and memory usages.

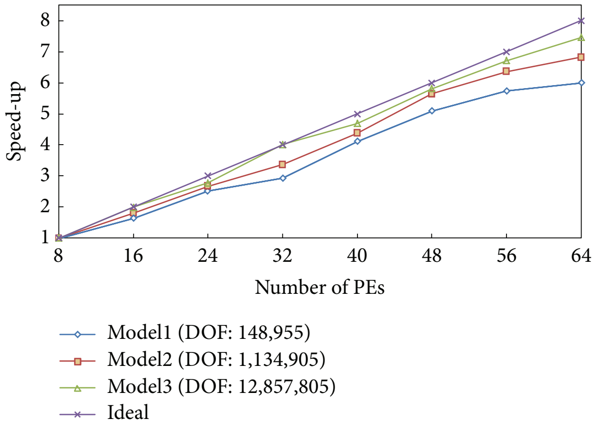


Figure 5: Numerical scalability.

of current scheme converges rapidly, and the convergence is independent of the number of subdomains.

Based on the paralyzed Lagrange-Galerkin method, the new solver makes a symmetric stiffness matrix, therefore only the lower/upper triangular matrix needs to be saved. Moreover, nonblocking MPI communication is used instead of constructing global arrays to keep the old solutions, and the current solver is expected to reduce the memory consumption without sacrificing the computation speed. The usage of time and memory of solving the thermal driven cavity problem by different solvers is compared, and the results are given by Figure 4.

The test problem was solved by the new solver and the ADV_sFlow 0.5 [25], which contains some nonsymmetric product-type solvers like GPBiCG, BiCGSTAB, and BiCGSTAB2 [20]. The ADV_sFlow 0.5 employed a domain decomposition system similar to the work; however, no precondition technique is used because of the non-symmetry of the stiffness matrix in (15). The comparisons of elapsed time and memory occupation of the new solver and that of product-type solvers in ADV_sFlow 0.5 are show in Figures 4(a) and 4(b). As can be seen, the current scheme reduces the demand of computation time and memory consumption remarkably, and it is more suitable for large scale problems than product-type solvers.

The parallel scalability of the searching algorithm is also a concern for us, as it characterizes the ability of an algorithm to deliver larger speed-up using a larger number of PEs. To know this, the number of *subdomains* in one part is fixed, and computations on the test problem of various mesh sizes are performed by the new scheme. The speed-up is shown in Figure 5. Three models were tested by the searching algorithm. With an increase in the mesh size of the computation model, the parallel scalability of the searching algorithm tends to be better. An explanation to this is that when the DOF increase, the number of elements in one subdomain is also increasing; therefore, the searching algorithm is accelerated more efficiently. However, too many elements in one subdomain will occupy more memory, and a trade-off strategy is necessary for parameterization.

4.2. Validation Tests. In this section, a variety of test problems have been presented in order to prove the capability of the parallel Lagrange-Galerkin algorithm. Benchmarks test of

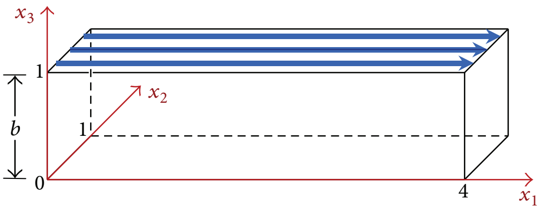


Figure 6: A plane Couette flow model.

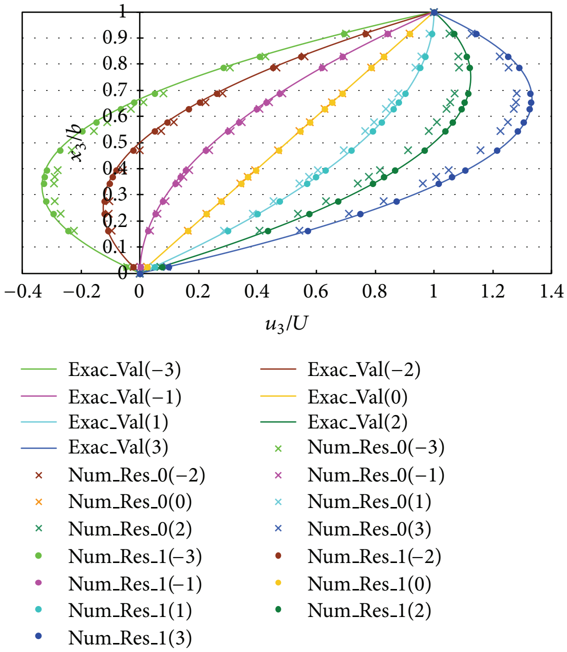


Figure 7: Numerical results versus exact solutions.

Navier-Stokes problems are in Sections 4.2.1, 4.2.2, and 4.2.3, and flow-thermal coupled problems are in Sections 4.2.4 and 4.2.5.

