Nodal velocity derivatives of finite element solutions: The FiND method for incompressible Navier-Stokes equations

Ilinca, Florin; Pelletier, Dominique

Publisher's version / Version de l'éditeur:
http://dx.doi.org/10.1002/nme.2328
International Journal for Numerical Methods in Engineering, 76, 4, pp. 455-481, 2008-10-22

INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN ENGINEERING
Int. J. Numer. Meth. Engng 2008; **76**:455–481
Published online 18 March 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nme.2328

Nodal velocity derivatives of finite element solutions: The FiND method for incompressible Navier–Stokes equations

F. Ilinca[1, *, †, ‡] and D. Pelletier[2, §]

[1]*Industrial Materials Institute, National Research Council, 75 de Mortagne, Boucherville, Que., Canada J4B 6Y4*
[2]*École Polytechnique de Montréal, Montréal, Que., Canada H3C 3A7*

SUMMARY

This paper presents the development and application of the finite node displacement (FiND) method to the incompressible Navier–Stokes equations. The method computes high-accuracy nodal derivatives of the finite element solutions. The approach imposes a small displacement to individual mesh nodes and solves a very small problem on the patch of elements surrounding the node. The only unknown is the value of the solution ($\mathbf{u}$, p) at the displaced node. A finite difference between the original and the perturbed values provides the directional derivative. Verification by grid refinement studies is shown for two-dimensional problems possessing a closed-form solution: a Poiseuille flow and a flow mimicking a boundary layer. For internal nodes, the method yields accuracy slightly superior to that of the superconvergent patch recovery (SPR) technique of Zienkiewicz and Zhu (ZZ). We also present a variant of the method to treat boundary nodes. The local discretization is enriched by inserting an additional mesh point very close to the boundary node of interest. Computations show that the resulting nodal derivatives are much more accurate than those obtained by the ZZ SPR technique. Copyright © 2008 John Wiley & Sons, Ltd.

Received 18 June 2007; Revised 1 November 2007; Accepted 19 January 2008

KEY WORDS: Navier–Stokes equations; finite elements; nodal derivatives; node displacement; error estimation; boundary derivatives

1. INTRODUCTION

The accurate computation of nodal values of finite element (FE) solution derivatives is of major importance to many numerical applications. For example, accurate boundary derivatives are crucial in the evaluation of skin friction for drag prediction or Nusselt number in heat transfer applications.

*Correspondence to: F. Ilinca, Industrial Materials Institute, National Research Council, 75 de Mortagne, Boucherville, Que., Canada J4B 6Y4.
†E-mail: florin.ilinca@cnrc-nrc.gc.ca
‡Research Officer, Industrial Materials Institute.
§Canada Research Chair, Mechanical Engineering Department.

Several error estimators used for mesh adaptation rely on some form of projected derivatives to quantify the solution error. A more recent application, design sensitivity analysis, requires nodal values of the solution derivatives to evaluate sensitivity boundary conditions in the sensitivity equation method (SEM). The success or failure of such applications is to a great extent dependent on the accuracy of the extracted boundary solution derivatives. This work is an extension to the incompressible Navier–Stokes equations of the finite node displacement (FiND) method developed by Ilinca and Pelletier [1] for scalar convection–diffusion equations.

It has long been known that the derivatives of the FE solution converge at a slower rate with mesh size refinement than the solution itself [2]. This observation was made initially in structural stress analysis where the stresses converge at a lower rate than the displacements. It was also observed that there are points inside elements at which the convergence rate of the derivatives exceeds the theoretical rate [3, 4]. This has spurred much efforts to develop techniques for evaluating the so-called superconvergent solution derivatives. Babuška and Miller developed extraction formulas for use in structural mechanics (stress analysis). In such problems, the stresses need to be evaluated very accurately at a few critical sites [5]. Although effective, the method proposed by Babuška is too expensive to use in projection error estimators, with the SEM or for the above-mentioned applications.

Many least-squares projection methods have been developed to evaluate solution derivatives. The pioneering work of Zienkiewicz proposed a technique involving the solution to a global least-squares system of order equal to the number of nodes in the mesh [6]. Although effective with linear elements, the method yields wrong answers when applied to quadratic elements [7]. A local least-squares reconstruction was presented and applied successfully to error estimation and mesh adaptation for a spectrum of elements (linear and quadratic triangles as well as 4- and 9-node quadrilaterals) [7–9]. Local averaging techniques in the form of the Zienkiewicz and Zhu (ZZ) error estimator are analyzed in [10] for low-order FEs in elasticity. Averaging techniques were also analyzed by Ainsworth and Craig [11] who propose a framework for developing and classifying error estimators. The asymptotic exactness and superconvergent properties of the reconstructed solution are discussed.

A family of projection methods is obtained depending on whether the solution itself or its derivatives are projected and depending on whether the least squares is performed in a continuum (minimizing an integral of squared quantities) or in a discrete fashion (minimizing a summation) [12, 13]. In many instances, the technique relies on the existence of *superconvergent* points that do not always exist. Variants have been proposed to bypass this difficulty [14–18]. However, either they are restricted to linear problems or the derivative estimations suffer from accuracy degradation near the boundary of the computational domain.

The authors have successfully applied local projection methods to adaptive solution of a broad spectrum of problems: incompressible laminar heat transfer [19–22], isothermal turbulent flows [23–25], turbulent heat transfer [26], laminar or turbulent compressible flows [27, 28], and more recently to verification and validation applications [29–31].

One of the main difficulties with the above-mentioned techniques is the lower accuracy that they deliver for boundary derivatives. Although this is not a major impediment in adaptive solution algorithms, it is critical in sensitivity analysis by the SEM. Indeed, for shape parameters, first derivatives of the solution appear in Dirichlet boundary conditions for the sensitivity variable. Second derivatives appear in the case of Neumann boundary conditions. The observed lower accuracy of boundary derivatives seems to occur because the derivatives extracted by most methods are not compatible or consistent with the solution field at or near the boundary [32].

DOI: 10.1002/nme

Duvigneau and Pelletier have recently proposed a Taylor-series-based constrained least-squares procedure to extract high-accuracy boundary derivatives that are compatible with the imposed Dirichlet conditions [32]. Improvements due to this new approach were demonstrated on examples of optimal design of thermal systems [33] and in the fast evaluation of flows on nearby geometries [34]. However, the cost of the Taylor series least-squares method is too high for practical three-dimensional applications.

In an effort to recover the boundary fluxes with higher accuracy, Carey [35] and Carey *et al.* [36] used the FE equations on nodes having Dirichlet boundary conditions to recover the boundary flux. The technique is based on the work of Wheeler [37], which recovers superconvergent derivatives for two-point boundary value problems. An extension to internal nodes is proposed, which can be applied for one-dimensional problems, but which could not be extended to unstructured multi-dimensional meshes. Evaluation of boundary fluxes was also investigated by Hughes *et al.* [38] in the context of global conservation. They introduce a mixed formulation with an auxiliary field, which amounts to the flux on the Dirichlet portion of the boundary. The auxiliary flux is shown to complete the global conservation of the FE solution and it is observed to have convergence characteristics superior to that of the FE derivatives. Kvamsdal [39] and Melbo and Kvamsdal [40] use a similar approach (variationally consistent postprocessing) to recover surface forces on FE boundaries.

The FiND method was introduced by Ilinca and Pelletier [1] for the computation of accurate nodal derivatives in the case of scalar convection–diffusion equations. The approach is based on imposing a finite displacement on vertices of the mesh and on solving a small local problem on the patch of elements surrounding each node. The only unknown of the resulting equation is the solution at the perturbed point. A finite difference between the values of the solution at the perturbed and original locations of the mesh node yields the directional derivative. For internal nodes, the method yields accuracy slightly superior to that of the superconvergent patch recovery (SPR) technique of Zienkiewicz and Zhu [7]. Significant improvement over the ZZ method is observed for boundary points.

In this paper, we extend the FiND method to the computation of nodal velocity derivatives in the case of the Navier–Stokes equations. The proposed method is cost effective as only small systems of equations corresponding to the displaced nodes are computed (4×4 for the Navier–Stokes equations in three dimensions). We show, via grid refinement studies, that the convergence rate of the extracted derivatives approaches that of the primary variable. We also show that the FiND method yields nodal derivatives equivalent to a discretization of the solution gradient by the same interpolation functions used for the primary variable.

On boundary nodes, the local discretization is enriched by inserting an additional mesh point very close to the boundary node of interest. The location of the additional point is given by applying a finite displacement to boundary nodes in the direction normal to the boundary and towards the interior of the domain. A finite difference between the solution at the additional point and the solution at the boundary node yields the normal derivative at the boundary. The method is shown to provide nodal gradients that are consistent with the auxiliary fluxes [38] on Dirichlet boundary nodes. However, unlike the mixed methods based on the auxiliary flux, which provide the boundary flux or force, the FiND method computes the derivatives of each component of the velocity. Boundary nodal derivatives computed with this new method are also much more accurate than those provided by the SPR technique [7].

In this work we focus on the recovery of velocity nodal derivatives. Although the FiND method is also able to provide the nodal derivatives of the pressure, we have observed that the dual role

DOI: 10.1002/nme

of the pressure for incompressible flows, namely to provide equilibrium of the momentum and to act as a Lagrange multiplier for the incompressibility constraint, makes it more sensitive to the FE discretization. Moreover, as can be observed from the momentum equations, the pressure gradient is expected to have the same accuracy as the second-order derivative of the velocity, which is less accurate than the velocity gradient. The FiND recovery of the nodal pressure gradient and the impact of the stabilized form of the FE equations will be the object of a future investigation.

This paper is organized as follows. First, the model problem and the associated FE formulation are presented. The FiND method for internal nodes is discussed in Section 3. Section 4 presents a variant of the method suited to evaluate boundary derivatives. Section 5 illustrates the performance of the FiND method on a few simple test problems. The paper ends with conclusions.

2. THE MODEL PROBLEM

The FiND method has been introduced in [1] for scalar transport equations. This paper extends the method to the case of the incompressible Navier–Stokes equations. The model problem is expressed as follows:

$$\rho\mathbf{u}\cdot\nabla\mathbf{u} = -\nabla p + \nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})] + \mathbf{f} \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{2}$$

where $\mathbf{u}$, p, ρ, and μ denote the velocity, pressure, density, and viscosity, respectively, and $\mathbf{f}$ is a given source term. The associated boundary conditions are

$$\mathbf{u} = \mathbf{U}_{\mathrm{D}}(\mathbf{x}) \quad \text{for } \mathbf{x}\in\Gamma_{\mathrm{D}} \tag{3}$$

$$\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\hat{\mathbf{n}} - p\hat{\mathbf{n}} = \mathbf{t}(\mathbf{x}) \quad \text{for } \mathbf{x}\in\Gamma_{\mathrm{t}} \tag{4}$$

where Γ_{D} is the portion of the domain boundary $\partial\Omega$, where Dirichlet conditions are imposed, and $\mathbf{t}$ is the traction imposed on $\Gamma_{\mathrm{t}} = \partial\Omega\backslash\Gamma_{\mathrm{D}}$.

