INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2010; **62**:119–137
Published online 26 February 2009 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.2014

A variational multiscale Newton–Schur approach for the incompressible Navier–Stokes equations

D. Z. Turner*, †, K. B. Nakshatrala and K. D. Hjelmstad

Department of Civil and Environmental Engineering, University of Illinois, 2103 Newmark Lab MC-250, 205 N Mathews Ave, Urbana, IL 61801, U.S.A.

SUMMARY

In the following paper, we present a consistent Newton–Schur (NS) solution approach for variational multiscale formulations of the time-dependent Navier–Stokes equations in three dimensions. The main contributions of this work are a systematic study of the variational multiscale method for three-dimensional problems and an implementation of a consistent formulation suitable for large problems with high nonlinearity, unstructured meshes, and non-symmetric matrices. In addition to the quadratic convergence characteristics of a Newton–Raphson-based scheme, the NS approach increases computational efficiency and parallel scalability by implementing the tangent stiffness matrix in Schur complement form. As a result, more computations are performed at the element level. Using a variational multiscale framework, we construct a two-level approach to stabilizing the incompressible Navier–Stokes equations based on a coarse and fine-scale subproblem. We then derive the Schur complement form of the consistent tangent matrix. We demonstrate the performance of the method for a number of three-dimensional problems for Reynolds number up to 1000 including steady and time-dependent flows. Copyright © 2009 John Wiley & Sons, Ltd.

Received 8 October 2008; Revised 6 January 2009; Accepted 11 January 2009

KEY WORDS: incompressible Navier–Stokes; stabilized finite elements; variational multiscale formulation; consistent Newton–Raphson; Schur complement; three-dimensional flows

1. INTRODUCTION

The incompressible Navier–Stokes equations are used to model a number of important physical phenomena, including pipe flow, flow around airfoils, ocean currents, weather, and blood flow in an artery, among others. Significant emphasis has been placed in the literature on developing

*Correspondence to: D. Z. Turner, Department of Civil and Environmental Engineering, University of Illinois, 2103 Newmark Lab MC-250, 205 N Mathews Ave, Urbana, IL 61801, U.S.A.
†E-mail: dzturne1@uiuc.edu

Contract/grant sponsor: The Department of Energy; contract/grant number: DOE DE-FC02-07ER64323

stabilized formulations robust enough to model complex flows at high Reynolds number [1–3]. The variational multiscale method, due to Hughes [4], has been gaining popularity as a robust stabilization technique. This method consists of decomposing the problem into a coarse (or resolvable) and fine-scale (unresolvable) subproblem. The inclusion of the fine-scale influence stabilizes the solution.

The variational multiscale method has been well studied for two-dimensional problems, but less so for three-dimensional problems. Also, three-dimensional effects become more influential as the Reynolds number increases. It is well known that two-dimensional analysis cannot accurately capture flow features for high Reynolds flows. For example, for the backward facing step problem at Reynolds number greater than 450, the reattachment length obtained by two-dimensional analysis does not match experimental results [5]. One of the main contributions of this paper is a systematic study of the variational multiscale method for three-dimensional problems. Solving three-dimensional problems requires many more degrees of freedom (DOF). Using a mixed formulation, in which velocity and pressure are treated as unknowns, the number of DOF easily exceeds several million. Another contribution of this work is the Schur complement implementation that greatly reduces the problem size and improves parallel performance.

The main challenge to numerically modeling the incompressible Navier–Stokes equations using the finite element method is due to the well-known instability associated with using equal order interpolation for the velocity and pressure (which is computationally most convenient). Many stabilized methods for the incompressible Navier–Stokes equations have been developed that allow for equal order interpolation including [4, 6–10]. Stabilized formulations that employ a variational multiscale framework have been gaining popularity. First introduced as a pathway to developing stabilized formulations from first principles, the variational multiscale concept has been used in a number of contexts, particularly for the incompressible Navier–Stokes equations [7, 11–17].

One can either solve the fine-scale problem in terms of the coarse-scale variables, thereby generating a stabilization term, and substitute back into the coarse-scale problem, or solve both problems simultaneously. Typically a fixed-point iteration technique is used to linearize the weak form of the governing equations. We have shown in [12] that by using a consistent Newton–Raphson (NR)-based linearization, quadratic convergence may be achieved and that convergence may be achieved for problems for which fixed-point iterative methods diverge or converge very slowly. In this work, we present a Schur complement implementation of the consistent NR-based linearization of incompressible Navier–Stokes equations that provides good computational efficiency and scalability. We extend the work presented in [12] to large, nonlinear, three-dimensional problems on unstructured meshes.

Using the Schur complement implementation for consistent formulations has been used primarily in the field of domain decomposition techniques and computational plasticity [18, 19]. The Schur complement implementation requires using block Gauss elimination to remove the fine-scale terms from the consistent tangent matrix. The resulting tangent matrix contains modified coarse-scale terms that take into account the fine-scale influence. If needed, the fine-scale terms can be obtained via post-processing. The Schur complement form of the consistent tangent matrix reduces the problem size, and provides for much greater parallel scalability.

Primary emphasis in this paper is placed on solving several canonical three-dimensional problems for which experimental and numerical results are available in the literature. We show that using the Newton–Schur (NS) approach not only improves convergence, but also allows for much more efficient use of parallel resources. In the first section, we introduce the variational multiscale framework and describe the decomposition of the problem into a coarse- and fine-scale subproblem.

In the next section, we present the consistent linearization of the Navier–Stokes equations and incorporate the Schur complement form of the consistent tangent matrix to complete the Newton–Schur solution approach. We then present a number of numerical examples that illustrate the advantages of the consistent NS solution approach.

2. GOVERNING EQUATIONS

Let Ω be a bounded open domain and Γ be its boundary, which is assumed to be piecewise smooth. Let the velocity vector field be denoted by $\mathbf{v}:\Omega \rightarrow \mathbb{R}^{\mathrm{nd}}$, where 'nd' is the number of spatial dimensions. Let the (kinematic) pressure field be denoted by $p:\Omega \rightarrow \mathbb{R}$. As usual, Γ is divided into two parts, denoted by $\Gamma^{\mathbf{v}}$ and $\Gamma^{\mathbf{t}}$, such that $\Gamma^{\mathbf{v}} \cap \Gamma^{\mathbf{t}} = \emptyset$ and $\Gamma^{\mathbf{v}} \cup \Gamma^{\mathbf{t}} = \Gamma$. $\Gamma^{\mathbf{v}}$, and $\Gamma^{\mathbf{t}}$ are the Dirichlet and Neumann boundaries, respectively. The governing equations for incompressible Navier–Stokes flow can be written as

$$\mathbf{v}\cdot\nabla\mathbf{v} - 2\nu\nabla^2\mathbf{v} + \nabla p = \mathbf{b} \quad \text{in } \Omega \tag{1}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega \tag{2}$$

$$\mathbf{v} = \mathbf{v}^{\mathrm{p}} \quad \text{on } \Gamma^{\mathbf{v}} \tag{3}$$

$$-p\mathbf{n} + \nu(\mathbf{n}\cdot\nabla)\mathbf{v} = \mathbf{t}^{\mathbf{n}} \quad \text{on } \Gamma^{\mathbf{t}} \tag{4}$$

where $\mathbf{v}$ is the velocity, p is the kinematic pressure (pressure divided by density), ∇ is the gradient operator, ∇^2 is the Laplacian operator, $\mathbf{b}$ is the body force, $\nu > 0$ is the kinematic viscosity, $\mathbf{v}^{\mathrm{p}}$ is the prescribed velocity vector field, $\mathbf{t}^{\mathbf{n}}$ is the prescribed traction, and $\mathbf{n}$ is the unit outward normal vector to Γ. Equation (1) represents the balance of linear momentum and Equation (2) represents the continuity equation for an incompressible continuum. Equations (3) and (4) are the Dirichlet and Neumann boundary conditions, respectively.