4.2.1. A Plane Couette Flow. The solver for Navier-Stokes equations in (1) was first tested with a 3D plane Couette flow. Under ideal conditions, the model is of infinite length; therefore, 4 times of the height is used as the length of the model see Figure 6. A constant velocity $(\hat{u}, 0, 0)$ is applied on the upper horizontal face, and no-slip conditions are set on the lower horizontal face. A pressure gradient is imposed along x_1 for all the faces as essential boundary conditions.

An unstructured 3D mesh was generated by ADVENTURE_TetMesh [25], and the local density around the plane of $x_1 = 2$, where the data was picked from, was enriched. The total DOF is around 1,024,000. The so-called Brinkman number [26] is believed to be the dominating parameter of the flow, and a serious of numerical experiments is done at various Brinkman number. To simulate the infinity length better, the exact solution is enforced on both the left face ($x_1 = 0$) and the right face ($x_1 = 4$) as Dirichlet boundary conditions. The comparisons between the computation results and exact solutions are given by Figure 7. Dotted lines are used to present the results, and they are named as "Num_Res_1(B)," where B stands for the Brinkman number. Crossed lines in Figure 7 present the computation results with no exact solutions setting on the left and right faces, and they are named as "Num_Res_0(B)."

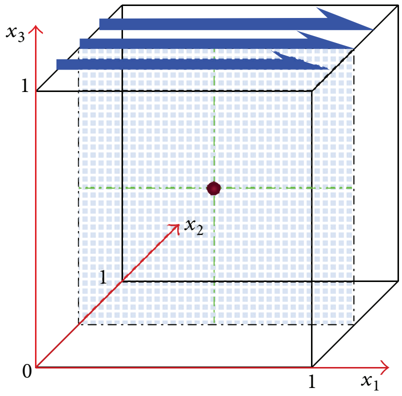


Figure 8: A lid-driven cavity model.

It can be seen form Figure 7 that both of these two sets of computation results show good agreement with the exact solution, and dotted lines are closer to the exact solution, representing a better simulation to the ideal condition (cf. [27, 28]).

4.2.2. A Lid-Driven Cavity. The Navier-Stokes problems solver was then verified by a lid-driven cavity flow. The ideal gas flows over the upper face of the cube, and no-slip conditions are applied to all other faces, as in Figure 8.

All the faces of the cube were set with Dirichlet boundary conditions, and a zero reference pressure was at the centre of the cube to keep the simulation stable. The pressure profiles of the scheme using localized stabilization parameter in (9) and the scheme using constant ($\delta = 1$) parameter are compared, and the results are show in Figure 9.

Figure 9(a) shows the pressure counters of the scheme with the localized stabilization parameter in (9) and the Figure 9(b), shows scheme with a constant parameter. The model was run at Re $= 10^4$, and oscillations are viewed in Figure 9(b); however, the isolines in Figure 9(a) is quite smooth, showing that the pressure-stabilization term has a better control on the pressure field at high Reynolds number.

The model was run at different Reynolds numbers with a $128 \times 128 \times 128$ mesh to test the solvability of the new scheme. As shown in Figure 10, when the Reynolds number increases, the eddy at right bottom of plane x_1x_3 vanishes, while the eddy at the left bottom appears due to the increasing in the speed, and the flow goes more likely around the wall. The primary eddy goes lower and lower when Reynolds number becomes higher, and the particle is no longer limited to a single side of the cavity; it can pass from one side to the other, and back again violating the mirror symmetry, as is seen from other planes of Figure 10. Similar 3D results for high Reynolds number were reported by [29], and the solvability of the new solver for high Reynolds number was confirmed.

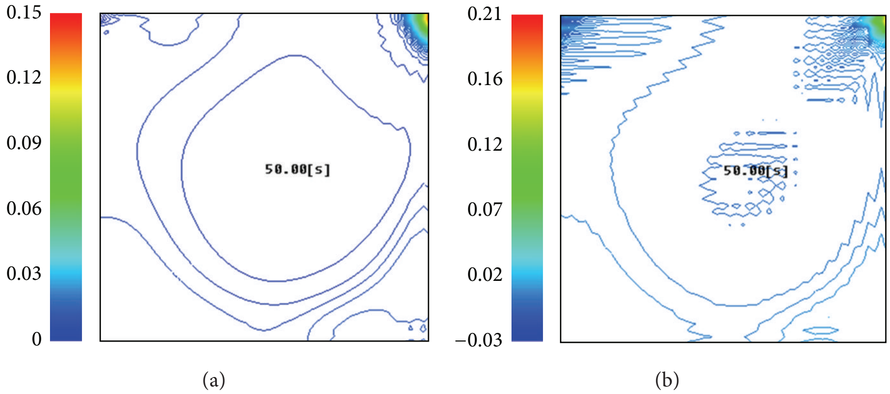


Figure 9: Pressure counters (Re = 10^4).