In this work we discretize both velocity and pressure using linear continuous interpolants. It is well known that the $P1$–$P1$ element used in conjunction with the Galerkin FE formulation produces spurious oscillations and some form of stabilization is needed. We use the Galerkin least-squares (GLS) method that leads to the following weak formulation:

$$\int_\Omega \rho\mathbf{u}\cdot\nabla\mathbf{u}N_i\,\mathrm{d}\Omega + \int_\Omega \mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\nabla N_i\,\mathrm{d}\Omega - \int_\Omega p\nabla N_i\,\mathrm{d}\Omega - \int_\Omega \mathbf{f}N_i\,\mathrm{d}\Omega$$
$$+\sum_K \int_{\Omega_K} \{\rho\mathbf{u}\cdot\nabla\mathbf{u}+\nabla p-\nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})]-\mathbf{f}\}\tau\mathbf{u}\cdot\nabla N_i\,\mathrm{d}\Omega_K = \int_{\Gamma_{\mathrm{t}}} \mathbf{t}N_i\,\mathrm{d}\Gamma \tag{5}$$

$$\int_\Omega \nabla\cdot\mathbf{u}N_i\,\mathrm{d}\Omega + \sum_K \int_{\Omega_K} \{\rho\mathbf{u}\cdot\nabla\mathbf{u}+\nabla p-\nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})]-\mathbf{f}\}\tau\nabla N_i\,\mathrm{d}\Omega_K = 0 \tag{6}$$

where N_i are continuous, piecewise linear test functions associated with mesh nodes. The first four integrals on the left-hand side of Equation (5) and the first integral in Equation (6) correspond to

DOI: 10.1002/nme

the Galerkin formulation, whereas the integrals over the element interiors are the GLS stabilization terms. The stabilization parameter τ is computed as from References [25, 41]:

$$\tau = \left[\left(\frac{2\rho|\mathbf{u}|}{h_K} \right)^2 + \left(\frac{4\mu}{m_k h_K^2} \right)^2 \right]^{-1/2} \tag{7}$$

where h_K is the size of the element K and m_k is a coefficient set to $\frac{1}{3}$ for linear elements (see [41, 42]).

The velocity (u, v, w) and pressure p are discretized as

$$u \approx u_h = \sum_j u_j N_j(\mathbf{x}) \tag{8}$$

$$v \approx v_h = \sum_j v_j N_j(\mathbf{x}) \tag{9}$$

$$w \approx w_h = \sum_j w_j N_j(\mathbf{x}) \tag{10}$$

$$p \approx p_h = \sum_j p_j N_j(\mathbf{x}) \tag{11}$$

where N_j are nodal shape functions, which in the present case are the same as the test functions.

To alleviate the writing, we regroup the four nodal unknowns into the vector $\mathbf{s}_j = (u_j, v_j, w_j, p_j)^{\mathrm{T}}$. Using this notation, Equations (5) and (6) lead to the following system of equations:

$$\sum_{j=1,N_n} \mathbf{A}_{ij} \mathbf{s}_j = \mathbf{b}_i \quad \text{for } i = 1, N_n \tag{12}$$

$$\mathbf{A}_{ij} = \begin{bmatrix} A_{ij}^{uu} & A_{ij}^{uv} & A_{ij}^{uw} & A_{ij}^{up} \\ A_{ij}^{vu} & A_{ij}^{vv} & A_{ij}^{vw} & A_{ij}^{vp} \\ A_{ij}^{wu} & A_{ij}^{wv} & A_{ij}^{ww} & A_{ij}^{wp} \\ A_{ij}^{pu} & A_{ij}^{pv} & A_{ij}^{pw} & A_{ij}^{pp} \end{bmatrix}, \quad \mathbf{s}_j = \begin{pmatrix} u_j \\ v_j \\ w_j \\ p_j \end{pmatrix}, \quad \mathbf{b}_i = \begin{pmatrix} b_i^u \\ b_i^v \\ b_i^w \\ b_i^p \end{pmatrix} \tag{13}$$

whose unknowns are the nodal values of the velocity and pressure, with N_n being the number of nodes in the mesh.

3. SOLUTION GRADIENTS AT INTERNAL NODES

The solution to system (13) provides the nodal values $\mathbf{s}_j = (u_j, v_j, w_j, p_j)^{\mathrm{T}}$ and from there the solution $\mathbf{s}_h = (u_h, v_h, w_h, p_h)^{\mathrm{T}}$ at any given point inside the computational domain is obtained using Equations (8)–(11). The solution derivative of $\mathbf{s}_h$ along coordinate axes x_k is given by

$$\frac{\partial \mathbf{s}_h}{\partial x_k} = \sum_j \mathbf{s}_j \frac{\partial N_j}{\partial x_k} \tag{14}$$

DOI: 10.1002/nme

where for three-dimensional problems $k=1,2,3$ (with $x_1=x$, $x_2=y$, $x_3=z$). Because the interpolation functions N_j are C^0 continuous, the derivatives of $\mathbf{s}_h$ will be discontinuous across element faces. The degree of the interpolant for the derivative is also lower than that of the primary variable $\mathbf{s}_h$. For example, linear interpolation functions for $\mathbf{s}_h$ lead to piecewise constant gradients.

3.1. General procedure

One would expect a better approximation of the solution derivatives if the gradients were known at the mesh nodes and interpolated using the basis functions of the primary variable. We propose a method for computing such nodal gradients that exhibit the same grid convergence rate as the solution itself. We start from the definition of the derivative of $\mathbf{s}$ with respect to x_k at a point $\mathbf{x}$ as

$$\frac{\partial \mathbf{s}}{\partial x_k}(\mathbf{x}) = \lim_{\delta x_k \to 0} \frac{\mathbf{s}(\mathbf{x}+\delta \mathbf{x}_k) - \mathbf{s}(\mathbf{x})}{\delta x_k} \tag{15}$$

where $\delta \mathbf{x}_k = \delta x_k \hat{\mathbf{i}}_k$ is a displacement of magnitude δx_k in the direction $\hat{\mathbf{i}}_k$ representing the unit vector along axis x_k. When we apply (15) to the FE solution to obtain the gradient at the mesh node i (located at $\mathbf{x}_i$), we obtain

$$\frac{\partial \mathbf{s}_h}{\partial x_k}(\mathbf{x}_i) = \lim_{\delta x_k \to 0} \frac{\mathbf{s}_h(\mathbf{x}_i+\delta \mathbf{x}_k) - \mathbf{s}_h(\mathbf{x}_i)}{\delta x_k} \tag{16}$$

Because the nodal gradient makes use of the solution computed at $(\mathbf{x}_i+\delta \mathbf{x}_k)$, it depends on the neighboring element that the point $(\mathbf{x}_i+\delta \mathbf{x}_k)$ is assumed to be in, resulting in different values of the derivative in the neighboring elements as given by Equation (14). This indeterminacy can be lifted if one could determine a unique value for $\mathbf{s}_h(\mathbf{x}_i+\delta \mathbf{x}_k)$. The FiND method considers a small displacement of the node i from its original location $\mathbf{x}_i$ to $(\mathbf{x}_i+\delta \mathbf{x}_k)$ (see Figure 1); the FE solution is then recomputed locally. This provides us with the needed data for $\mathbf{s}(\mathbf{x}_i+\delta \mathbf{x}_k)$ used in Equation (16) to obtain derivatives of the solution at node P_i.

The nodal displacement δx_k is very small, much smaller than the element mesh size. In practice, good results are obtained for a displacement equal to 10^{-4}–10^{-3} of the smallest element size in the patch surrounding node P (see [1] for a detailed discussion). It is therefore reasonable to assume that, when moving node i, the only nodal value affected is that of the node i itself, whereas values at neighboring nodes remain unchanged, so that the only unknown is the value of the solution $\mathbf{s}$ at the perturbed location. This approximation leads to the following small local problem for the

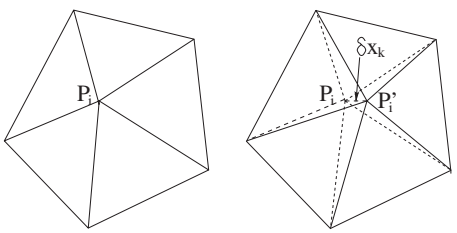

Figure 1. Finite node displacement method.

DOI: 10.1002/nme

nodal solution at the displaced point:

$$\tilde{\mathbf{A}}_{ii}\tilde{\mathbf{s}}_i = \tilde{b}_i - \sum_{j,j\neq i} \tilde{\mathbf{A}}_{ij}\mathbf{s}_j \tag{17}$$

where

$$\tilde{\mathbf{s}}_i = \begin{pmatrix} \tilde{u}_i \\ \tilde{v}_i \\ \tilde{w}_i \\ \tilde{p}_i \end{pmatrix}, \quad \mathbf{s}_j = \begin{pmatrix} u_j \\ v_j \\ w_j \\ p_j \end{pmatrix}, \quad \tilde{\mathbf{A}}_{ij} = \begin{bmatrix} \tilde{A}_{ij}^{uu} & \tilde{A}_{ij}^{uv} & \tilde{A}_{ij}^{uw} & \tilde{A}_{ij}^{up} \\ \tilde{A}_{ij}^{vu} & \tilde{A}_{ij}^{vv} & \tilde{A}_{ij}^{vw} & \tilde{A}_{ij}^{vp} \\ \tilde{A}_{ij}^{wu} & \tilde{A}_{ij}^{wv} & \tilde{A}_{ij}^{ww} & \tilde{A}_{ij}^{wp} \\ \tilde{A}_{ij}^{pu} & \tilde{A}_{ij}^{pv} & \tilde{A}_{ij}^{pw} & \tilde{A}_{ij}^{pp} \end{bmatrix}, \quad \tilde{\mathbf{b}}_i = \begin{pmatrix} \tilde{b}_i^u \\ \tilde{b}_i^v \\ \tilde{b}_i^w \\ \tilde{b}_i^p \end{pmatrix} \tag{18}$$

In the above equations, the parameters depending on the location of the perturbed node $(\mathbf{x}_i + \delta\mathbf{x}_k)$ are identified by a $(\tilde{})$. It indicates that $\tilde{\mathbf{s}}_i = (\tilde{u}_i, \tilde{v}_i, \tilde{w}_i, \tilde{p}_i)^{\mathrm{T}}$ is the solution at the perturbed location of the displaced point $\tilde{\mathbf{s}}_i = \mathbf{s}(\mathbf{x}_i + \delta\mathbf{x}_k)$, whereas $\mathbf{s}_j = \mathbf{s}(\mathbf{x}_j), (j \neq i)$ are independent of δx_k and correspond to the FE solution to the original problem at the neighboring nodes. The solution at the displaced node is, therefore, given by

$$\tilde{\mathbf{s}}_i = \tilde{\mathbf{A}}_{ii}^{-1}\left(\tilde{\mathbf{b}}_i - \sum_{j,j\neq i} \tilde{\mathbf{A}}_{ij}\mathbf{s}_j\right) \tag{19}$$

In the case of linear elements, this reduces to the solution to small systems of equations for each mesh node (3×3 in the case of two-dimensional problems and 4×4 for three-dimensional applications).