3. VARIATIONAL MULTISCALE FRAMEWORK

It is well known that the classical mixed formulation for the incompressible Navier–Stokes equations does not produce stable numerical results. The instability of the standard Galerkin formulation may be mathematically explained by the Ladyzhenskaya-Babuska-Brezzi (LBB) *inf-sup* stability condition [1, 20]. To get stable numerical results, one must stabilize the standard Galerkin formulation. The variational multiscale concept, which stems from the pioneering work by Hughes [4], decomposes the underlying fields into coarse or resolvable scales and subgrid or unresolvable scales.

3.1. Multiscale decomposition

Let us divide the domain Ω into N non-overlapping subdomains Ω^e (which in the finite element context will be elements) such that

$$\Omega = \bigcup_{e=1}^{N} \Omega^e \tag{5}$$

DOI: 10.1002/fld

The boundary of the element Ω^e is denoted by Γ^e. We decompose the velocity field $\mathbf{v}(\mathbf{x})$ into coarse-scale and fine-scale components, indicated as $\bar{\mathbf{v}}(\mathbf{x})$ and $\mathbf{v}'(\mathbf{x})$, respectively.

$$\mathbf{v}(\mathbf{x}) = \bar{\mathbf{v}}(\mathbf{x}) + \mathbf{v}'(\mathbf{x}) \tag{6}$$

Likewise, we decompose the weighting function $\mathbf{w}(\mathbf{x})$ into coarse-scale $\bar{\mathbf{w}}(\mathbf{x})$ and fine-scale $\mathbf{w}'(\mathbf{x})$ components.

$$\mathbf{w}(\mathbf{x}) = \bar{\mathbf{w}}(\mathbf{x}) + \mathbf{w}'(\mathbf{x}) \tag{7}$$

We further make an assumption that the fine-scale components vanish along each element boundary.

$$\mathbf{v}'(\mathbf{x}) = \mathbf{w}'(\mathbf{x}) = \mathbf{0} \quad \text{on } \Gamma^e, \;\; e = 1, \ldots, N \tag{8}$$

Let $\mathscr{V}$ be the function space for the coarse-scale component of the velocity $\bar{\mathbf{v}}$ and $\mathscr{W}$ be the function space for $\bar{\mathbf{w}}$ that are defined as

$$\mathscr{V} := \{\mathbf{v} \,|\, \mathbf{v} \in (H^1(\Omega))^{\mathrm{nd}}, \mathbf{v} = \mathbf{v}^{\mathrm{p}} \text{ on } \Gamma^{\mathbf{v}}\} \tag{9}$$

$$\mathscr{W} := \{\mathbf{w} \,|\, \mathbf{w} \in (H^1(\Omega))^{\mathrm{nd}}, \mathbf{w} = \mathbf{0} \text{ on } \Gamma^{\mathbf{v}}\} \tag{10}$$

Let $\mathscr{V}'$ be the function space for both the fine-scale component of the velocity $\mathbf{v}'$ and its corresponding weighting function $\mathbf{w}'$, and is defined as

$$\mathscr{V}' := \{\mathbf{v} \,|\, \mathbf{v} \in (H^1(\Omega^e))^{\mathrm{nd}}, \mathbf{v} = \mathbf{0} \text{ on } \Gamma^e, e = 1, \ldots, N\} \tag{11}$$

In theory, we could decompose the pressure field into coarse-scale and fine-scale components. However, for simplicity we assume that there are no fine-scale terms for the pressure $p(\mathbf{x})$ and for its corresponding weighting function $q(\mathbf{x})$. Hence, the function space for the fields $p(\mathbf{x})$ and $q(\mathbf{x})$ is $\mathscr{P}$

$$\mathscr{P} := \{p \,|\, p \in L^2(\Omega)\} \tag{12}$$

where $L^2(\Omega)$ is the space of square-integrable functions on the domain Ω. For further details on function spaces refer to Brezzi and Fortin [20].

3.2. *Two-level classical mixed formulation*

After substitution of Equations (6) and (7) into the classical mixed formulation and decomposing the problem into a coarse- and fine-scale subproblem, the resulting weak form can be written as two parts. The *coarse-scale problem* can be written as

$$(\bar{\mathbf{w}}, (\bar{\mathbf{v}} + \mathbf{v}') \cdot \nabla(\bar{\mathbf{v}} + \mathbf{v}')) + (\nabla \bar{\mathbf{w}}, 2\nu \nabla(\bar{\mathbf{v}} + \mathbf{v}')) - (\nabla \cdot \bar{\mathbf{w}}, p) = (\bar{\mathbf{w}}, \mathbf{b}) + (\bar{\mathbf{w}}, \mathbf{h})_{\Gamma_t} \quad \forall \bar{\mathbf{w}} \in \bar{\mathscr{W}} \tag{13}$$

$$(q, \nabla \cdot (\bar{\mathbf{v}} + \mathbf{v}')) = 0 \quad \forall q \in \mathscr{P} \tag{14}$$

The *fine-scale problem* can be written as

$$(\mathbf{w}', (\bar{\mathbf{v}} + \mathbf{v}') \cdot \nabla(\bar{\mathbf{v}} + \mathbf{v}')) + (\nabla \mathbf{w}', 2\nu \nabla(\bar{\mathbf{v}} + \mathbf{v}')) - (\nabla \cdot \mathbf{w}', p) = (\mathbf{w}', \mathbf{b}) + (\mathbf{w}', \mathbf{h})_{\Gamma_t} \quad \forall \mathbf{w}' \in \mathscr{W}' \tag{15}$$

For convenience, we define the L^2 inner-product over a spatial domain, K, as

$$(\mathbf{a}, \mathbf{b})_K = \int_K \mathbf{a} \cdot \mathbf{b} \, \mathrm{d}K \tag{16}$$

DOI: 10.1002/fld

The subscript K will be dropped if K is the whole of Ω, that is $K=\Omega$. Notice that the weak form is nothing more than the classical mixed formulation written for the coarse- and fine-scale variables, with the exception that there is no weak form of the fine-scale incompressibility constraint. For a more detailed derivation of the weak form, see [12].

3.3. *Fine-scale interpolation and bubble functions*

If one chooses a single bubble function for interpolating the fine-scale variables (similar to the MINI element [21]), then we have

$$\mathbf{v}'=b^e\boldsymbol{\beta}, \quad \mathbf{w}'=b^e\boldsymbol{\gamma} \tag{17}$$

where b^e is a bubble function, and $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are constant vectors. The gradients of the fine-scale velocity and weighting functions are

$$\nabla\mathbf{v}'=\boldsymbol{\beta}\nabla b^{e\mathrm{T}}, \quad \nabla\mathbf{w}'=\boldsymbol{\gamma}\nabla b^{e\mathrm{T}} \tag{18}$$

where ∇b^e is a $\dim\times 1$ vector of the derivatives of the bubble function.

4. CONSISTENT NS SOLUTION STRATEGY

For the consistent NS method, we treat Equations (13)–(15) as global residuals and obtain the solution using an iterative update equation.