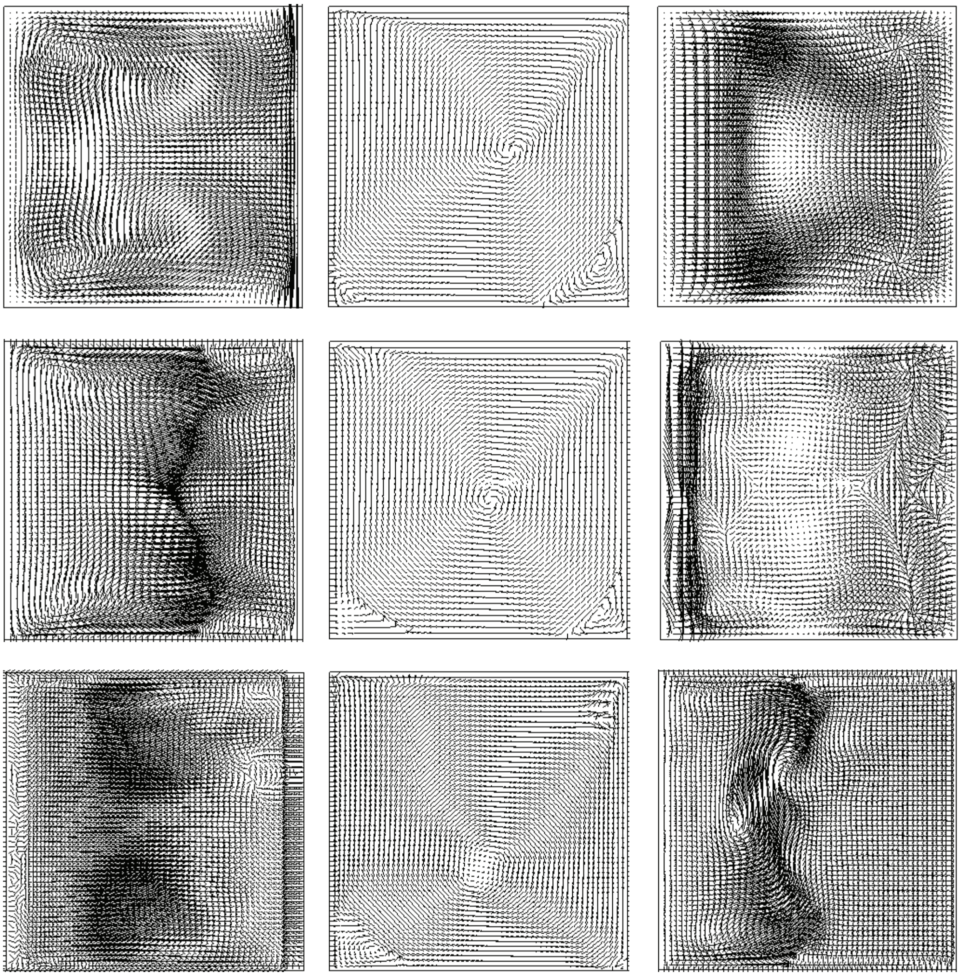


Figure 10: Velocity and pressure profiles for different Reynolds number: Re = 1,000 (top), Re = 3,200 (middle), and Re = 12,000 (bottom) along different middle planes: plane x_1x_2 (left), plane x_1x_3 (middle), and plane x_2x_3 (right).

4.2.3. Backward Facing Step. The solver for Navier-Stokes equations was then tested with backward facing step, the fluid considered was air. The problem definition is shown in Figure 11, and the height of the step h is the characteristic length. An unstructured 3D mesh was generated with 419,415 nodal points and 2,417,575 tetrahedral elements, and the local density of mesh was increased around the step.

A laminar flow is considered to enter the domain at inlet section, the inlet velocity profile is parabolic, and the Reynolds number is based on the average velocity at the inlet.

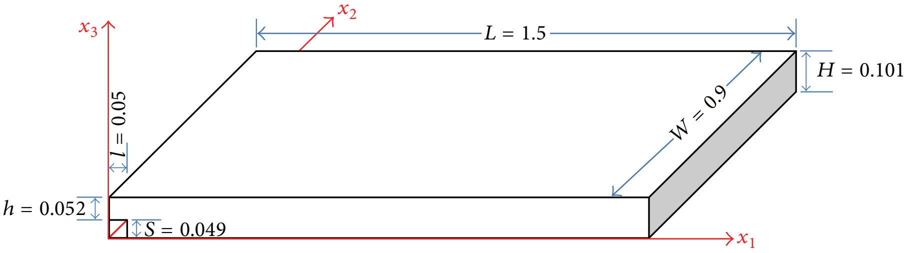


FIGURE 11: Backward facing steps.

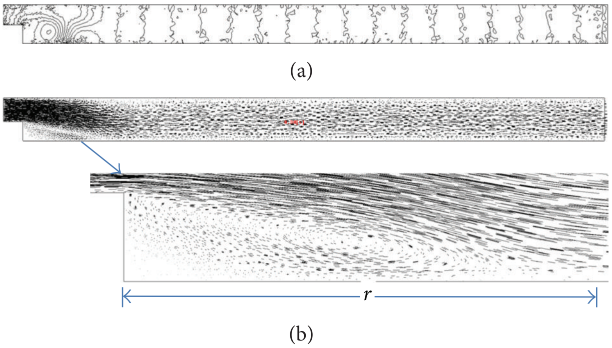


FIGURE 12: Pressure counters (a) and velocity vectors (b) (Re = 200).