Once the value of the solution is computed at the perturbed location $(\mathbf{x}_i + \delta\mathbf{x}_k)$, the derivative of the solution in the direction of the increment $\delta\mathbf{x}_k$ is determined using Equation (16) in the form

$$\frac{\partial \mathbf{s}}{\partial x_k}(\mathbf{x}_i) = \frac{\tilde{\mathbf{s}}_i - \mathbf{s}_i}{\delta x_k} \tag{20}$$

The procedure for computing the derivatives of the solution can be extended to locations other than the mesh nodes. We propose the following formula:

$$\frac{\partial \mathbf{s}}{\partial x_k}(\mathbf{x}) = \frac{\tilde{\mathbf{s}} - \mathbf{s}}{\delta x_k} \tag{21}$$

where $\mathbf{s}$ and $\tilde{\mathbf{s}}$ are the solutions at points $\mathbf{x}$ and $(\mathbf{x}+\delta\mathbf{x}_k)$, respectively, given by

$$\mathbf{s} = \sum_i \mathbf{s}_i N_i \tag{22}$$

$$\tilde{\mathbf{s}} = \sum_i \tilde{\mathbf{s}}_i N_i \tag{23}$$

Note that in Equation (23) the solution at the perturbed point is reconstructed from the solution at the displaced mesh nodes evaluated separately by solving individual local problems. Figure 2 illustrates how the solution at point P is determined by interpolation inside element $P_1P_2P_3$, whereas the solution at the displaced point P' is obtained by interpolation inside the element formed with the displaced points $P_1'P_2'P_3'$. Note that boundary nodes cannot be displaced if this

DOI: 10.1002/nme

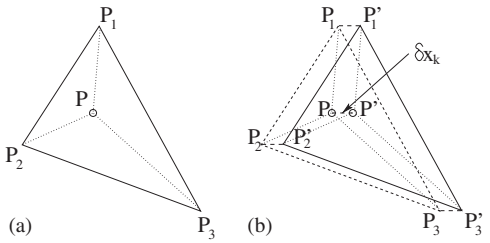

Figure 2. FiND method for extracting the derivative inside elements: (a) original mesh and (b) displaced mesh points.

affects the shape of the computational domain. In such a case, instead of the displaced node we use either an additional mesh node inserted close to the boundary node (see Section 4 for how the method applies to boundary nodes) or a fictitious node whose solution is computed from the nodal values of the solution and of its gradient at the original mesh node location.

Substituting (22) and (23) into Equation (21) yields

$$\frac{\partial \mathbf{s}}{\partial x_k}(\mathbf{x}) = \sum_i \left(\frac{\tilde{\mathbf{s}}_i - \mathbf{s}_i}{\delta x_k} \right) N_i = \sum_i \frac{\partial \mathbf{s}}{\partial x_k}(\mathbf{x}_i) N_i \tag{24}$$

which shows that the solution derivatives $\partial \mathbf{s}/\partial x_k$ are interpolated inside elements from their nodal values (from FiND calculation) using the interpolation functions of the primary variable $\mathbf{s}_h$. This means that our choice for interpolating the derivatives $\partial \mathbf{s}/\partial x_k$ using the interpolation functions for $\mathbf{s}_h$ is not arbitrary; it is determined by the way in which the derivatives are computed by the FiND method.

4. EXTRACTING SOLUTION GRADIENTS AT BOUNDARY NODES

In this section we show how to extract the velocity derivatives on the boundary. For boundary nodes, the derivatives in the normal and tangential directions are first recovered and from there we compute the derivatives in Cartesian directions. In the case of the velocity, this is performed by using

$$\frac{\partial \mathbf{u}}{\partial n} = \frac{\partial \mathbf{u}}{\partial x} n_x + \frac{\partial \mathbf{u}}{\partial y} n_y + \frac{\partial \mathbf{u}}{\partial z} n_z \tag{25}$$

$$\frac{\partial \mathbf{u}}{\partial t_1} = \frac{\partial \mathbf{u}}{\partial x} t_{1x} + \frac{\partial \mathbf{u}}{\partial y} t_{1y} + \frac{\partial \mathbf{u}}{\partial z} t_{1z} \tag{26}$$

$$\frac{\partial \mathbf{u}}{\partial t_2} = \frac{\partial \mathbf{u}}{\partial x} t_{2x} + \frac{\partial \mathbf{u}}{\partial y} t_{2y} + \frac{\partial \mathbf{u}}{\partial z} t_{2z} \tag{27}$$

In the above equations, $\hat{\mathbf{n}}$ is a unit vector normal to the boundary, and $\hat{\mathbf{t}}_1 = (t_{1x}, t_{1y}, t_{1z})$, $\hat{\mathbf{t}}_2 = (t_{2x}, t_{2y}, t_{2z})$ are two orthogonal unit vectors tangent to the boundary.

DOI: 10.1002/nme

4.1. Extracting the tangential derivative

In the case of boundary nodes, the above procedure can be applied directly to obtain the tangential derivative along the boundary surface. For Dirichlet nodes, the value of the solution at the displaced node is obtained directly from the boundary condition (Equation (3)) evaluated at the displaced location. Hence, there is no need to solve an auxiliary FE problem on a patch. The reconstructed tangential derivatives at the boundary surface are given by the derivatives of the Dirichlet boundary condition, which means that they are exact. For example, on boundaries with homogeneous Dirichlet conditions, the tangential derivatives are zero, $\partial\mathbf{u}/\partial t_1 = 0$, $\partial\mathbf{u}/\partial t_2 = 0$. For nodes with Neumann boundary conditions, the tangential derivative is computed from the solution at the displaced node (see Figure 3) using Equation (20) as was the case for internal nodes.

4.2. Extracting the normal derivative

Extracting the normal derivative is more challenging because a boundary node cannot be displaced in the direction normal to the boundary without changing the geometry of the computational domain. Normal derivatives at boundary nodes must be computed in a slightly different manner. The derivative of $\mathbf{u}$ in the direction $\hat{\mathbf{n}}$, the nodal normal to the boundary, is given by

$$\frac{\partial \mathbf{u}}{\partial n}(\mathbf{x}_i) = -\lim_{\delta x_n \to 0} \frac{\mathbf{u}(\mathbf{x}_i + \delta\mathbf{x}_n) - \mathbf{u}(\mathbf{x}_i)}{\delta x_n} \tag{28}$$

where $\delta\mathbf{x}_n = -\delta x_n \hat{\mathbf{n}}$ is a displacement of magnitude δx_n in the normal direction towards the interior of the domain (the minus sign appears because the FE normal vector $\hat{\mathbf{n}}$ is the unit outward normal). To extract the normal derivative, we must determine the solution $\mathbf{u}(\mathbf{x}_i + \delta\mathbf{x}_n)$. To avoid changes in the geometry of the domain, an additional point P' is inserted in the mesh at $\mathbf{x}_i + \delta\mathbf{x}_n$. A local problem is solved for the solution at P'.

The procedure is illustrated in Figure 4 for two-dimensional problems and linear triangular elements. The node P is the boundary mesh node where the normal derivative is needed (located at $\mathbf{x}_i$) and P' is the node inserted at $\mathbf{x}_i + \delta\mathbf{x}_n$. Integration of the FE equation for node P is performed in Ω_P, and Γ_P denotes the portion of the patch boundary $\partial\Omega_P$ coinciding with the domain boundary. Insertion of the point P' determines the division of Ω_P into two sub-volumes denoted by $\Omega_{P'}$ and ω_P as shown in Figure 4. The solution at node P' is obtained by solving the

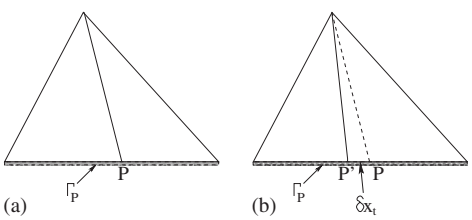

Figure 3. FiND method for extracting the tangential derivative on boundary nodes: (a) original mesh and (b) displaced mesh point.

DOI: 10.1002/nme

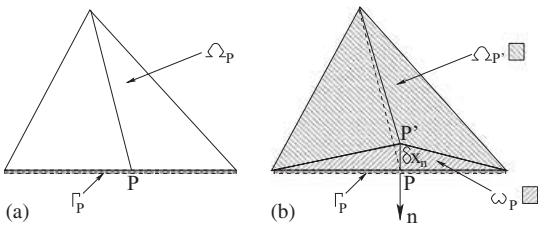

Figure 4. FiND method for extracting the normal derivative on boundary nodes: (a) element connectivity of boundary node P and (b) element connectivity of inserted node P'.

following local FE problem:

$$\int_{\Omega_P} \rho \mathbf{u} \cdot \nabla \mathbf{u} N_{P'} \, \mathrm{d}\Omega_P + \int_{\Omega_P} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \cdot \nabla N_{P'} \, \mathrm{d}\Omega_P - \int_{\Omega_P} p \nabla N_{P'} \, \mathrm{d}\Omega_P - \int_{\Omega_P} \mathbf{f} N_{P'} \, \mathrm{d}\Omega_P$$

$$+ \sum_K \int_{\Omega_{K_{P'}}} \{\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}})] - \mathbf{f}\} \tau \mathbf{u} \cdot \nabla N_{P'} \, \mathrm{d}\Omega_{K_{P'}} = 0 \tag{29}$$

$$\int_{\Omega_P} \nabla \cdot \mathbf{u} N_{P'} \, \mathrm{d}\Omega_P + \sum_K \int_{\Omega_{K_{P'}}} \{\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}})] - \mathbf{f}\} \tau \nabla N_{P'} \, \mathrm{d}\Omega_{K_{P'}} = 0 \tag{30}$$

where $N_{P'}$ is the test function associated with node P'.