4.1. *Vector residual*

First, we express the solution field and its weighting function in terms of nodal values $\bar{\mathbf{v}}=\hat{\bar{\mathbf{v}}}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$ and $\bar{\mathbf{w}}=\hat{\bar{\mathbf{w}}}^{\mathrm{T}}\mathbf{N}^{\mathrm{T}}$, respectively (where $\mathbf{N}$ is a row vector of shape functions for each node). After substituting these expressions into the global residuals, the vector residuals, $\mathbf{R}$, that are the sum contributions of the vector residuals at the element level, $\mathbf{R}^e$, can be written as

$$\begin{aligned}\mathbf{R}^e_{\mathrm{c}}(\bar{\mathbf{v}};\mathbf{v}',p) :=& \int_\Omega (\mathbf{N}^{\mathrm{T}}\odot\mathbf{I})((\bar{\mathbf{v}}+\mathbf{v}')\cdot\nabla)(\bar{\mathbf{v}}+\mathbf{v}')\,\mathrm{d}\Omega+2\nu\int_\Omega ((\mathbf{DN})\mathbf{J}^{-1}\odot\mathbf{I})\mathrm{vec}[\nabla\bar{\mathbf{v}}+\nabla\mathbf{v}']\,\mathrm{d}\Omega\\ &-\int_\Omega \mathrm{vec}[\mathbf{J}^{-\mathrm{T}}(\mathbf{DN})^{\mathrm{T}}]p\,\mathrm{d}\Omega-\int_\Omega(\mathbf{N}^{\mathrm{T}}\odot\mathbf{I})\mathbf{b}\,\mathrm{d}\Omega-\int_{\Gamma_t}(\mathbf{N}^{\mathrm{T}}\odot\mathbf{I})\mathbf{h}\,\mathrm{d}\Gamma\end{aligned} \tag{19}$$

$$\mathbf{R}^e_{\mathrm{p}}(\bar{\mathbf{v}};\mathbf{v}') := -\int_\Omega \mathbf{N}^{\mathrm{T}}\nabla\cdot(\bar{\mathbf{v}}+\mathbf{v}')\,\mathrm{d}\Omega \tag{20}$$

$$\begin{aligned}\mathbf{R}^e_{\mathrm{f}}(\bar{\mathbf{v}};\mathbf{v}',p) :=& \int_{\Omega_e} b^e\mathbf{I}((\bar{\mathbf{v}}+\mathbf{v}')\cdot\nabla)(\bar{\mathbf{v}}+\mathbf{v}')\mathrm{d}\Omega+2\nu\int_{\Omega_e}(\nabla b^{e\mathrm{T}}\odot\mathbf{I})\mathrm{vec}[\nabla\bar{\mathbf{v}}+\nabla\mathbf{v}']\,\mathrm{d}\Omega\\ &-\int_{\Omega_e}(\nabla b^{e\mathrm{T}}\odot\mathbf{I})\mathrm{vec}[\mathbf{I}]p\,\mathrm{d}\Omega-\int_{\Omega_e} b^e\mathbf{I}\mathbf{b}\,\mathrm{d}\Omega\end{aligned} \tag{21}$$

where the subscripts 'c', 'p', and 'f' stand for *coarse*, *pressure*, and *fine*, and **DN** represents a matrix of the first derivatives of the element shape functions, which is defined for a triangular element as

$$\mathbf{DN} := \begin{bmatrix} \dfrac{\partial N_1}{\partial \xi_1} & \dfrac{\partial N_1}{\partial \xi_2} \\ \vdots & \vdots \\ \dfrac{\partial N_3}{\partial \xi_1} & \dfrac{\partial N_3}{\partial \xi_2} \end{bmatrix} \tag{22}$$

$\mathbf{J}$ is the element Jacobian matrix, vec[·] is an operation that represents a matrix with a vector, and $\odot$ is the Kronecker product [22].

Remark 1

For transient problems, we simply add the time derivative of the velocity to Equation (1). Using the backward Euler method to integrate in time from step n to step $n+1$, we add the following terms, $\int_\Omega (1/\Delta t)(\mathbf{N}^\mathrm{T} \odot \mathbf{I})\bar{\mathbf{v}}_{n+1}\,\mathrm{d}\Omega - \int_\Omega (1/\Delta t)(\mathbf{N}^\mathrm{T} \odot \mathbf{I})\bar{\mathbf{v}}_n\,\mathrm{d}\Omega$ and $\int_{\Omega_e} (b^e/\Delta t)\mathbf{I}\bar{\mathbf{v}}_{n+1}\,\mathrm{d}\Omega - \int_{\Omega_e} (b^e/\Delta t)\mathbf{I}\bar{\mathbf{v}}_n\,\mathrm{d}\Omega$ to Equations (19) and (21), respectively, where $\bar{\mathbf{v}}_n$ is the converged velocity from the previous time step. For further details, see [12].

4.2. *Tangent matrix*

Using a NR-type approach, we obtain the solution in an iterative fashion using the following update equation until the residual is under a prescribed tolerance:

$$\bar{\mathbf{v}}^{i+1} = \bar{\mathbf{v}}^i + \Delta\bar{\mathbf{v}}^i, \quad p^{i+1} = p^i + \Delta p^i, \quad \mathbf{v}'^{i+1} = \mathbf{v}'^i + \Delta\mathbf{v}'^i \tag{23}$$

where the updates at each iteration, i, are calculated from the following system of equations:

$$\begin{bmatrix} \dfrac{\mathrm{D}\mathbf{R}_\mathrm{c}}{\mathrm{D}\bar{\mathbf{v}}} & \dfrac{\mathrm{D}\mathbf{R}_\mathrm{c}}{\mathrm{D}p} & \dfrac{\mathrm{D}\mathbf{R}_\mathrm{c}}{\mathrm{D}\mathbf{v}'} \\ \dfrac{\mathrm{D}\mathbf{R}_\mathrm{p}}{\mathrm{D}\bar{\mathbf{v}}} & \dfrac{\mathrm{D}\mathbf{R}_\mathrm{p}}{\mathrm{D}p} & \dfrac{\mathrm{D}\mathbf{R}_\mathrm{p}}{\mathrm{D}\mathbf{v}'} \\ \dfrac{\mathrm{D}\mathbf{R}_\mathrm{f}}{\mathrm{D}\bar{\mathbf{v}}} & \dfrac{\mathrm{D}\mathbf{R}_\mathrm{f}}{\mathrm{D}p} & \dfrac{\mathrm{D}\mathbf{R}_\mathrm{f}}{\mathrm{D}\mathbf{v}'} \end{bmatrix} \begin{Bmatrix} \Delta\bar{\mathbf{v}} \\ \Delta p \\ \Delta\mathbf{v}' \end{Bmatrix} = - \begin{Bmatrix} \mathbf{R}_\mathrm{c} \\ \mathbf{R}_\mathrm{p} \\ \mathbf{R}_\mathrm{f} \end{Bmatrix} \tag{24}$$

The element matrices, $\mathrm{D}\mathbf{R}^e/\mathrm{D}(\cdot)$, which are assembled to form the consistent tangent matrix, are presented in [12].

5. SCHUR COMPLEMENT IMPLEMENTATION

By applying block Gauss elimination to Equation (24), we obtain the Schur complement representation of the update equation,

$$\begin{bmatrix} \mathbf{K}_{\bar{\mathbf{v}}\bar{\mathbf{v}}} & \mathbf{K}_{\bar{\mathbf{v}}p} \\ \mathbf{K}_{p\bar{\mathbf{v}}} & \mathbf{K}_{pp} \end{bmatrix} \begin{Bmatrix} \Delta\bar{\mathbf{v}} \\ \Delta p \end{Bmatrix} = - \begin{Bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \end{Bmatrix} \tag{25}$$

DOI: 10.1002/fld

where each of the stiffness terms, **K**, represents the assembled element contributions as follows:

$$\begin{aligned}
\mathbf{K}_{\bar{\mathbf{v}}\bar{\mathbf{v}}} &= \mathop{\mathsf{A}}_{e=1}^{\text{Nele}}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{c}}^{e}}{\mathrm{D}\bar{\mathbf{v}}}-\frac{\mathrm{D}\mathbf{R}_{\mathrm{c}}^{e}}{\mathrm{D}\mathbf{v}'}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\mathbf{v}'}\right]^{-1}\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\bar{\mathbf{v}}}\right]\\
\mathbf{K}_{\bar{\mathbf{v}}p} &= \mathop{\mathsf{A}}_{e=1}^{\text{Nele}}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{c}}^{e}}{\mathrm{D}p}-\frac{\mathrm{D}\mathbf{R}_{\mathrm{c}}^{e}}{\mathrm{D}\mathbf{v}'}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\mathbf{v}'}\right]^{-1}\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}p}\right]\\
\mathbf{K}_{p\bar{\mathbf{v}}} &= \mathop{\mathsf{A}}_{e=1}^{\text{Nele}}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{p}}^{e}}{\mathrm{D}\bar{\mathbf{v}}}-\frac{\mathrm{D}\mathbf{R}_{\mathrm{p}}^{e}}{\mathrm{D}\mathbf{v}'}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\mathbf{v}'}\right]^{-1}\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\bar{\mathbf{v}}}\right]\\
\mathbf{K}_{pp} &= \mathop{\mathsf{A}}_{e=1}^{\text{Nele}}\left[-\frac{\mathrm{D}\mathbf{R}_{\mathrm{p}}^{e}}{\mathrm{D}\mathbf{v}'}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\mathbf{v}'}\right]^{-1}\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}p}\right]
\end{aligned}\tag{26}$$