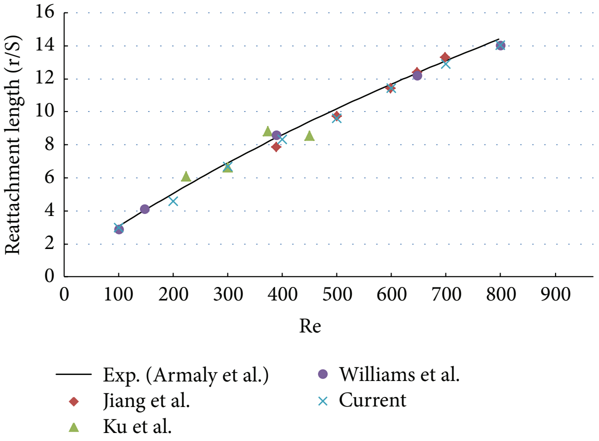


FIGURE 13: Primary reattachment lengths.

The total length of the domain is 30 times the step height, so that the zero pressure is set at the outlet. A full 3D simulation of the step geometry for 100 ≤ Re ≤ 800 is present in this paper, and the primary reattachment lengths are predicted.

To determine the reattachment length, the position of the zero-mean-velocity line was measured. The points of detachment and reattachment were taken as the extrapolated zero-velocity line down the wall. The pressure contour in Figure 12(a) confirms the success of the pressure-stabilization method; velocity vectors and the primary attachment are demonstrated in Figure 12(b); similar results have been documented by many, like in [10, 30].

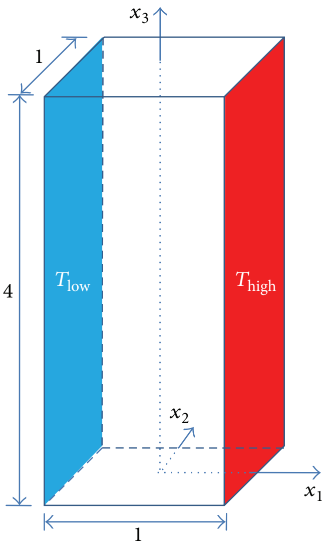


FIGURE 14: The model of infinite plates.

The comparison of primary reattachment length between current results and other available benchmark results are show in Figure 13. It is seen that the agreement is excellent at different Rayleigh numbers (cf. [31, 32]).

4.2.4. Natural Convection of Flat Plates. In order to test the coupled solver of Navier-Stokes equations and the convection-diffusion equation, the third application model was the natural convection between two infinite flat plates. The geometry is given in 3-dimensional by Figure 14. No-slip boundary conditions applied on the left and right vertical walls. The temperature on the left wall is assumed to be lower and set at $5[K]$; the right wall is set at $6[K]$. An unstructured 3D mesh about 1 million tetrahedral elements was generated, and the local grid density around the mid-plane was enriched.

The model was run at the size of 20 × 20 × 80 to get the numerical solutions, and it was compared with the exact solutions in Figure 15. To simulate the infinity length of the plate better, the exact solution is enforced on both the upper face ($x_3 = 4$) and the lower face ($x_3 = 0$) as what is done in Section 4.2.1, and the results are present by a dotted line ("Num_Res_1") in Figure 12. And the model without exact solution set as boundary is named as "Num_Res_0" in Figure 15.

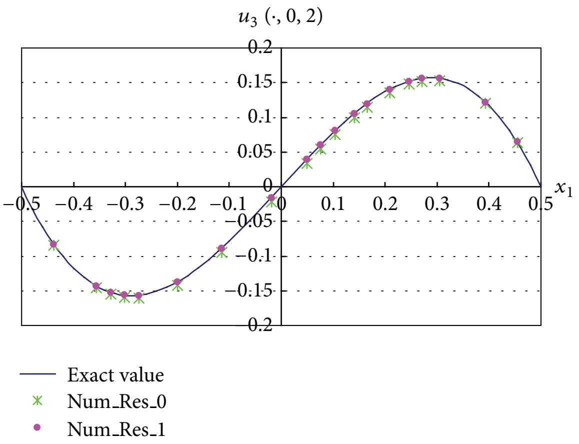


Figure 15: Numerical results versus exact solutions.

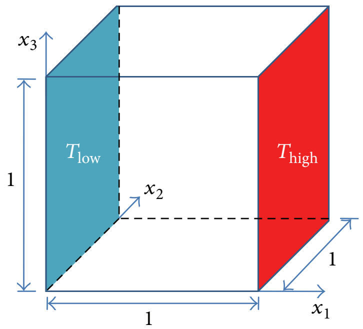


Figure 16: A thermal-driven cavity.

With the parameter setting of $\nu = 0.5$, $T_r = 5.5$, $\beta = 1.0$, and $a = 1$, the numerical experiment was performed. Results of the profile on $u_3(\cdot, 0, 2)$, which are believed by many to be very sensitive, are shown in Figure 15. Both "Num_Res_0" and "Num_Res_1" are in great agreement with the exact solutions, and "Num_Res_1" is closer to the exact solution, producing a better simulation to the ideal condition. Similar results have been documented in [33].