Note that for this local FE problem, we assume that the insertion of P' will not affect the values at the vertices of the patch Ω_P. Thus, the only unknowns are the values of $\mathbf{u}$ and p at node P'. Once this local problem is solved, a finite difference between the values of the solution at the new node P' and boundary node P yields the normal derivative of the velocity:

$$\frac{\partial \mathbf{u}}{\partial n}(\mathbf{x}_P) = -\frac{\mathbf{u}(\mathbf{x}_P + \delta \mathbf{x}_n) - \mathbf{u}(\mathbf{x}_P)}{\delta x_n} \tag{31}$$

4.3. Recovering the Cartesian derivatives

Once the normal and tangential derivatives are computed, Cartesian derivatives of the velocity are obtained by solving Equations (25)–(27). Improved accuracy is achieved by projecting the resulting velocity gradient into a divergence-free space since the flows of interest to us are incompressible. The incompressibility condition is, therefore, used as a constraint when solving Equations (25)–(27). This is performed by minimizing the functional

$$J = \frac{1}{2} \left(\mathbf{B} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} - \frac{\partial \mathbf{u}}{\partial \mathbf{d}} \right)^2 + l \mathbf{C}^{\mathrm{T}} \frac{\partial \mathbf{u}}{\partial \mathbf{x}} \tag{32}$$

DOI: 10.1002/nme

where $\mathbf{C}^{\mathrm{T}}=\{1,0,0,0,1,0,0,0,1\}$, l is the Lagrange multiplier associated with the incompressibility constraint and $\mathbf{B}$, $\partial\mathbf{u}/\partial\mathbf{x}$ and $\partial\mathbf{u}/\partial\mathbf{d}$ denote

$$\mathbf{B}=\begin{bmatrix}\mathbf{B}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{B}_{3\times3} & \mathbf{0}_{3\times3}\\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{B}_{3\times3}\end{bmatrix},\quad \mathbf{B}_{3\times3}=\begin{bmatrix}n_x & n_y & n_z\\ t_{1x} & t_{1y} & t_{1z}\\ t_{2x} & t_{2y} & t_{2z}\end{bmatrix},\quad \mathbf{0}_{3\times3}=\begin{bmatrix}0 & 0 & 0\\ 0 & 0 & 0\\ 0 & 0 & 0\end{bmatrix} \tag{33}$$

$$\frac{\partial\mathbf{u}}{\partial\mathbf{x}}=\left(\frac{\partial u}{\partial x},\frac{\partial u}{\partial y},\frac{\partial u}{\partial z},\frac{\partial v}{\partial x},\frac{\partial v}{\partial y},\frac{\partial v}{\partial z},\frac{\partial w}{\partial x},\frac{\partial w}{\partial y},\frac{\partial w}{\partial z}\right)^{\mathrm{T}} \tag{34}$$

$$\frac{\partial\mathbf{u}}{\partial\mathbf{d}}=\left(\frac{\partial u}{\partial n},\frac{\partial u}{\partial t_1},\frac{\partial u}{\partial t_2},\frac{\partial v}{\partial n},\frac{\partial v}{\partial t_1},\frac{\partial v}{\partial t_2},\frac{\partial w}{\partial n},\frac{\partial w}{\partial t_1},\frac{\partial w}{\partial t_2}\right)^{\mathrm{T}} \tag{35}$$

With this notation, minimization of the first term in the right-hand side of (32) leads to Equations (25)–(27), whereas minimization of the second term enforces the incompressibility of the velocity field:

$$\mathbf{C}^{\mathrm{T}}\frac{\partial\mathbf{u}}{\partial\mathbf{x}}=\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0 \tag{36}$$

Minimization of the functional J leads to the following system of equations:

$$\begin{bmatrix}\mathbf{B} & C\\ \mathbf{C}^{\mathrm{T}} & 0\end{bmatrix}\begin{pmatrix}\dfrac{\partial\mathbf{u}}{\partial\mathbf{x}}\\ l\end{pmatrix}=\begin{pmatrix}\dfrac{\partial\mathbf{u}}{\partial\mathbf{d}}\\ 0\end{pmatrix} \tag{37}$$

whose solution represents the divergence-free derivatives of the velocity.

4.4. Relationship between the FiND derivatives and the auxiliary flux

The solution to Equations (29) and (30) involves integrals on very thin elements having one dimension equal to the node displacement δx_n. In this section, we investigate the behavior of those integrals as $\delta x_n\to0$. For this, the integration volume Ω_P in Equations (29) and (30) is decomposed into $\Omega_{P'}$ and ω_P. The integrals on Ω_P can, therefore, be expressed as a sum of integrals on the sub-volumes $\Omega_{P'}$ and ω_P. This leads to the following decomposition of convective and diffusive integrals:

$$\int_{\Omega_P}\rho\mathbf{u}\cdot\nabla\mathbf{u}N_{P'}\,\mathrm{d}\Omega_P=\int_{\Omega_{P'}}\rho\mathbf{u}\cdot\nabla\mathbf{u}N_{P'}\,\mathrm{d}\Omega_{P'}+\int_{\omega_P}\rho\mathbf{u}\cdot\nabla\mathbf{u}N_{P'}\,\mathrm{d}\omega_P \tag{38}$$

$$\begin{aligned}\int_{\Omega_P}\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\nabla N_{P'}\,\mathrm{d}\Omega_P=&\int_{\Omega_{P'}}\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\nabla N_{P'}\,\mathrm{d}\Omega_{P'}\\ &+\int_{\omega_P}\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\nabla N_{P'}\,\mathrm{d}\omega_P\end{aligned} \tag{39}$$

DOI: 10.1002/nme

We will now focus our attention on the behavior of the integrals in (38) and (39) as $\delta x_n \to 0$. First it is easy to see that $\Omega_{P'}$ tends to Ω_P as δx_n tends to zero

$$\lim_{\delta x_n \to 0} \Omega_{P'} = \Omega_P \tag{40}$$

The area of ω_P is proportional to δx_n and its differential is given by

$$\mathrm{d}\omega_P = \delta x_n N_P \, \mathrm{d}\Gamma_P \tag{41}$$

where N_P is the test function associated with node P. Note also that in $\Omega_{P'}$ we have

$$\lim_{\delta x_n \to 0} N_{P'} = N_P \tag{42}$$

whereas in ω_P we have the following identities:

$$\int_{\omega_P} \rho \mathbf{u} \cdot \nabla \mathbf{u} N_{P'} \, \mathrm{d}\omega_P = \int_{\Gamma_P} \frac{1}{2} \rho \mathbf{u} \cdot \nabla \mathbf{u} N_P \delta x_n N_P \, \mathrm{d}\Gamma_P \tag{43}$$

$$\nabla N_{P'} = -\frac{1}{\delta x_n} \hat{\mathbf{n}} \tag{44}$$

Substituting Equations (40)–(44) into Equations (38) and (39) leads to

$$\lim_{\delta x_n \to 0} \int_{\Omega_P} \rho \mathbf{u} \cdot \nabla \mathbf{u} N_{P'} \, \mathrm{d}\Omega_P = \int_{\Omega_P} \rho \mathbf{u} \cdot \nabla \mathbf{u} N_P \, \mathrm{d}\Omega_P + \int_{\Gamma_P} \frac{1}{2} \rho \mathbf{u} \cdot \nabla \mathbf{u} N_P \delta x_n N_P \, \mathrm{d}\Gamma_P \tag{45}$$

$$\begin{aligned} \lim_{\delta x_n \to 0} \int_{\Omega_P} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \cdot \nabla N_{P'} \, \mathrm{d}\Omega_P = & \int_{\Omega_P} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \cdot \nabla N_P \, \mathrm{d}\Omega_P \\ & + \int_{\Gamma_P} \mu (\nabla \mathbf{u}_{\omega_P} + \nabla \mathbf{u}_{\omega_P}^{\mathrm{T}}) \cdot \hat{\mathbf{n}} \left(-\frac{1}{\delta x_n} \right) \delta x_n N_P \, \mathrm{d}\Gamma_P \end{aligned} \tag{46}$$

In the limit $\delta x_n \to 0$, the boundary integral in (45) vanishes and only the volume integral involving the test function N_P remains. A similar result is obtained for the source term in the momentum equation and for the velocity divergence term in the continuity equation. In the diffusion term decomposition (46), the boundary integral does not depend on δx_n and the integral is non-zero as $\delta x_n \to 0$:

$$\begin{aligned} & \lim_{\delta x_n \to 0} \int_{\Omega_P} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \cdot \nabla N_{P'} \, \mathrm{d}\Omega_P \\ & = \int_{\Omega_P} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \cdot \nabla N_P \, \mathrm{d}\Omega_P - \int_{\Gamma_P} \mu (\nabla \mathbf{u}_{\omega_P} + \nabla \mathbf{u}_{\omega_P}^{\mathrm{T}}) \cdot \hat{\mathbf{n}} N_P \, \mathrm{d}\Gamma_P \end{aligned} \tag{47}$$

Note that the velocity gradient in the boundary integral of (47) does not represent the gradient of the original FE solution $\mathbf{u}$, but the gradient inside the elements forming the sub-volume ω_P once the node P' is inserted; it depends on the velocity at the boundary and the velocity at the inserted point P'. A similar treatment applies to the pressure and stabilization terms in the momentum