In addition the residuals are given as

$$\begin{aligned}
\mathbf{R}_1 &= \mathop{\mathsf{A}}_{e=1}^{\text{Nele}}\left[\mathbf{R}_{\mathrm{c}}^{e}-\frac{\mathrm{D}\mathbf{R}_{\mathrm{c}}^{e}}{\mathrm{D}\mathbf{v}'}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\mathbf{v}'}\right]^{-1}\mathbf{R}_{\mathrm{f}}^{e}\right]\\
\mathbf{R}_2 &= \mathop{\mathsf{A}}_{e=1}^{\text{Nele}}\left[\mathbf{R}_{\mathrm{p}}^{e}-\frac{\mathrm{D}\mathbf{R}_{\mathrm{p}}^{e}}{\mathrm{D}\mathbf{v}'}\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}^{e}}{\mathrm{D}\mathbf{v}'}\right]^{-1}\mathbf{R}_{\mathrm{f}}^{e}\right]
\end{aligned}\tag{27}$$

From Equation (25) we update the coarse-scale velocity and pressure. We then solve for the fine-scale increment from

$$\Delta\mathbf{v}'=\left[\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}}{\mathrm{D}\mathbf{v}'}\right]^{-1}\left[-\mathbf{R}_{\mathrm{f}}-\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}}{\mathrm{D}\bar{\mathbf{v}}}\Delta\bar{\mathbf{v}}-\frac{\mathrm{D}\mathbf{R}_{\mathrm{f}}}{\mathrm{D}p}\Delta p\right]\tag{28}$$

Again, the coarse- and fine-scale velocity and pressure are updated until convergence. In addition to faster convergence properties obtained for the NR-type solution procedure, the Schur complement representation increases parallel performance by taking advantage of the fine-scale terms vanishing on the boundary of the element. The fine-scale computations can therefore take place at the element level without the need for inter-processor communication.

6. NUMERICAL RESULTS

The above formulation was implemented in C++ on the Turing cluster, which consists of 768 Apple Xservers, each with two 2 GHz G5 processors and 4 GB of RAM, for a total of 1536 processors [23]. The cluster is connected using a high-bandwidth, low-latency Myrinet network from Myricom. The operating system used is Mac OS X Server, version 10.3. The data structures were implemented in parallel using the portable, extensible toolkit for scientific computation (PETSc) [24] Vec and Mat objects (note that the Vec object is different than the vec[·] operator). PETSc, in turn, uses MPICH for parallel communication. Linear solutions for each iteration were obtained using the direct solver provided by PETSc. The tolerance was set to 1E-12 and the maximum number of iterations was set to 50. The code aborts if the max number of iterations is reached without meeting the tolerance.

DOI: 10.1002/fld

Between 1 and 128 processors were used for each simulation and the results were visualized using Tecplot 360 [25] and VisIt 1.8.1 [26]. The mesh was partitioned using a simple block partitioning strategy. The parallel results may be improved with the use of a more sophisticated partitioning algorithm, but that is beyond the scope of this work. In order to avoid deadlock, matrix assembly routines are called by each process after all of its elements have been computed.

6.1. Body force-driven cavity

Using the body force-driven cavity problem, we compare the speedup on a single processor between a consistent formulation using the Schur complement implementation and one without. The geometry and boundary conditions for the body force-driven cavity problem, shown in Figure 1, consists of a unit domain with boundaries that prohibit flow in the tangential and normal directions ($v_x = v_y = 0.0$). The prescribed constant body force, **b**, in components, is given as

$$\begin{aligned}
b_x = {} & (12-24y)x^4+(-24+48y)x^3+(-48y+72y^2-48y^3+12)x^2 \\
& +(-2+24y-72y^2+48y^3)x+1-4y+12y^2-8y^3 \\
& +(4xy-12xy^2+8xy^3-12x^2y+36x^2y^2-24x^2y^3+8x^3y-24x^3y^2+16x^3y^3) \\
& \times(x^2(1-x)^2(2y-6y^2+4y^3))+(2x^2-12x^2y+12x^2y^2 \\
& -4x^3+24x^3y-24x^3y^2+2x^4-12x^4y+12x^4y^2) \\
& \times(-y^2(1-y)^2(2x-6x^2+4x^3))
\end{aligned} \tag{29}$$

$$\begin{aligned}
b_y = {} & (8-48y+48y^2)x^3+(-12+72y-72y^2)x^2 \\
& +(4-24y+48y^2-48y^3+24y^4)x-12y^2+24y^3-12y^4 \\
& +(-2y^2+12y^2x-12y^2x^2+4y^3-24y^3x+24y^3x^2-2y^4+12y^4x-12y^4x^2) \\
& \times(x^2(1-x)^2(2y-6y^2+4y^3))+(-4yx+12yx^2-8yx^3 \\
& +12y^2x-36y^2x^2+24y^2x^3-8y^3x+24y^3x^2-16y^3x^3) \\
& \times(-y^2(1-y)^2(2x-6x^2+4x^3))
\end{aligned} \tag{30}$$

For a unit viscosity, the exact solution is

$$\begin{aligned}
v_x &= x^2(1-x)^2(2y-6y^2+4y^3) \\
v_y &= -y^2(1-y)^2(2x-6x^2+4x^3) \\
p &= x(1-x)
\end{aligned} \tag{31}$$

The results for both the NR approach, presented in [12], and the NS approach, presented here, are shown in Figure 2. Figure 2 shows the velocity in the x-direction along the center of the cavity as y increases from the bottom to the top. Notice that the results are identical for both methods.

Figure 3 compares the computational time required by the NR and NS approaches as the number of processors increases. Notice that for 32 processors, the computational costs are reduced by over 40%. Since the computational advantages of the Schur complement implementation are based

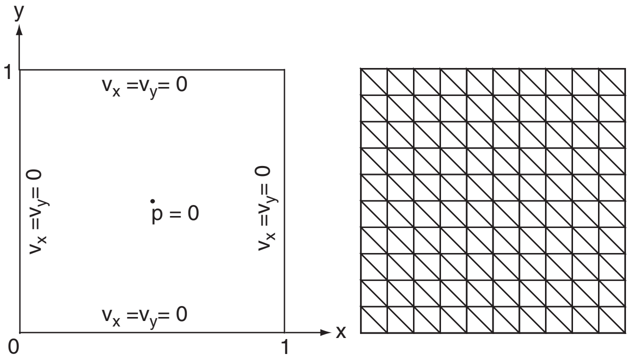

Figure 1. Body force-driven cavity: problem statement and boundary conditions (left) computational mesh (right).