4.2.5. Thermal-Driven Cavity. The new solver is also applied to a 3-dimensional nonlinear thermal driven cavity flow problem, which is cavity full of ideal gas; see Figure 16.

No-slip boundary conditions are assumed to prevail on all the walls of the cavity. Both the horizontal walls are assumed to be thermally insulated, and the left and right sides are kept at different temperatures. The cube is divided into $120 \times 120 \times 120$ small cubes, and each small cube contains six tetrahedral elements. The time step is set to 0.01 s, with $\mathrm{Pr} = 0.71$ and $\mathrm{Ra} = 10^4$; the steady state is achieved after 0.39 s, as in Figure 17.

Figures 17(a) and 17(b) show the contour of vorticity and the velocity vectors at the steady stage, respectively, from the front view. The temperature contour is shown in Figure 17(c), and pressure profiles are show in Figure 17(d). The previous results convince us of the success in solving flow-thermal coupled problems described by (1) and (4). Similar three-dimensional results can also be found in [33, 34]. The pressure profile in Figure 17(d) is smooth and symmetric, implying that the stabilization item in (8) works well.

In order to further validate the new solver, a comparison of temperature and velocity profiles of the current solver and other benchmark results was made. The centreline velocity results $w(\cdot, 0.5, 0.5)$ and the temperature results $T(\cdot, 0.5, 0.5)$, which are believed to be very sensitive in this simulation, are present in diagrams (a) and (b) of Figure 18, respectively. The velocity results share close resemblance to that of the ADV_sFlow 0.5, and they both show the more end-wall effects compared with the results of 2D case. The three temperature results show good agreement with each other, and the line representing the current results is the smoothest, as the mesh is the finest among the three. Similar results have been reported by other researchers (cf. [10, 33, 34]).

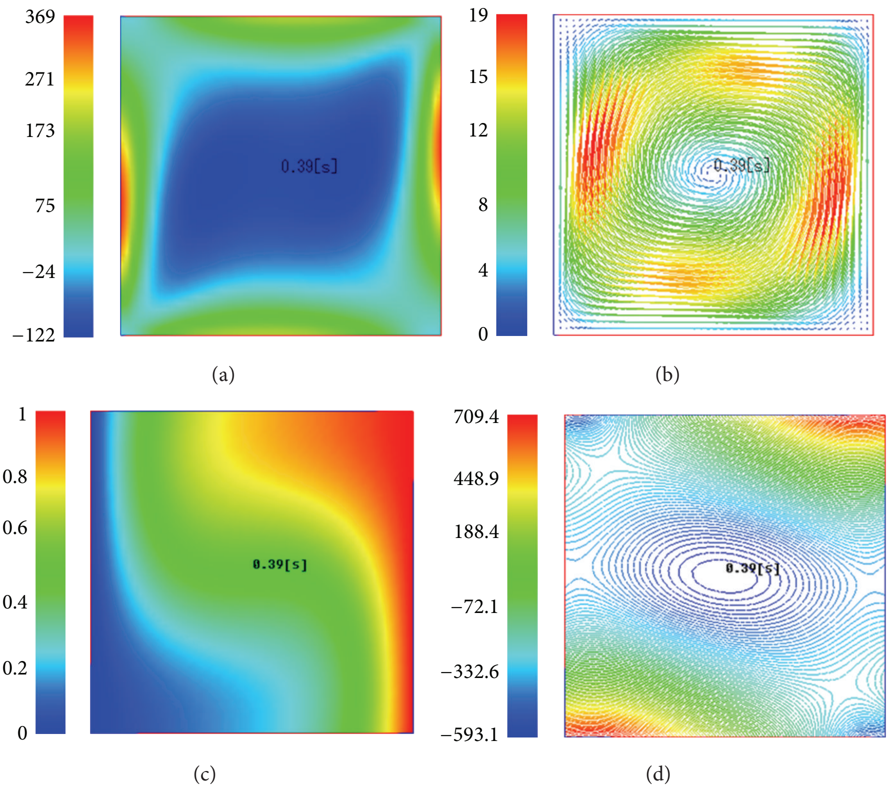
Thermal convection problems are believed to be dominated by two dimensionless numbers by many researchers, the Prandtl number and the Rayleigh number. To acquaint ourselves with the solvability of the new solver and to challenge applications of higher difficulty, a wide range of Rayleigh numbers from 10^3 to 10^7 is studied with $\mathrm{Pr} = 0.71$, and the results for the steady-state solution are presented in Figure 19. Dimensionless length is used and the variation of Rayleigh number is determined by changing the characteristic length of the model.

The local Nusselt number ($\mathrm{Nu} = \partial T/\partial x_1$) is a concern of many researchers, as they are sensitive to the mesh size. In Figure 19, the diagram (a) and the diagram (b) represent the local Nusselt number at the hot wall and the cold wall, respectively. Similar results can also be found in [10, 30, 35, 36]. The capability of the solver based on domain decomposed Lagrange-Galerkin scheme for high Rayleigh number is also confirmed by this figure.