DOI: 10.1002/nme

equation and to the stabilization term in the continuity equation. Thus, solving for $(\mathbf{u}, p)$ at node P' is equivalent to solving the following FE equation:

$$\begin{aligned}
&\int_{\Omega_P} \rho\mathbf{u}\cdot\nabla\mathbf{u}N_P\,\mathrm{d}\Omega_P + \int_{\Omega_P} \mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\nabla N_P\,\mathrm{d}\Omega_P - \int_{\Omega_P} p\nabla N_P\,\mathrm{d}\Omega_P - \int_{\Omega_P} \mathbf{f}N_P\,\mathrm{d}\Omega_P \\
&\quad + \sum_K \int_{\Omega_{K_P}} \{\rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})] - \mathbf{f}\}\tau\mathbf{u}\cdot\nabla N_P\,\mathrm{d}\Omega_{K_P} \\
&= \int_{\Gamma_P} [\mu(\nabla\mathbf{u}_{\omega_P} + \nabla\mathbf{u}_{\omega_P}^{\mathrm{T}})\cdot\hat{\mathbf{n}} - p\hat{\mathbf{n}}]N_P\,\mathrm{d}\Gamma_P \\
&\quad + \int_{\Gamma_P} \{\rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})] - \mathbf{f}\}_{\omega_P}\tau_{\omega_P}\mathbf{u}\cdot\hat{\mathbf{n}}N_P\,\mathrm{d}\Gamma_P
\end{aligned} \tag{48}$$

$$\begin{aligned}
&\int_{\Omega_P} \nabla\cdot\mathbf{u}N_P\,\mathrm{d}\Omega_P + \sum_K \int_{\Omega_{K_P}} \{\rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})] - \mathbf{f}\}\tau\nabla N_P\,\mathrm{d}\Omega_{K_P} \\
&= \int_{\Gamma_P} \{\rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})] - \mathbf{f}\}_{\omega_P}\tau_{\omega_P}\cdot\hat{\mathbf{n}}N_P\,\mathrm{d}\Gamma_P
\end{aligned} \tag{49}$$

Note that the stabilization terms in the momentum and continuity equations contribute to the boundary integrals over Γ_P representing the portion of the boundary spanned by the elements adjacent to node P. However, these terms involve the stabilization parameter τ_{ω_P} computed inside the elements of ω_P. The evaluation of the stabilization parameter inside degenerated elements, having one side negligible compared with the others, depends on the FE implementation. Observe, however, that the gradient of interest is in the direction normal to the boundary and that the element size in this direction is $\delta x_n \to 0$. Since the stabilization parameter is proportional to the element size in the convective limit and to the square of the element size in the diffusive limit (see Equation (7)), one may conclude that it becomes negligible when δx_n is very small. The net result is that the boundary contribution of the stabilization terms vanishes when $\delta x_n \to 0$. The solution at node P' is, therefore, equivalent to solving

$$\begin{aligned}
&\int_{\Omega_P} \rho\mathbf{u}\cdot\nabla\mathbf{u}N_P\,\mathrm{d}\Omega_P + \int_{\Omega_P} \mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})\cdot\nabla N_P\,\mathrm{d}\Omega_P - \int_{\Omega_P} p\nabla N_P\,\mathrm{d}\Omega_P - \int_{\Omega_P} \mathbf{f}N_P\,\mathrm{d}\Omega_P \\
&\quad + \sum_K \int_{\Omega_{K_P}} \{\rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})] - \mathbf{f}\}\tau\mathbf{u}\cdot\nabla N_P\,\mathrm{d}\Omega_{K_P} \\
&= \int_{\Gamma_P} [\mu(\nabla\mathbf{u}_{\omega_P} + \nabla\mathbf{u}_{\omega_P}^{\mathrm{T}})\cdot\hat{\mathbf{n}} - p\hat{\mathbf{n}}]N_P\,\mathrm{d}\Gamma_P
\end{aligned} \tag{50}$$

$$\int_{\Omega_P} \nabla\cdot\mathbf{u}N_P\,\mathrm{d}\Omega_P + \sum_K \int_{\Omega_{K_P}} \{\rho\mathbf{u}\cdot\nabla\mathbf{u} + \nabla p - \nabla\cdot[\mu(\nabla\mathbf{u}+\nabla\mathbf{u}^{\mathrm{T}})] - \mathbf{f}\}\tau\nabla N_P\,\mathrm{d}\Omega_{K_P} = 0 \tag{51}$$

DOI: 10.1002/nme

where on the right-hand side of Equation (50)

$$\nabla \mathbf{u}_{\omega_P} \cdot \hat{\mathbf{n}} = \left. \frac{\partial \mathbf{u}}{\partial n} \right|_P = - \lim_{\delta x_n \to 0} \frac{\mathbf{u}(P') - \mathbf{u}(P)}{\delta x_n} \tag{52}$$

is the normal derivative of the velocity computed at node P.

Alternatively, an FE problem for a boundary node with a Dirichlet condition can also be expressed using of the auxiliary flux as proposed by Hughes *et al.* [38] in the form:

$$\int_{\Omega_P} \rho \mathbf{u} \cdot \nabla \mathbf{u} N_P \, \mathrm{d}\Omega_P + \int_{\Omega_P} \mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}}) \cdot \nabla N_P \, \mathrm{d}\Omega_P - \int_{\Omega_P} p \nabla N_P \, \mathrm{d}\Omega_P - \int_{\Omega_P} \mathbf{f} N_P \, \mathrm{d}\Omega_P$$
$$+ \sum_K \int_{\Omega_{K_P}} \{\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}})] - \mathbf{f}\} \tau \mathbf{u} \cdot \nabla N_P \, \mathrm{d}\Omega_{K_P} = \int_{\Gamma_P} \mathbf{t}_{\mathrm{n}} N_P \, \mathrm{d}\Gamma_P \tag{53}$$

$$\int_{\Omega_P} \nabla \cdot \mathbf{u} N_P \, \mathrm{d}\Omega_P + \sum_K \int_{\Omega_{K_P}} \{\rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - \nabla \cdot [\mu (\nabla \mathbf{u} + \nabla \mathbf{u}^{\mathrm{T}})] - \mathbf{f}\} \tau \nabla N_P \, \mathrm{d}\Omega_{K_P} = 0 \tag{54}$$

In the above equation, $\mathbf{t}_{\mathrm{n}}$ denotes the normal traction on the boundary Γ_P. Comparing Equations (50) and (53), one concludes that the FiND normal derivative at boundary node P (52) satisfies the following compatibility relationship with the auxiliary boundary flux $\mathbf{t}_{\mathrm{n}}$:

$$\int_{\Gamma_P} [\mu (\nabla \mathbf{u}_{\omega_P} + \nabla \mathbf{u}_{\omega_P}^{\mathrm{T}}) \cdot \hat{\mathbf{n}} - p \hat{\mathbf{n}}] N_P \, \mathrm{d}\Gamma_P = \int_{\Gamma_P} \mathbf{t}_{\mathrm{n}} N_P \, \mathrm{d}\Gamma_P \tag{55}$$

Observe that the FiND method results in a boundary flux represented by a discontinuous discretization that is constant on overlapping patches. Figure 5 illustrates the discretization of the boundary flux on a portion of the boundary for a two-dimensional problem solved using linear continuous interpolation functions for the velocity. When computing the normal derivative at a boundary point (say point P_2 in Figure 5), the boundary flux is constant on the patch of elements surrounding the respective point. Its value is given by the derivative of the velocity at node P_2. This discrete form of the boundary flux is consistent with the FE discretization of the velocity in the sense of the FiND method. For non-planar boundaries, we should define nodal normal directions

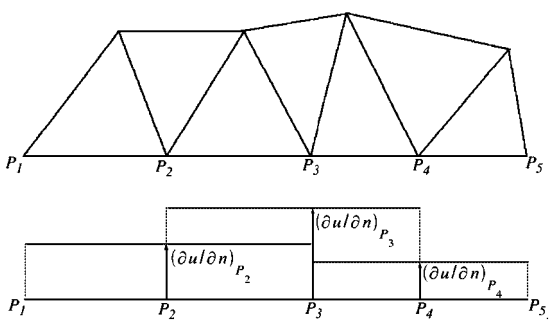

Figure 5. Discrete FiND derivative on boundary nodes.

DOI: 10.1002/nme

on which the boundary nodes would be displaced. A discussion on how to choose the nodal normal direction can be found in [1].

5. NUMERICAL EXAMPLES

The accuracy of the proposed method is studied on two-dimensional problems with known analytical solutions. This provides a rigorous framework to perform systematic grid refinement studies to assess:

- The grid convergence of the FE solution to the exact solution.
- The grid convergence of the reconstructed derivatives to the exact derivatives.
- The effective convergence rate of the derivatives recovered by FiND compared with that of the ZZ local projection method.

The ZZ and FiND methods are verified on several problems. First, we solve a Poiseuille flow for which the velocity changes only in the direction transverse to the flow. Then we consider a manufactured solution [32] behaving like a two-dimensional boundary layer. This problem is solved on both uniform and non-uniform meshes. For the present examples, the ZZ method using continuum and discrete least-squares projections (SPR) yields similar results. Hence, only the results obtained by the continuum (minimizing the integral of squared quantities) method are shown.

5.1. Poiseuille flow

This problem consists in solving the flow through a channel with parallel walls located at $y=\pm h$. In the fully developed flow region, the velocity distribution is parabolic:

$$u=U_0\left(1-\frac{y^2}{h^2}\right), \quad v=0 \tag{56}$$

where U_0 is the maximum velocity. The pressure p varies linearly with x

$$\frac{\partial p}{\partial x}=\mu\frac{\partial^2 u}{\partial y^2}=-\frac{2\mu U_0}{h^2} \tag{57}$$

where μ is the fluid viscosity.

The problem is solved in the square $[-h;h]\times[-h;h]$ with the following boundary conditions: both velocity components are zero at the walls ($y=\pm h$) and imposed from Equation (56) on the inlet ($x=-h$). At the outlet ($x=h$), the vertical velocity v and the normal traction are set to zero. This results in a pressure field given by

$$p(x)=\frac{2\mu U_0}{h}\left(1-\frac{x}{h}\right) \tag{58}$$

Computations are carried out on a sequence of uniform meshes of triangles obtained by mesh halving. The velocity component $u(x,y)$ and its true error $e_u(x,y)=u-u_{\text{ex}}$ for a 20×20 mesh are shown in Figure 6. The results for the y derivative extracted using the ZZ and FiND methods, $\partial u/\partial y\,|_{\text{ZZ}}$ and $\partial u/\partial y\,|_{\text{FiND}}$, are shown in Figure 7 for the 20×20 mesh (for the sake of clarity, we have plotted $-\partial u/\partial y$).

DOI: 10.1002/nme

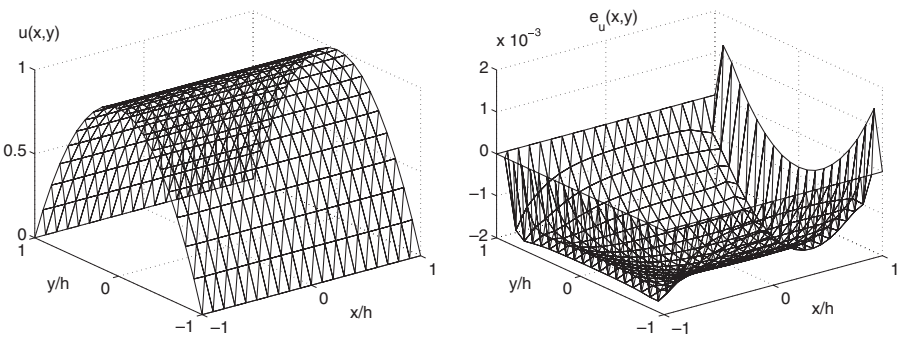

Figure 6. Poiseuille flow: solution (left) and true error (right) on a 20×20 mesh.