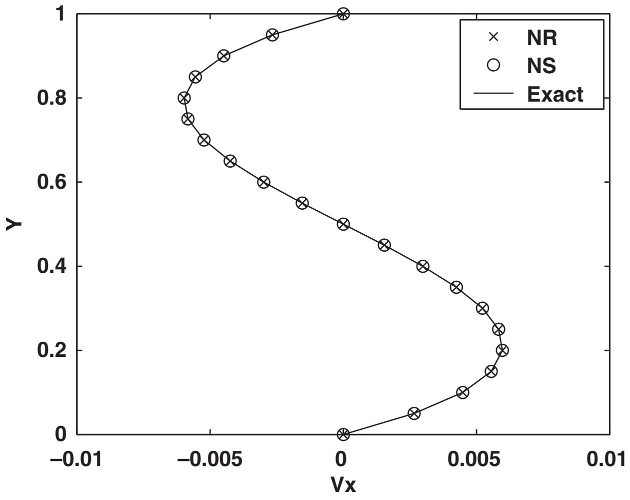

Figure 2. Body force-driven cavity: velocity in the x-direction along the centerline of the cavity for the Newton–Schur (NS) and Newton–Raphson (NR) solution approaches.

largely on parallel features, the benefits of the NS approach grow as the number of processors increases.

6.2. *Three-dimensional lid-driven cavity*

The geometry and boundary conditions for the three-dimensional lid-driven cavity are shown in Figure 4. The computational mesh consists of 633 832 tetrahedral elements with a total of 113 793 nodes. A unit velocity is prescribed in the positive x-direction across the top of a cube of unit volume. The viscosity is adjusted to obtain higher Reynolds numbers. Steady solutions to the three-dimensional lid-driven cavity have only been shown to exist for Reynolds number less than 2000.

DOI: 10.1002/fld

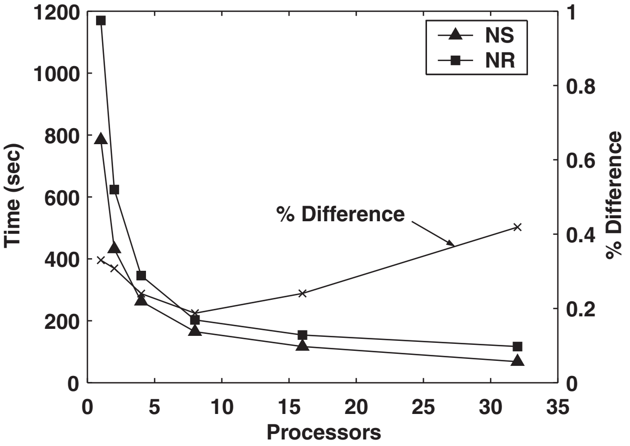

Figure 3. Body force-driven cavity: computational time in seconds comparing the Newton–Schur (NS) and Newton–Raphson (NR) solution approaches for a given number of processors and a fixed problem size of 200 000 degrees of freedom.

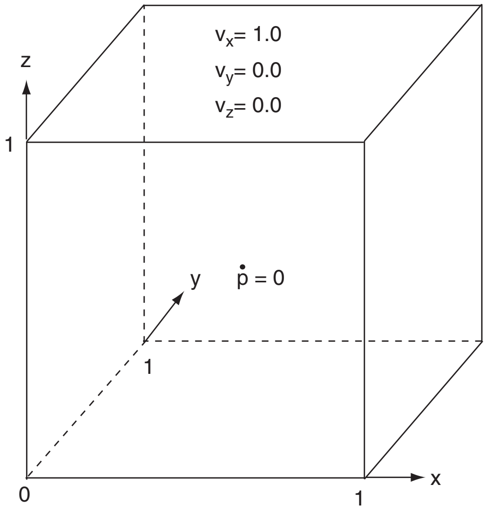

Figure 4. Three-dimensional lid-driven cavity: problem description and boundary conditions. All boundaries are given no-slip wall conditions ($v_x = v_y = v_z = 0$) except for the top boundary, which is given a unit velocity in the x-direction and zero velocity in y and z. The pressure at one node at the center of the domain is prescribed as zero.

Experimental studies of the three-dimensional lid-driven cavity show a number of unsteady motions for $Re{>}2000$, for example eddies and Görtler vortices [27].

Figure 5 shows the velocity in the x-direction along the centerline of the cavity along the height of the cavity for Reynolds number 100, 400, 800, and 1000. Notice that the results correspond well with those reported in [5]. Using the NS solution approach we were able to obtain steady-state

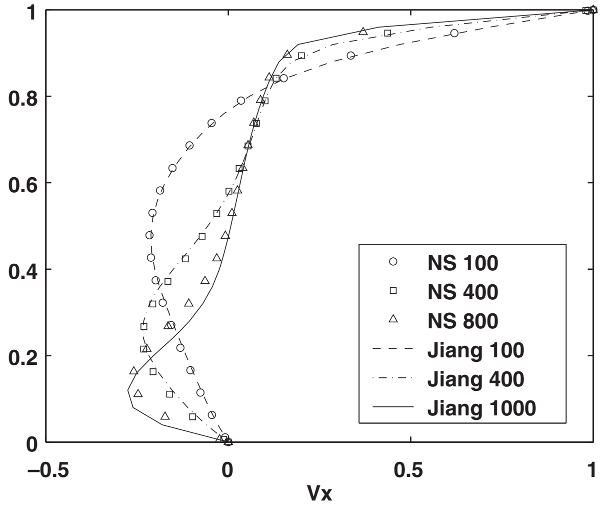

Figure 5. Three-dimensional lid-driven cavity: velocity in the x-direction as measured along the center of the cavity in the z-direction $(x=0, y=0, z)$ compared with the results obtained in [5].

solutions up to $Re=865$. Above $Re=865$, the NS solution approach diverges. In [5] steady-state solutions are obtained for $Re=1000$ using a velocity–pressure–vorticity formulation and a time-marching scheme. To our knowledge, this paper presents the first results for high Reynolds flows for the lid-driven cavity without the use of a time-marching scheme. As the Reynolds number increases, the instabilities created by boundary layer flow over concave surfaces, like the corners of the cavity, begin to take effect. These instabilities represent the transition into turbulent flow [28]. The failure to converge above $Re=800$ for the NS solution strategy could be due to the presence of Taylor–Görtler vortices as reported in [5].

Taylor–Görtler vortices occur when the velocity profile approaches zero at the boundary over a concave (and in some cases convex) surface. They are the result of a centrifugal instability. Rayleigh first recognized this instability in 1916 and showed that as the distance increases radially from the center point of a curved surface, the velocity must also increase, otherwise an inviscid axisymmetric instability occurs. In the case of the corners of the three-dimensional cavity, the flow decreases due to friction with the walls causing a decrease in velocity with radial distance.

Figure 6 shows the results of the three-dimensional lid-driven cavity problem for $Re=800$, which were obtained by direct solution of the steady-state Navier–Stokes equations using the NS solution approach. Notice the presence of the Taylor–Görtler vortices shown in the bottom corners of the z-y profile. The presence of these vortices at $Re=800$ may suggest that the transition to turbulent flow may be occurring well before $Re=2000$. Notice the similarity between the results presented in Figure 6 and those reported for $Re=1000$ in [5], which also suggests that unstable flow may be occurring at a lower Reynolds number than previously reported.