The new solver enables the simulation of large scale problems, thus models of Rayleigh number up to 10^7 can be run on small PC cluster. In this simulation, an unstructured mesh of 30,099,775 DOF is generated, the time step, is 0.01 s and it takes about 24 hours to finish, using the a small Linux cluster of 64 PEs (64 cores@2.66 GHz).

5. Conclusions

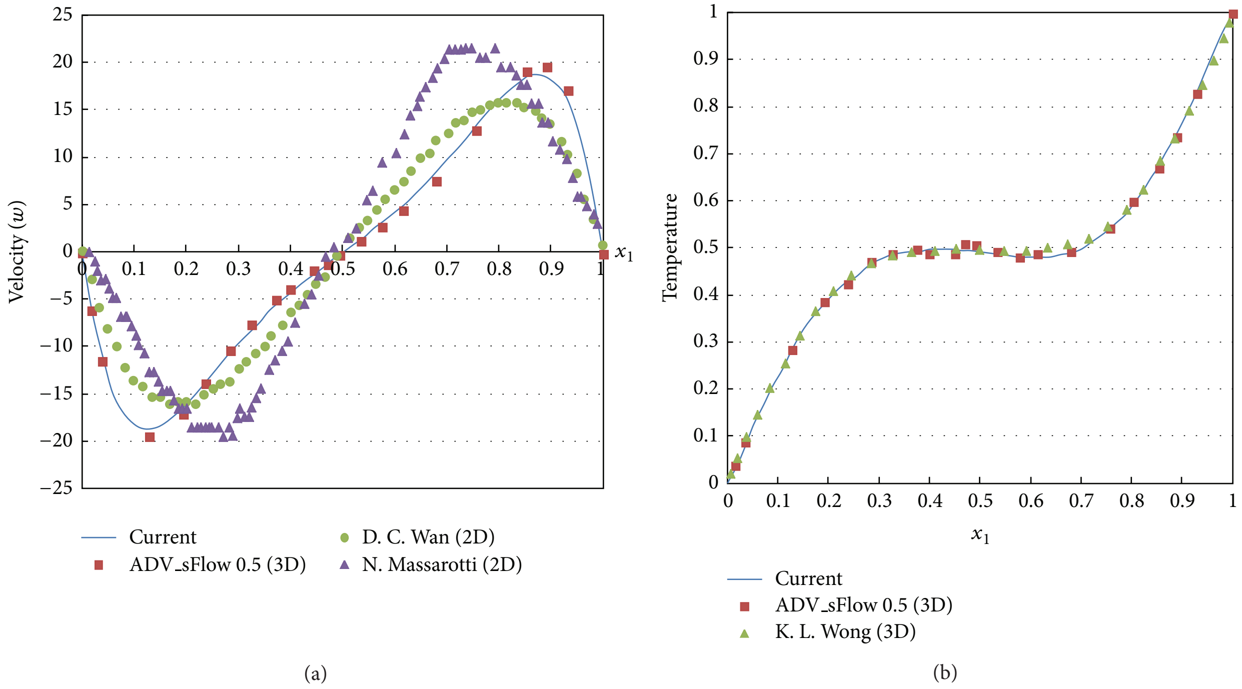
A pressure-stabilized Lagrange-Galerkin method is implemented in a domain decomposition system in this research. By using localized stabilization parameter, the new scheme shows better control in the pressure field than constant stabilization parameter; therefore it has good solvability at high Reynolds number and high Rayleigh number. The reliability and accuracy of the present numerical results are validated by comparing with the exact solutions and recognized numerical results. Based on a domain decomposition method, the element searching algorithm shows good numerical scalability and parallel efficiency. The new solver reduces the memory consumption and is faster than classical product-type solvers. It is able to solve large scale problems of over 30 million degrees of freedom within one day by a small PC cluster.



(a) (b)

(c) (d)

FIGURE 17: Steady state of the thermal-driven cavity (Ra = 10^4).



(a) (b)

FIGURE 18: Centerline temperature velocity profiles of the symmetry plane (Ra = 10^4).