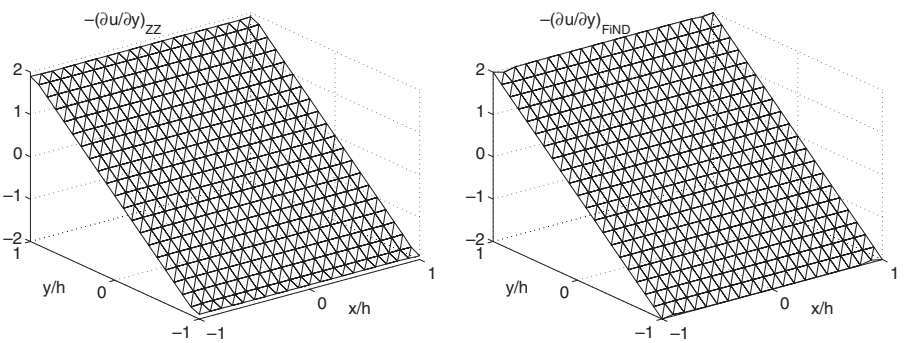

Figure 7. Poiseuille flow: reconstructions of $\partial u/\partial y$ by the ZZ method (left) and the FiND method (right).

The true errors of the reconstructed derivatives are defined as follows:

$$e_{\mathrm{ZZ}} = \left.\frac{\partial u}{\partial y}\right|_{\mathrm{ZZ}} - \frac{\partial u_{\mathrm{ex}}}{\partial y} \tag{59}$$

$$e_{\mathrm{FiND}} = \left.\frac{\partial u}{\partial y}\right|_{\mathrm{FiND}} - \frac{\partial u_{\mathrm{ex}}}{\partial y} \tag{60}$$

and shown to the same scale on Figure 8.

The two methods produce comparable accuracy for nodes inside the computational domain. However, the FiND method yields much higher accuracy for the derivatives at Dirichlet boundary nodes ($y = \pm h$). This indicates the superiority of the FiND method over the ZZ technique. The FiND derivative is very close to the exact derivative at all nodes, except at the wall near the corners of the computational domain where the error is slightly higher.

A grid convergence study of the ZZ and FiND methods was carried out by solving the problem on meshes with 5, 10, 20, and 40 elements in each direction. The results are summarized in

DOI: 10.1002/nme

Figure 9 for the L_2 norm and H_1 semi-norm of the FE solution error, and for the H_1 semi-norm of the reconstruction errors for the ZZ and FiND methods, $\|\mathbf{u}_{ZZ}-\mathbf{u}_{ex}\|_{H_1}$ and $\|\mathbf{u}_{FiND}-\mathbf{u}_{ex}\|_{H_1}$:

$$\|\mathbf{u}_h-\mathbf{u}_{ex}\|_{L_2}=\left\{\int_\Omega(\mathbf{u}_h-\mathbf{u}_{ex})^2\,d\Omega\right\}^{1/2} \tag{61}$$

$$\|\mathbf{u}_h-\mathbf{u}_{ex}\|_{H_1}=\left\{\int_\Omega(\nabla\mathbf{u}_h-\nabla\mathbf{u}_{ex})^2\,d\Omega\right\}^{1/2} \tag{62}$$

$$\|\mathbf{u}_{ZZ}-\mathbf{u}_{ex}\|_{H_1}=\left\{\int_\Omega(\nabla\mathbf{u}_{ZZ}-\nabla\mathbf{u}_{ex})^2\,d\Omega\right\}^{1/2} \tag{63}$$

$$\|\mathbf{u}_{FiND}-\mathbf{u}_{ex}\|_{H_1}=\left\{\int_\Omega(\nabla\mathbf{u}_{FiND}-\nabla\mathbf{u}_{ex})^2\,d\Omega\right\}^{1/2} \tag{64}$$

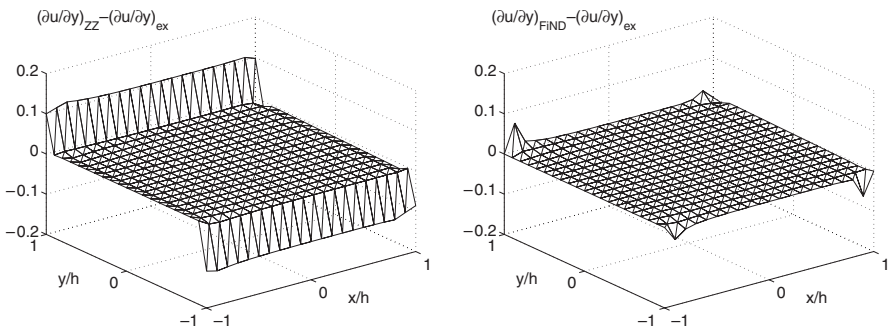

Figure 8. Poiseuille flow: reconstruction errors of $\partial u/\partial y$ by the ZZ method (left) and the FiND method (right).

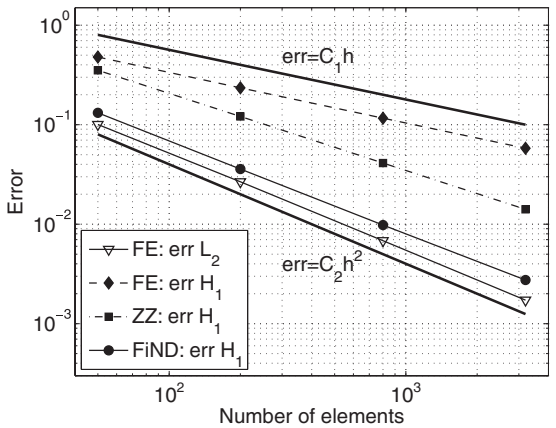

Figure 9. Errors of the finite element solution, ZZ and FiND derivatives.

DOI: 10.1002/nme

For comparison purposes, the reference lines for linear and quadratic grid convergence with respect to h (mesh size) are shown in thick continuous lines above and below the lines with symbols. Several observations can be drawn from these results. First we note that, as expected, the L_2-norm of the error exhibits second-order accuracy with mesh refinement (see inverted open triangles). The H_1 semi-norm of the error $\|\mathbf{u}_h - \mathbf{u}_{\text{ex}}\|_{H_1}$ (solid diamonds), which measures the accuracy of the FE derivatives, exhibits first-order mesh convergence. The error in the ZZ derivatives (solid squares) converges faster than the error in the FE derivatives (the H_1 semi-norm of the true error). The convergence rate of the ZZ derivatives, $\|\mathbf{u}_{\text{ZZ}} - \mathbf{u}_{\text{ex}}\|_{H_1}$, is somewhere between 1 and 2, i.e. higher than the convergence rate of the FE derivatives but lower than that of the L_2-norm of the solution error $\|\mathbf{u}_h - \mathbf{u}_{\text{ex}}\|_{L_2}$. Figure 9 shows that the FiND derivatives (solid circles) are much more accurate than the ZZ derivatives for all meshes. The FiND derivatives exhibit quadratic convergence similar to $\mathbf{u}_h$. That is, $\|\mathbf{u}_{\text{FiND}} - \mathbf{u}_{\text{ex}}\|_{H_1} = \|\nabla\mathbf{u}_{\text{FiND}} - \nabla\mathbf{u}_{\text{ex}}\|_{L_2}$ converges at the same rate as $\|\mathbf{u}_h - \mathbf{u}_{\text{ex}}\|_{L_2}$, i.e. second order.

To further investigate the behavior of the two methods, we have computed separately the mean nodal error of the y derivative for internal nodes and boundary nodes. The results are shown in Figure 10. Again, reference lines for linear and quadratic convergences are included. For internal nodes, the errors of the ZZ and FiND derivatives are of comparable accuracy (solid symbols), with the FiND method appearing slightly more accurate. With mesh refinement, both derivatives exhibit almost second-order accuracy. For Dirichlet boundary nodes, the two methods yield completely different results. The FiND method computes boundary derivatives that are as accurate as derivatives at internal nodes. FiND boundary derivatives (open circles) exhibit the same convergence rate as internal nodes, whereas the ZZ boundary derivatives (open squares) are less accurate than in the interior and the convergence rate is down to one. This poor behavior of the ZZ projection method at boundary points has a notable effect on the global performance of the method. One concludes that the FiND method improves the accuracy of recovered derivatives compared with the ZZ method, especially at boundary points.

5.2. *Boundary layer flow*

The next verification test case is a manufactured solution mimicking the flow along a flat plate [32]. The velocity field is given by

$$u = U_0(1 - e^{-\varepsilon}) \tag{65}$$

$$v = \frac{U_0}{2\delta\sqrt{x}}[1 - (1+\varepsilon)e^{-\varepsilon}] \tag{66}$$

$$p = x^2 - 0.01 \tag{67}$$

where $\delta = \sqrt{\rho U_0/\mu}$ and $\varepsilon = (y - y_0)\sqrt{\rho U_0/(\mu x)}$. The problem is solved on the square $[0.1, 1.1] \times [0, 1]$ and the parameters take the values $U_0 = 1$, $\rho = 1$, $\mu = 0.01$, and $y_0 = 0$. The source term $\mathbf{f}$ is determined such that (65)–(67) are solutions to Equations (1), (2). Dirichlet conditions are imposed on all boundaries using the exact solution, except on the outlet ($x = 1.1$) where the exact traction is imposed in the horizontal direction and the velocity is imposed in the vertical direction.

The horizontal velocity (u) and its true error are shown in Figure 11 for a 20×20 mesh. As can be seen, the velocity exhibits larger errors in the boundary layer near the wall. Results for the y derivative obtained by the ZZ and FiND methods are shown in Figure 12 (for illustration purposes $-\partial u/\partial y$ is shown). The reconstruction errors for the derivative are shown in Figure 13. We observe

DOI: 10.1002/nme

that the ZZ method performs very well inside the computational domain. However, it produces less accurate values of $\partial u/\partial y$ at and near the wall ($y=0$). This is especially true near the inflow where the boundary layer is thinnest. The FiND method produces derivatives of excellent accuracy for internal nodes and improves the accuracy of the derivative estimate at boundary points.