6.3. *Three-dimensional jet flow from an orifice*

The three-dimensional jet flow from an orifice problem represents the laminar evolution of vortices formed by flow through a small hole in an infinite no-slip wall. The problem description and

DOI: 10.1002/fld

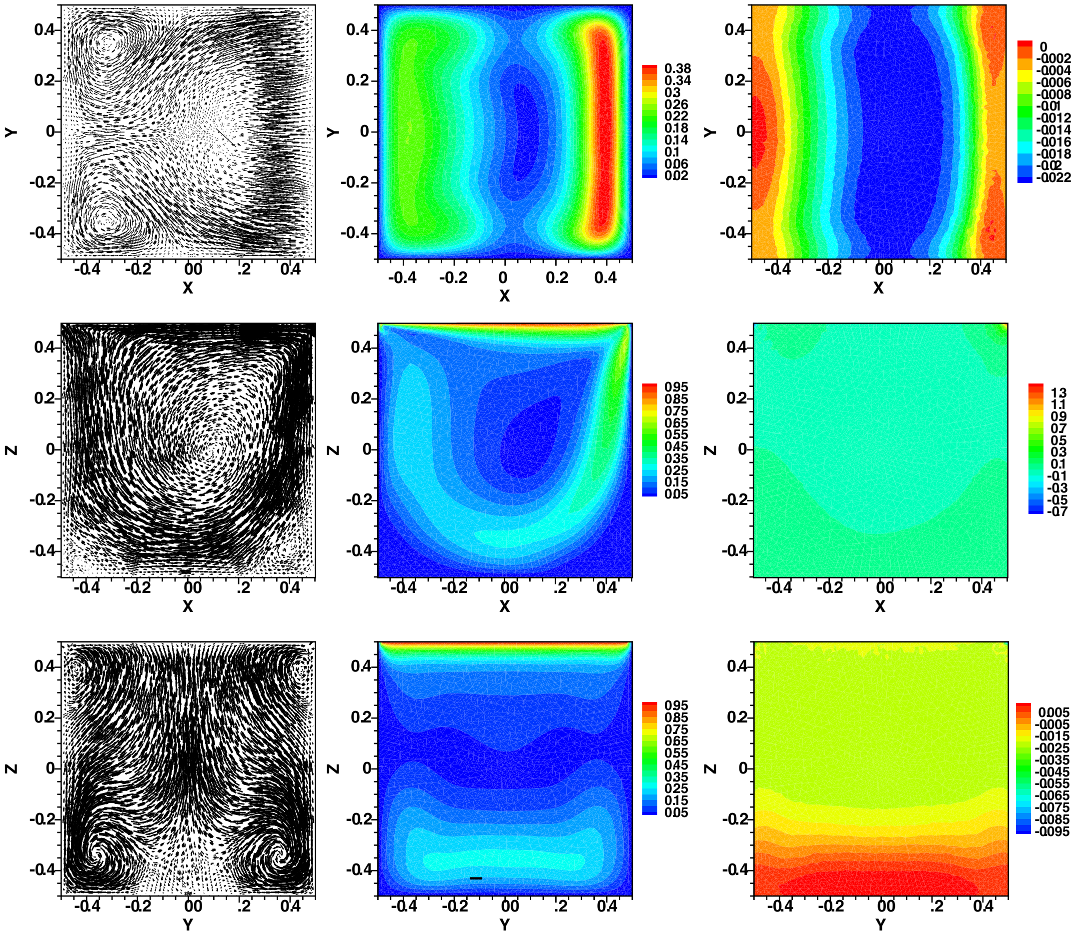

Figure 6. Three-dimensional lid-driven cavity: numerical results. The left column shows vector plots of the velocity, the middle column shows the velocity magnitude, and the right column shows pressure contours in the x–y plane (top row), x–z plane (middle row), and y–z plane (bottom row).

boundary conditions are shown in Figure 7. As the flow is injected into the domain, a mushroom like vortex forms that grows in time. The viscosity was given as 0.001, which corresponds with a Reynolds number of 1000. A time step of 0.01 was also used, which is typical for similar problems.

Figure 8 shows the pressure contours at time $t = 1.0$s. A section of the contours have been removed in order to view the inside of the vortex. Figure 10 shows the velocity magnitude contours and the streamtraces for the three-dimensional jet problem. Notice that the streamtraces are similar to those from the two-dimensional analysis found in [3, 12]. Notice that the pressure contours are moving forward and spreading out. A large region of high pressure forms in front of the spreading region of low pressure. The low pressure region corresponds with the vortex shown in the streamtraces of Figure 10. The pressure contours in the x–z plane of the domain are shown in Figure 9.

DOI: 10.1002/fld

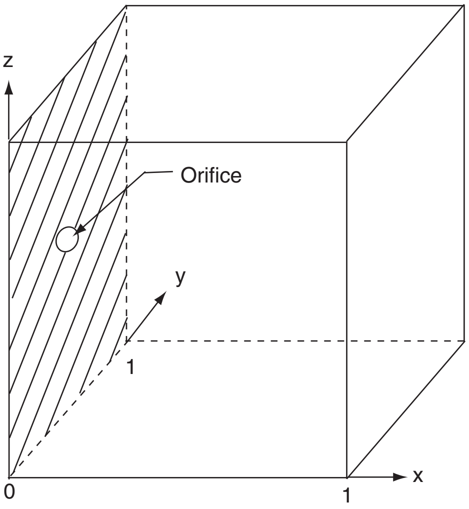

Figure 7. Three-dimensional jet problem: problem description and boundary conditions. All boundaries are given traction-free conditions ($\boldsymbol{\sigma}=0$) except for the no-slip wall in which the orifice lies. The velocity is prescribed as zero in all directions on the no-slip wall. Over the orifice a parabolic velocity is prescribed in the x-direction with maximum magnitude of 1.0 and zero along the edges of the orifice. The velocities in y and z over the orifice are zero.

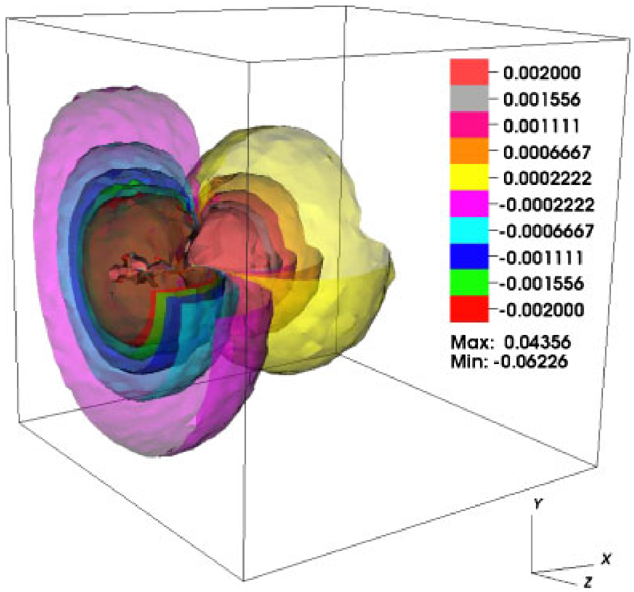

Figure 8. Three-dimensional jet problem: pressure contours with a slice removed to show detail at time $t=1.0$s. The pressure contours are moving forward and spreading sideways. Also note the large region of high pressure that forms in front of a spreading region of low pressure.

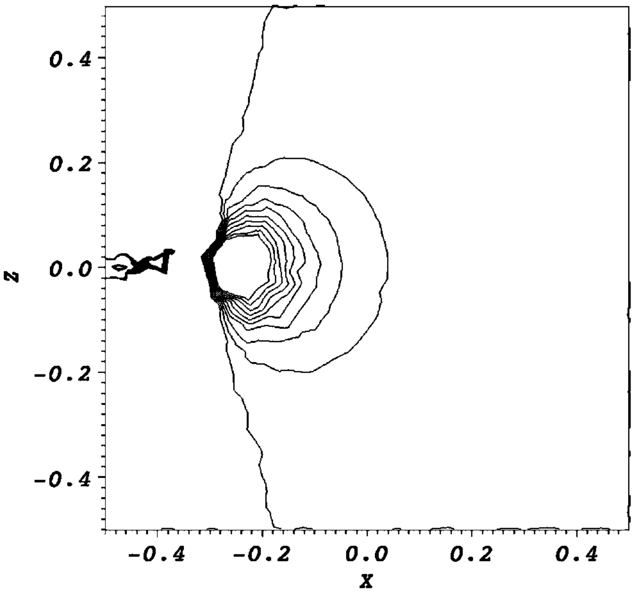

Figure 9. Three-dimensional jet problem: pressure contours in the x–z plane at time $t = 1.0\,\mathrm{s}$.