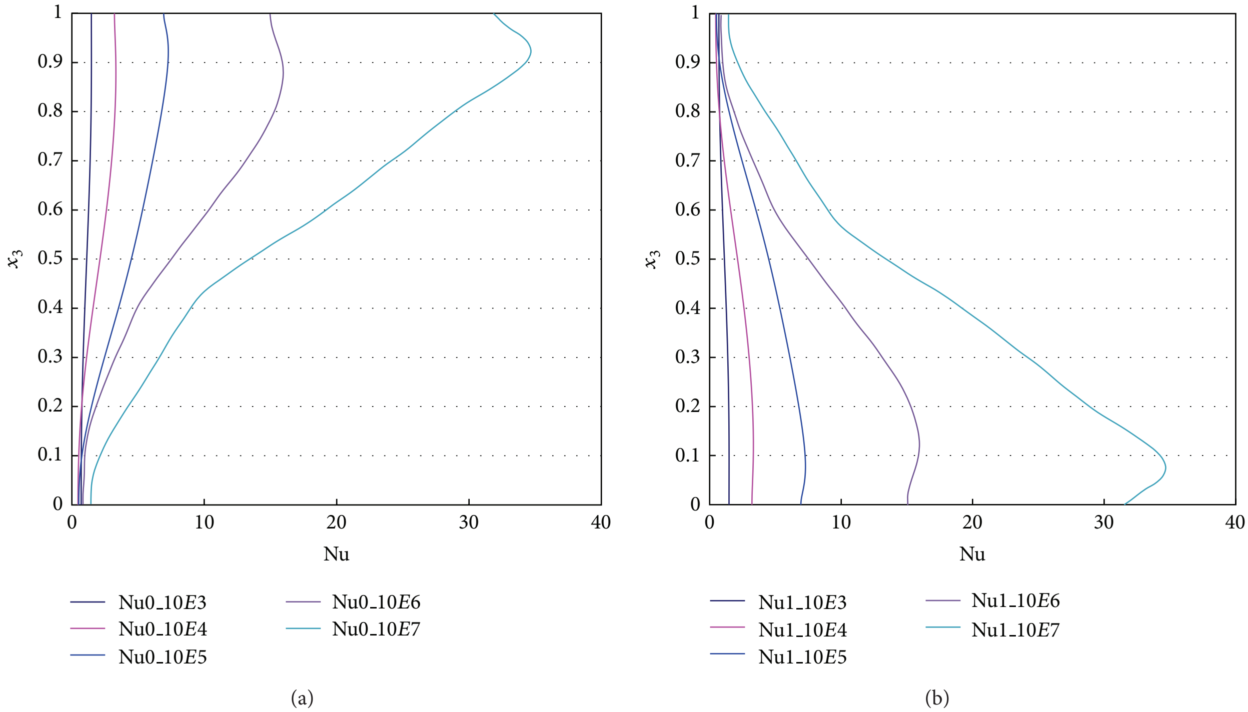


FIGURE 19: Local Nusselt number along the hot wall (a) and the cold wall (b).

Acknowledgments

This work was supported by the National Science Foundation of China (NSFC), Grants 11202248, 91230114, and 11072272; the China Postdoctoral Science Foundation, Grant 2012M521646, and the Guangdong National Science Foundation, Grant S2012040007687.

References

[1] M. Bercovier, O. Pironneau, and V. Sastri, "Finite elements and characteristics for some parabolic-hyperbolic problems," *Applied Mathematical Modelling*, vol. 7, no. 2, pp. 89–96, 1983.

[2] J. Douglas Jr. and T. F. Russell, "Numerical methods for convection-dominated diffusion problems based on combining the method of characteristics with finite element or finite difference procedures," *SIAM Journal on Numerical Analysis*, vol. 19, no. 5, pp. 871–885, 1982.

[3] O. Pironneau, "On the transport-diffusion algorithm and its applications to the Navier-Stokes equations," *Numerische Mathematik*, vol. 38, no. 3, pp. 309–332, 1982.

[4] T. F. Russell, "Time stepping along characteristics with incomplete iteration for a Galerkin approximation of miscible displacement in porous media," *SIAM Journal on Numerical Analysis*, vol. 22, no. 5, pp. 970–1013, 1985.

[5] O. Pironneau, *Finite Element Methods for Fluids*, John Wiley & Sons, Chichester, UK, 1989.

[6] H. Notsu, "Numerical computations of cavity flow problems by a pressure stabilized characteristic-curve finite element scheme," in *Japan Society for Computational Engineering and Science*, 2008.

[7] H. Notsu and M. Tabata, "A combined finite element scheme with a pressure stabilization and a characteristic-curve method for the Navier-Stokes equations," *Transactions of the Japan Society for Industrial and Applied Mathematics*, vol. 18, no. 3, pp. 427–445, 2008.

[8] M. Tabata and S. Fujima, "Robustness of a characteristic finite element scheme of second order in time increment," in *Computational Fluid Dynamics 2004*, pp. 177–182, 2006.

[9] H. Rui and M. Tabata, "A second order characteristic finite element scheme for convection-diffusion problems," *Numerische Mathematik*, vol. 92, no. 1, pp. 161–177, 2002.

[10] N. Massarotti, P. Nithiarasu, and O. C. Zienkiewicz, "Characteristic-based split (CBS) algorithm for incompressible flow problems with heat transfer," *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 11, no. 3, p. 278, 2001.

[11] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, vol. 15 of *Springer Series in Computational Mathematics*, Springer, New York, NY, USA, 1991.

[12] F. Brezzi and J. Douglas Jr., "Stabilized mixed methods for the Stokes problem," *Numerische Mathematik*, vol. 53, no. 1-2, pp. 225–235, 1988.

[13] H. Jia, D. Liu, and K. Li, "A characteristic stabilized finite element method for the non-stationary Navier-Stokes equations," *Computing*, vol. 93, no. 1, pp. 65–87, 2011.