A grid convergence study is carried out for both methods on meshes with 6×6, 11×11, 21×21, and 41×41 nodes. Figure 14 presents error trajectories for the solution and approximations of its derivative. Here again, the FiND method yields superconvergent nodal derivatives (solid circles) with a rate close to that of the solution itself (open triangles). The ZZ derivatives (solid squares) are less accurate and exhibit a lower grid convergence rate.

The performances of the two methods for internal and boundary nodes are compared in Figure 15. Boundary nodes include only points located on the bottom wall ($y=0$). For internal nodes, the ZZ (solid squares) and FiND (solid circles) reconstructed derivatives are of comparable accuracy, with a slightly more accurate prediction for the FiND method. Both derivatives exhibit second-order

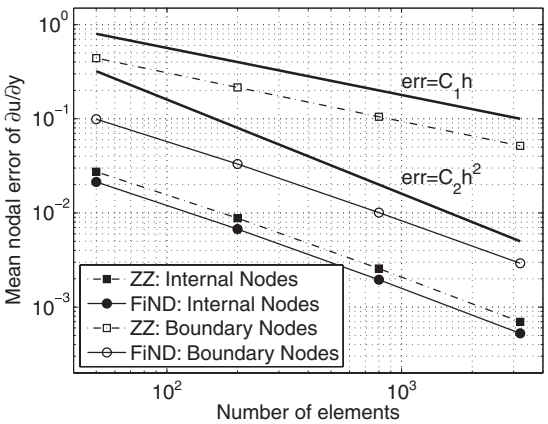

Figure 10. Nodal errors of the ZZ and the FiND derivatives.

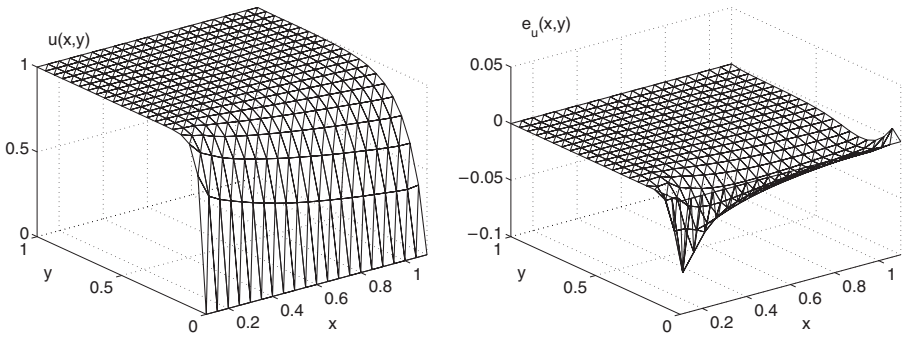

Figure 11. Boundary layer flow: solution (left) and true error (right) on a 20×20 mesh.

DOI: 10.1002/nme

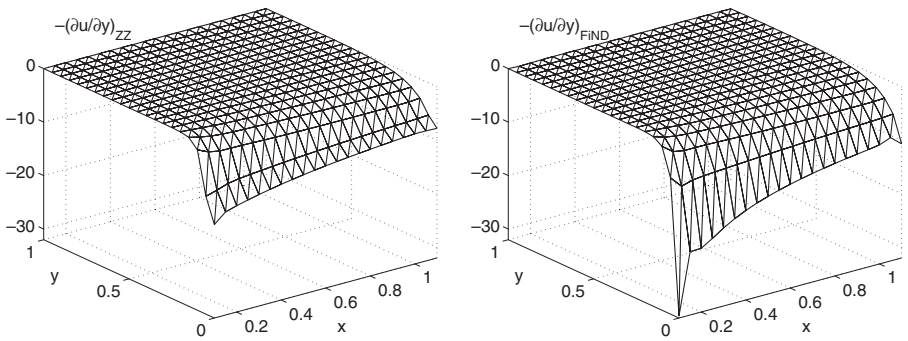

Figure 12. Boundary layer flow: reconstruction of $\partial u/\partial y$ using the ZZ method (left) and the FiND method (right).

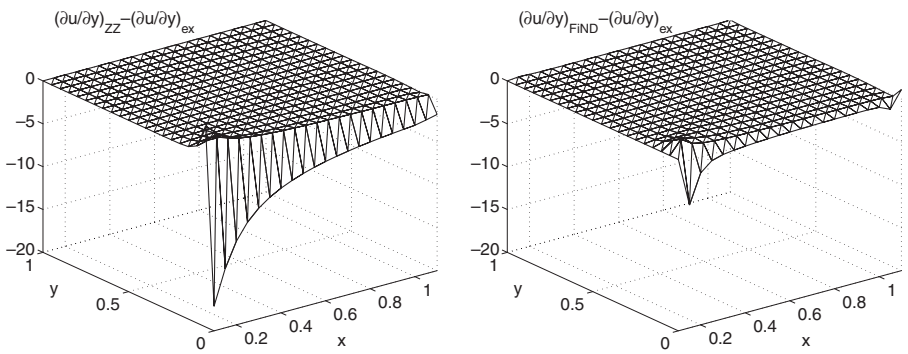

Figure 13. Boundary layer flow: true reconstruction errors of $\partial u/\partial y$ by the ZZ method (left) and the FiND method (right).

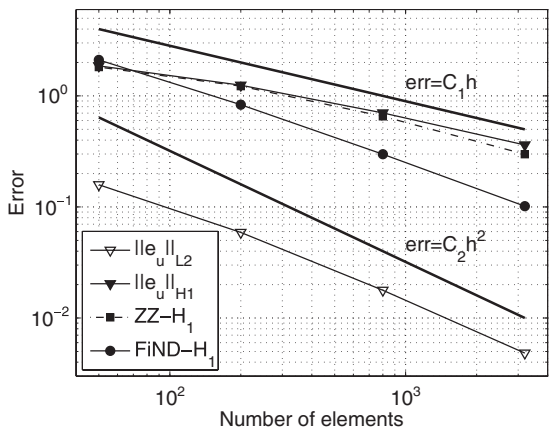

Figure 14. Errors of the finite element solution, ZZ and FiND derivatives.

DOI: 10.1002/nme

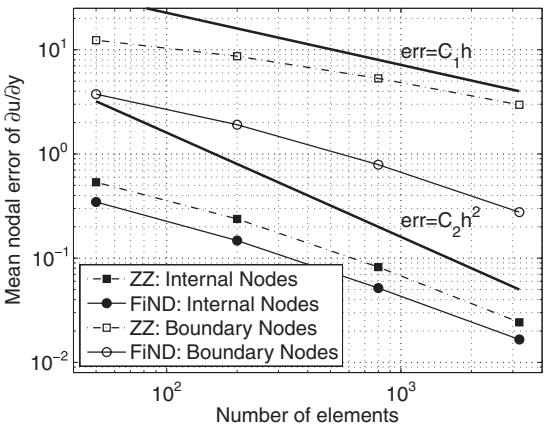

Figure 15. Nodal errors of the ZZ and FiND derivatives.

accuracy with mesh refinement. Along the wall, the two methods provide different results. The FiND derivatives (open circles) have a convergence rate close to $\mathcal{O}(h^{3/2})$ which agree well with observations of Pehlivanov *et al.* [43] for the accuracy of recovered boundary fluxes. The ZZ derivatives (open squares) are less accurate at the wall and the grid convergence rate is close to 1.

We now apply the FiND derivative for computing the wall friction

$$F_{\mathrm{w}} = \int_{\Gamma_{\mathrm{w}}} \mu \frac{\partial u}{\partial y} \, \mathrm{d}\Gamma_{\mathrm{w}} \tag{68}$$

where Γ_{w} denotes the wall boundary ($y=0$). The wall friction computed by the two methods is compared with the analytical value in Figure 16. The wall friction computed from nodally exact values but interpolated using linear functions inside the boundary elements is also shown. As can be seen, the predictions from both FiND and ZZ estimates improve with mesh refinement. However, the FiND derivatives provide a much more accurate wall friction than the ZZ approximation. On the final mesh, the FiND approximation of the wall friction is 12 times more accurate than the ZZ prediction. The improvement in the FiND derivative when compared with the ZZ derivative is also apparent in Figure 17, illustrating the local wall shear stress. The errors in the FiND approximation are slightly higher near the extremities of the boundary. This is caused by the rapid adjustment of the solution from the imposed boundary conditions to a divergence-free velocity field satisfying the momentum equations.

Another use of the approximated derivatives is for error estimation. In practical problems, the exact solution $\mathbf{u}_{\mathrm{ex}}$ is unknown, so that the norm of the true error $\|\mathbf{u}_h - \mathbf{u}_{\mathrm{ex}}\|_{H_1}$ cannot be evaluated. However, the high accuracy of the FiND derivatives makes them an excellent surrogate for $\mathbf{u}_{\mathrm{ex}}$, so that the true error may be approximated as follows:

$$\|\mathbf{u}_h - \mathbf{u}_{\mathrm{ex}}\|_{H_1} \approx \|\mathbf{u}_h - \mathbf{u}_{\mathrm{FiND}}\|_{H_1} \tag{69}$$

Figure 18 compares the two error estimators $\|\mathbf{u}_h - \mathbf{u}_{\mathrm{FiND}}\|_{H_1}$ and $\|\mathbf{u}_h - \mathbf{u}_{\mathrm{ZZ}}\|_{H_1}$ with the true error. As can be seen, the accuracy of both error estimators improves with mesh refinement, but the FiND error is much closer to the exact error than the ZZ approximation. Results are presented in

DOI: 10.1002/nme

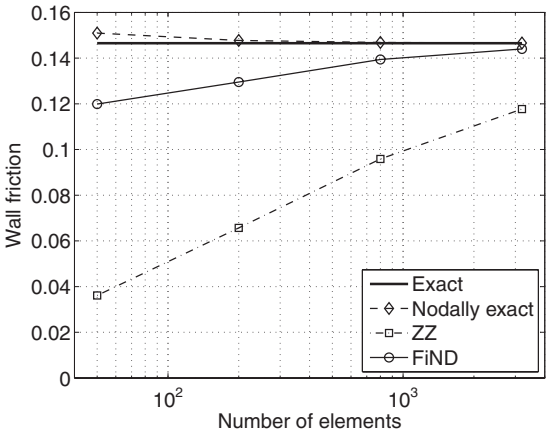

Figure 16. Wall friction estimate using the ZZ and FiND derivatives.