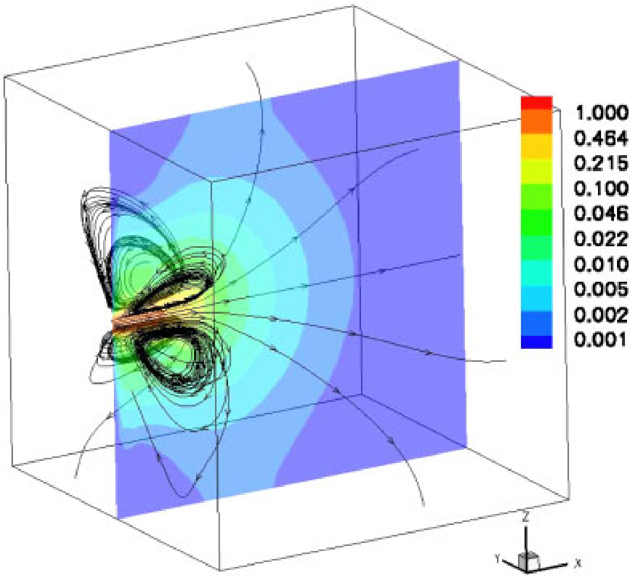

Figure 10. Three-dimensional jet problem: velocity magnitude contours and streamtraces at time $t = 1.0\,\mathrm{s}$. The mushroom-shaped vortex formed by the jet at the orifice is clearly shown by the streamtraces.

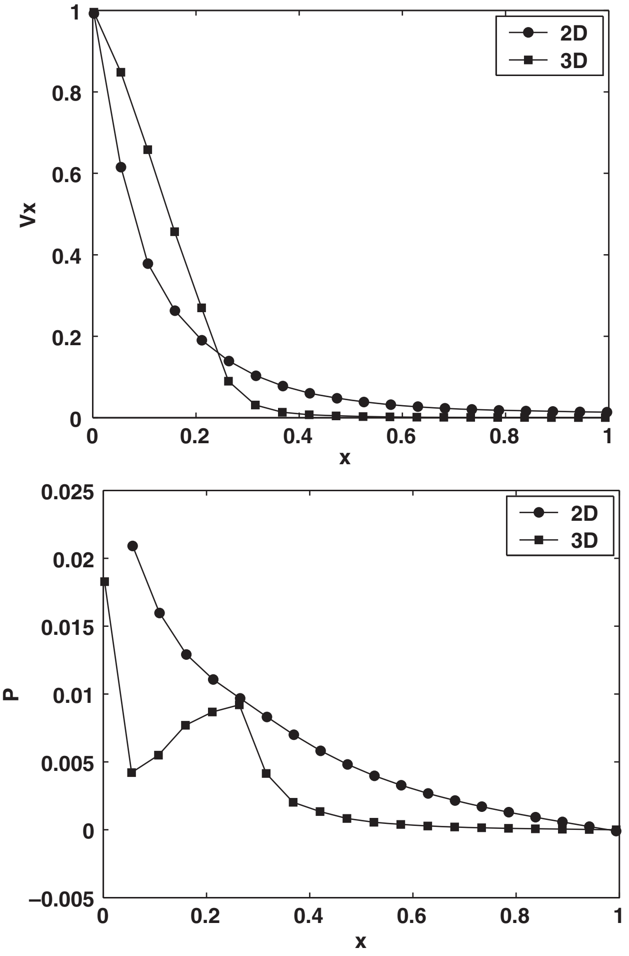

Figure 11. Three-dimensional jet problem: a comparison between two-dimensional and three-dimensional results.

In Figure 11 we show a comparison between the two-dimensional results found in [12] and the three-dimensional results presented herein. Notice that for both pressure and velocity, the results do not match closely suggesting that three-dimensional effects are present for the jet flow problem.

6.4. *Parallel speedup and isoefficiency*

To explore the parallel features of the NS solution approach, a parallel performance study was performed. The lid-driven cavity problem was solved using a variety of mesh sizes and number of processors. The boundary conditions for the three-dimensional lid-driven cavity are shown in Figure 4. The sizes of each mesh are listed in Table I. The largest mesh consists of over one million DOF. The problem was solved for each mesh size on up to 128 processors.

Table I. Mesh sizes in degrees of freedom (DOF) for parallel results.

Mesh	Coarse DOF	Fine DOF	Total DOF
A	34 591	167 289	201 880
B	61 697	296 496	358 193
C	95 803	457 476	553 279
D	135 571	644 676	780 247
E	224 243	1 061 157	1 285 400

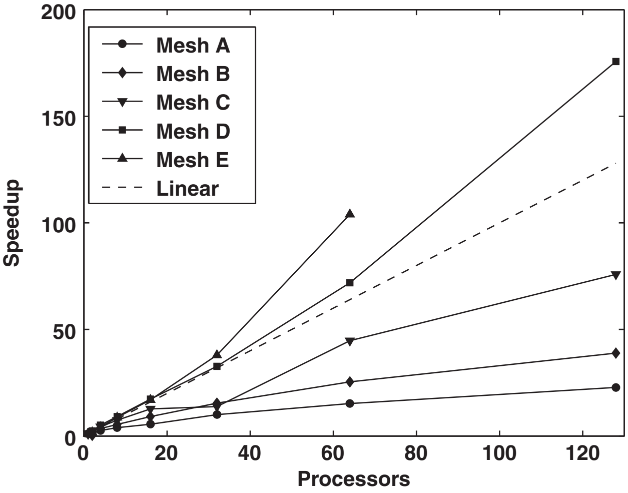

Figure 12. Parallel speedup for various mesh sizes. For the number of degrees of freedom in each mesh, see Table I.

The parallel speedup for each mesh is shown in Figure 12. We measure the speedup as the ratio of the parallel execution time over the serial execution time. Notice that for Meshes D and E superlinear speedup is obtained, while for Meshes A, B, and C, the speedup is less than linear. These results correspond with the expected performance as the problem size increases. For a larger problem size, the ratio of computation time to communication time becomes smaller, allowing for more efficient computation. For a fixed problem size, no algorithm is scalable in the sense that eventually the communication costs will far exceed the computation cost as the number of processors increases. In order to better gage the parallel features of the NS solution approach, an isoefficiency study was performed. Isoefficiency contours characterize the problem size required for a given number of processors to achieve a particular value of efficiency. By efficiency, we are referring to the ratio of the serial costs over the parallel costs. For a given serial computation time, T_1, and parallel time, T_p, for p processors, the efficiency, E, is given as

$$E_\mathrm{p} = \frac{T_1}{pT_\mathrm{p}} \tag{32}$$

DOI: 10.1002/fld

Figure 13 shows the isoefficiency contours for each of the meshes in Table I. An algorithm that is poorly scalable will have almost vertical isoefficiency contours, which correspond to a much greater problem size required to obtain a particular efficiency for a given number of processors [29]. Notice that for greater than 32 processors, the isoefficiency contours are almost horizontal, which suggests that to sustain a certain amount of efficiency, a greater number of processors does not require a substantially larger problem size. The isoefficiency contours in Figure 13 reveal that the NS solution approach is very scalable. Good parallel performance is obtained for a reasonable problem size.

The parallel performance data is summarized in Table II. Values for the theoretical speedup were obtained using Amdahl's law, which estimates the speedup based on the serial fraction of the computation. The speedup cannot exceed the inverse of the serial fraction. For meshes D and E, Amdahl's law is not appropriate because the efficiency is greater than 100%. This may be due to the fact that for such a large problem size, the data does not fit in the cache memory on one processor leading to an extremely long serial computation time. Once the problem has been partitioned over several processors, the data fit into the cache, which results in an efficiency greater than 100%. The superlinear speedup obtained for Meshes D and E is also explained by this circumstance.