[14] H. Kanayama, D. Tagami, T. Araki, and H. Kume, "A stabilization technique for steady flow problems," *International Journal of Computational Fluid Dynamics*, vol. 18, no. 4, pp. 297–301, 2004.

[15] C. R. Dohrmann and P. B. Bochev, "A stabilized finite element method for the Stokes problem based on polynomial pressure projections," *International Journal for Numerical Methods in Fluids*, vol. 46, no. 2, pp. 183–201, 2004.

[16] H. M. Park and M. C. Sung, "Stabilization of two-dimensional Rayleigh-Bénard convection by means of optimal feedback control," *Physica D*, vol. 186, no. 3-4, pp. 185–204, 2003.

[17] T. Barth, P. Bochev, M. Gunzburger, and J. Shadid, “A taxonomy of consistently stabilized finite element methods for the Stokes problem,” *SIAM Journal on Scientific Computing*, vol. 25, no. 5, pp. 1585–1607, 2004.

[18] P. B. Bochev, M. D. Gunzburger, and R. B. Lehoucq, “On stabilized finite element methods for the Stokes problem in the small time step limit,” *International Journal for Numerical Methods in Fluids*, vol. 53, no. 4, pp. 573–597, 2007.

[19] G. C. Buscaglia and E. A. Dari, “Implementation of the Lagrange-Galerkin method for the incompressible Navier-Stokes equations,” *International Journal for Numerical Methods in Fluids*, vol. 15, no. 1, pp. 23–26, 1992.

[20] H. A. van der Vorst, *Iterative Krylov Methods for Large Linear Systems*, vol. 13 of *Cambridge Monographs on Applied and Computational Mathematics*, Cambridge University Press, Cambridge, UK, 2003.

[21] T. J. R. Hughes, L. P. Franca, and M. Balestra, “A new finite element formulation for computational fluid dynamics. V. Circumventing the Babuška-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations,” *Computer Methods in Applied Mechanics and Engineering*, vol. 59, no. 1, pp. 85–99, 1986.

[22] A. Toselli and O. Widlund, *Domain Decomposition Methods—Algorithms and Theory*, vol. 34 of *Springer Series in Computational Mathematics*, Springer, Berlin, Germany, 2005.

[23] Q. Yao, H. Kanayama, H. Notsu, and M. Ogino, “Balancing domain decomposition for non-stationary incompressible flow problems using a characteristic-curve method,” *Journal of Computational Science and Technology*, vol. 4, pp. 121–135, 2010.

[24] Q. H. Yao and Q. Y. Zhu, “Investigation of the contamination control in a cleaning room with a moving Agv by 3D large-scale simulation,” *Journal of Applied Mathematics*, vol. 2013, Article ID 570237, 10 pages, 2013.

[25] “Adventure Project,” http://adventure.sys.t.u-tokyo.ac.jp/.

[26] C. W. Hall, *Laws and Models: Science, Engineering, and Technology*, CRC Press, 2000.

[27] R. S. Brodkey and H. C. Hershey, *Basic Concepts in Transport Phenomena*, Brodkey, 2001.

[28] S. M. Richardson, *Fluid Mechanics*, Hemisphere, 1989.

[29] E. Hachem, B. Rivaux, T. Kloczko, H. Digonnet, and T. Coupez, “Stabilized finite element method for incompressible flows with high Reynolds number,” *Journal of Computational Physics*, vol. 229, no. 23, pp. 8643–8665, 2010.

[30] M. T. Manzari, “An explicit finite element algorithm for convection heat transfer problems,” *International Journal of Numerical Methods for Heat and Fluid Flow*, vol. 9, no. 8, pp. 860–877, 1999.

[31] N. A. Malamataris, “A numerical investigation of wall effects in three-dimensional, laminar flow over a backward facing step with a constant aspect and expansion ratio,” *International Journal for Numerical Methods in Fluids*, vol. 71, pp. 1073–1102, 2013.

[32] P. T. Williams and A. J. Baker, “Numerical simulations of laminar flow over a 3D backward-facing step,” *International Journal for Numerical Methods in Fluids*, vol. 24, no. 11, pp. 1159–1183, 1997.

[33] H. Kanayama, K. Komori, and D. Sato, “Development of a Thermal Convection Solver with Hierarchical Domain Decomposition Method,” in *Proceedings of the 8th World Congress on Computational Mechanics and the 5th European Congress on Computational Methods in Applied Sciences and Engineering*, Venice, Italy, 2008.

[34] K. L. Wong and A. J. Baker, “A 3D incompressible Navier-Stokes velocity-vorticity weak form finite element algorithm,” *International Journal for Numerical Methods in Fluids*, vol. 38, no. 2, pp. 99–123, 2002.

[35] G. de Vahl Davis, “Natural convection of air in a square cavity: a bench mark numerical solution,” *International Journal for Numerical Methods in Fluids*, vol. 3, no. 3, pp. 249–264, 1983.

[36] G. de Vahl Davis and I. P. Jones, “Natural convection in square cavity: a comparison exercise,” *International Journal for Numerical Methods in Fluids*, vol. 3, no. 3, pp. 227–248, 1983.