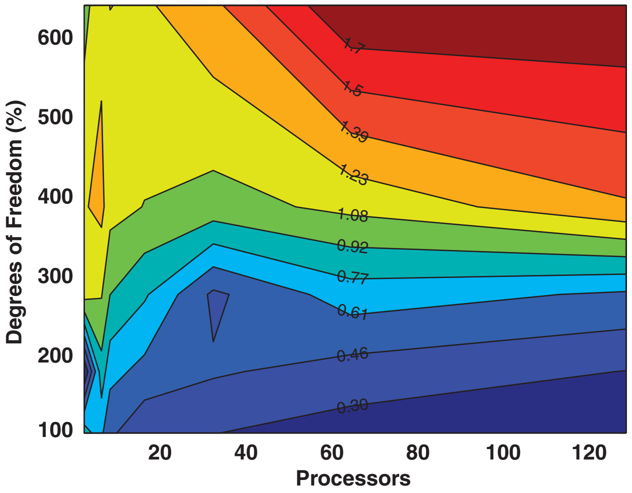

Figure 13. Isoefficiency contours for the Newton–Schur solution approach.

Table II. Summary of parallel results.

Mesh	Max. efficiency obtained (%)	Theoretical speedup	Speedup obtained
A	86	27	22
B	86	42	38
C	92	146	75
D	122	—	175
E	135	—	103

DOI: 10.1002/fld

It is important to note that the number of unknowns is greatly increased by the addition of the fine scale variables. For example, for the lid-driven cavity problem, the total number of DOF for the most refined mesh is 1 285 400, whereas the number of coarse-scale DOF is 224 243, which makes up only 21% of the total. By using the Schur complement implementation, the initial problem size is reduced by roughly 80%. The scalability of the algorithm helps to offset the extra computational cost produced by the dual scales. Although the problem size is much greater, the parallel performance improves as the problem size grows. Also, the fine-scale problem may be further partitioned due to the uncoupled nature of the fine-scale terms.

Remark 2

This example also illustrates the increased cost associated with using the variational multiscale framework. The computational cost associated with the variational multiscale framework is approximately four times the cost of solving the coarse-scale problem alone.

7. CONCLUSIONS

In summary, we have investigated the performance of a Schur complement implementation of a consistent linearization of the incompressible Navier–Stokes equations for a variety of three-dimensional problems. We obtained high Reynolds solutions to the lid-driven cavity problem without the use of a time-marching scheme. The results obtained for $Re=800$ suggest that instabilities in the flow, which represent the onset of the transition to turbulent flow, may occur at Reynolds numbers less than 1000. We also showed that three-dimensional effects may be present for the jet flow through an orifice problem. The results reveal a number of interesting features of the variational multiscale framework. For example, for large problems the dual scales introduce a significant amount of additional DOF. For the lid-driven cavity problem, the fine-scale variable comprised over 80% of the total DOF. Regarding the parallel performance of the NS solution approach, we were able to achieve superlinear speedup for meshes with greater than one million DOF. By means of an isoefficiency study, we also showed that the NS solution algorithm is scalable for reasonable problem sizes.

ACKNOWLEDGEMENTS

The research reported herein was supported by the Computational Science and Engineering Fellowship (D. Z. Turner) and The Department of Energy (K. B. Nakshatrala) through a SciDAC-2 project (Grant No. DOE DE-FC02-07ER64323). This support is gratefully acknowledged. The opinions expressed in this paper are those of the authors and do not necessarily reflect that of the sponsor.

REFERENCES

1. Gunzburger MD. *Finite Element Methods for Viscous Incompressible Flows*: *A Guide to Theory*, *Practice*, *and Algorithms*. Academic Press: San Diego, CA, 1989.
2. Gresho PM, Sani RL. *Incompressible Flow and the Finite Element Method*, *Volume 2*: *Isothermal Laminar Flow*. Wiley: New York, 2000.
3. Donea J, Huerta A. *Finite Element Methods for Flow Problems*. Wiley: West Sussex, England, 2003.
4. Hughes TJR. Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer Methods in Applied Mechanics and Engineering* 1995; **127**:387–401.

DOI: 10.1002/fld

5. Jiang BN. *The Least-squares Finite Element Method*. Springer: New York, 1998.
6. Russo A. Bubble stabilization of finite element methods for the linearized incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1996; **132**:335–343.
7. Masud A, Khurram RA. A multiscale finite element method for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**:1750–1777.
8. Franca LP, Nesliturk A. On a two-level finite element method for the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Engineering* 2001; **52**:433–453.
9. Hughes TJR, Franca LP, Balestra M. A new FEM for CFD: V. Circumventing the Babuška–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accomodating equal order interpolations. *Computer Methods in Applied Mechanics and Engineering* 1986; **59**:85–99.
10. Brooks AN, Hughes TJR. Streamline upwind/Petrov–Galerkin methods for convection dominated flows with emphasis on the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
11. Turner DZ, Nakshatrala KB, Hjelmstad KD. On the stability of bubble functions and a mixed finite element formulation for the Stokes problem. *International Journal for Numerical Methods in Fluids* 2008; DOI: 10.1002/fld.1936.
12. Turner DZ, Nakshatrala KB, Hjelmstad KD. Consistent Newton–Raphson vs. fixed-point for variational multiscale formulations for incompressible Navier–Stokes. 2008. Available at: arXiv as arXiv:0806.3514v1 [cs.NA]. URL: http://arxiv.org/abs/0806.3514.
13. Nakshatrala KB, Masud A, Hjelmstad KD. On finite element formulations for nearly incompressible linear elasticity. *Computational Mechanics* 2008; **41**:547–561.
14. Nakshatrala KB, Turner DZ, Hjelmstad KD, Masud A. A stabilized mixed finite element method for Darcy flow based on a multiscale decomposition of the solution. *Computer Methods in Applied Mechanics and Engineering* 2006; **195**(33–36):4036–4049.
15. Masud A, Khurram RA. A multiscale/stabilized finite element method for the advection–diffusion equation. *Computer Methods in Applied Mechanics and Engineering* 2004; **193**:1997–2018.
16. Koobus B, Farhat C. A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes—applications to vortex shedding. *Computer Methods in Applied Mechanics and Engineering* 2008; **193**:1367–1383.
17. Hughes TJR, Mazzei L, Jansen KE. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science* 2000; **3**:47–59.
18. Kulkarni DV, Tortorelli DA. A domain decomposition based two-level Newton scheme for nonlinear problems. *Domain Decomposition Methods in Science and Engineering*. Springer: Berlin, 2004.
19. Kulkarni DV, Tortorelli DA, Wallin M. A Newton–Schur alternative to the consistent tangent approach in computational plasticity. *Computer Methods in Applied Mechanics and Engineering* 2007; **196**:1169–1177.
20. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer Series in Computational Mathematics, vol. 15. Springer: New York, U.S.A., 1997.
21. Arnold DN, Brezzi F, Fortin M. A stable finite element for the Stokes equations. *Estratto da Calcolo* 1984; **21**(4):337–344.
22. Graham A. *Kronecker Products and Matrix Calculus*: *With Applications*. Halsted Press: New York, 1981.
23. University of Illinois at Urbana-Champaign: Computational Science and Engineering. Turing Cluster Web page, 2008. http://www.cse.uiuc.edu/turing/.
24. Balay S, Buschelman K, Gropp WD, Kaushik D, Knepley MG, McInnes LC, Smith BF, Zhang H. PETSc Web page, 2001. http://www.mcs.anl.gov/petsc.
25. Inc. Tecplot. Tecplot 360, 2008. http://www.tecplot.com.
26. Lawrence Livermore National Laboratory. VisIt 1.8.1, 2005. http://www.llnl.gov/visit.
27. Koseff JR, Street RL. On end wall effects in a lid-driven cavity flow. *Journal of Fluids Engineering* 1984; **106**:385–389.
28. Saric WS. Görtler vortices. *Annual Review of Fluid Mechanics* 1994; **26**:379–409.
29. Heath M. Parallel performance, chapter 4. *CS 554 Course Notes*, University of Illinois, 2006.

DOI: 10.1002/fld