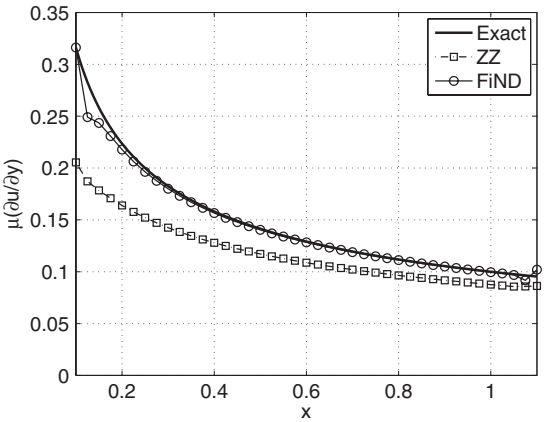

Figure 17. Wall shear stress from the ZZ and FiND derivatives.

Figure 19 in the form of an efficiency index (ratio of estimator to true error):

$$\theta_{\mathrm{FiND}} = \frac{\|\mathbf{u}_h - \mathbf{u}_{\mathrm{FiND}}\|_{H_1}}{\|\mathbf{u}_h - \mathbf{u}_{\mathrm{ex}}\|_{H_1}} \tag{70}$$

$$\theta_{\mathrm{ZZ}} = \frac{\|\mathbf{u}_h - \mathbf{u}_{\mathrm{ZZ}}\|_{H_1}}{\|\mathbf{u}_h - \mathbf{u}_{\mathrm{ex}}\|_{H_1}} \tag{71}$$

The closer to one the efficiency index is, the more reliable and accurate the estimator will be. Both estimators tend to 1 as mesh is refined, indicating that both estimators are asymptotically exact. This is important in practice as it means that the accuracy of both the solution and its error estimate improves with mesh refinement. As θ goes to 1, the estimator becomes more and more

DOI: 10.1002/nme

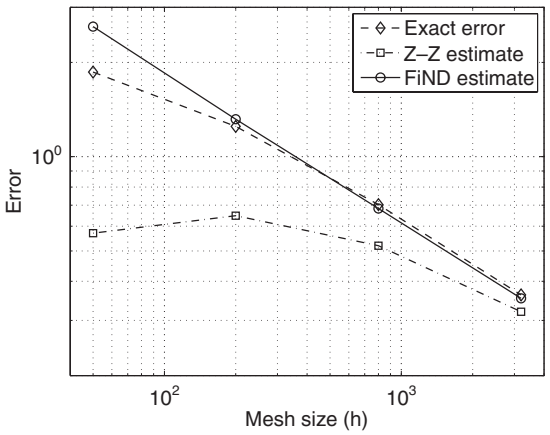

Figure 18. True and estimated errors of the solution.

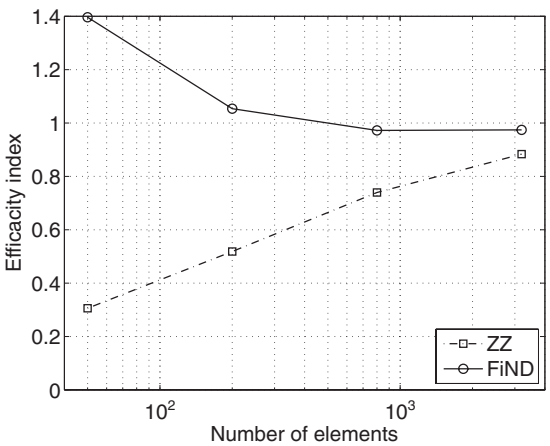

Figure 19. Global efficiency index of the ZZ and FiND error estimators.

accurate. Figure 19 indicates that the FiND estimator achieves asymptotic exactness on coarser meshes than its ZZ counterpart.

The boundary layer problem was also solved on non-uniform unstructured meshes in order to verify the behavior of the FiND method on such grids. The initial mesh having 102 elements is shown in Figure 20. Uniform grid refinement was performed to confirm the convergence rate of the recovered derivatives. The errors of the reconstructed derivatives are compared with the error of the FE solution derivatives in Figure 21. The performances of the ZZ and FiND methods for internal and boundary nodes are compared in Figure 22. Only the y derivative is considered in Figure 22 and boundary nodes include only points located on the bottom wall ($y=0$). We observe that the FiND derivatives are always more accurate than the ZZ derivatives and that the global rate of convergence is higher for the FiND derivatives when compared with the ZZ approximation.

DOI: 10.1002/nme

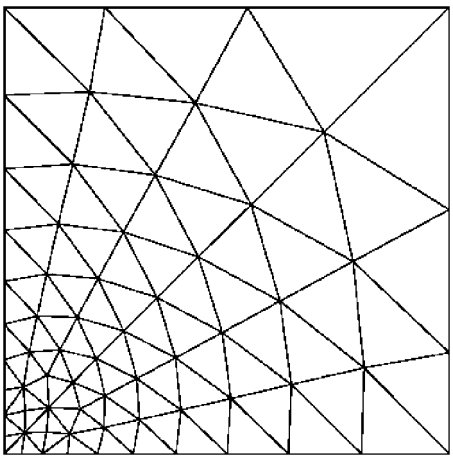

Figure 20. Initial non-uniform mesh for the boundary layer problem.

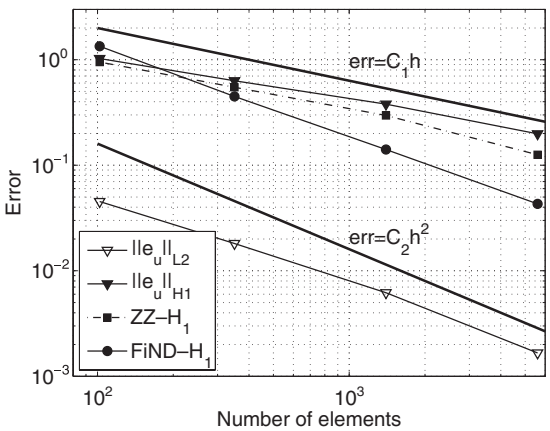

Figure 21. Errors of the finite element solution, ZZ and FiND derivatives on non-uniform meshes.

The convergence rate for internal nodes is quadratic, but it becomes linear for boundary nodes. However, the accuracy of the FiND derivatives at boundary nodes is much higher than that of the ZZ derivatives. These observations are similar with those made by Ilinca and Pelletier [1] when applying FiND to scalar convection–diffusion equations.

6. CONCLUSIONS

A new method, the finite node displacement (FiND) method, for computing highly accurate nodal derivatives of the Navier–Stokes solution is presented. The FiND approach is very cost effective. It uses an imposed finite displacement of mesh nodes and solves small local problems on the patch

DOI: 10.1002/nme

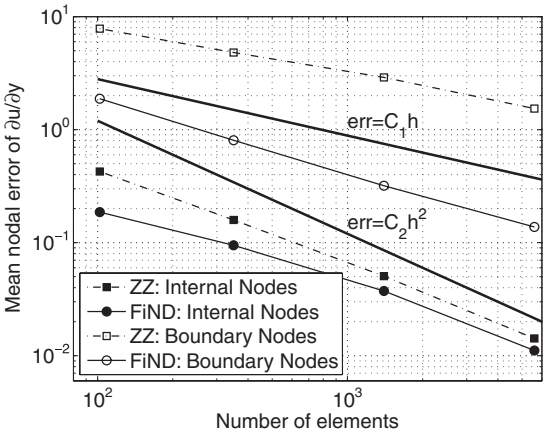

Figure 22. Nodal errors of the ZZ and FiND derivatives on non-uniform meshes.

of elements surrounding the displaced point, for the nodal values of the velocity at the perturbed locations. For internal nodes, the method provides accuracy that is as good as or better than that of the ZZ SPR technique. A variant of the method suited for boundary nodes is developed. Nodal values of velocity normal derivatives at the boundary are computed by inserting an additional mesh point inside the domain at a small distance from the boundary point in the boundary normal direction. The FiND method yields normal derivatives at boundary points that are a consistent representation of the auxiliary fluxes. These boundary derivatives are more accurate than those provided by the superconvergent patch recovery technique. Results show that the error estimator using the FiND derivatives is more accurate than the original ZZ estimator.

REFERENCES

1. Ilinca F, Pelletier D. Computation of accurate nodal derivatives of finite element solutions: the finite node displacement method. *International Journal for Numerical Methods in Engineering* 2007; **71**(10):1181–1207.
2. Zienkiewicz OC, Taylor RL, Zhu JZ. *The Finite Element Method*, *its Basis & Fundamentals*, vol. 1 (6th edn). Elsevier: Amsterdam, 2005.
3. Barlow J. Optimal stress location in finite element method. *International Journal for Numerical Methods in Engineering* 1976; **10**:243–251.
4. Zlamal M. Superconvergence and reduced integration in the finite element method. *Mathematics of Computations* 1974; **32**:663–685.
5. Babuska I, Miller A. The post-processing approach in the finite element method—Part 1: calculation of displacements, stresses and other higher derivatives of the displacements. *International Journal for Numerical Methods in Engineering* 1984; **20**:1085–1109.
6. Zienkiewicz OC, Zhu JZ. A simple error estimator and adaptive procedure for practical engineering analysis. *International Journal for Numerical Methods in Engineering* 1987; **24**:337–357.
7. Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery and *a posteriori* error estimates. Part 1: the recovery technique. *International Journal for Numerical Methods in Engineering* 1992; **33**:1331–1364.
8. Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery and *a posteriori* error estimates. Part 2: error estimates and adaptivity. *International Journal for Numerical Methods in Engineering* 1992; **33**:1365–1382.
9. Zienkiewicz OC, Zhu JZ. The superconvergent patch recovery (SPR) and adaptive mesh refinement. *Computer Methods in Applied Mechanics and Engineering* 1992; **101**(1–3):207–224.
10. Carstensen C, Funken SA. Averaging techniques for FE—a posteriori error control in elasticity. Part I: conforming FEM. *Computer Methods in Applied Mechanics and Engineering* 2001; **190**:2483–2498.

DOI: 10.1002/nme

11. Ainsworth M, Craig A. A posteriori error estimators in the finite element method. *Numerische Mathematik* 1992; **60**:429–463.
12. Pelletier D, Trepanier J-Y. Implementation of error analysis and norms to computational fluid dynamics applications. Report 1: bilinear finite elements. *CDT Project p2223 Report 1*, École Polytechnique de Montréal, July 1997.
13. Pelletier D, Trepanier J-Y. Implementation of error analysis and norms to computational fluid dynamics applications. Report 2: biquadratic and bilinear finite elements. *CDT Project p2223 Report 2b*, École Polytechnique de Montréal, July 1997.
14. Boroomand B, Zienkiewicz OC. Recovery by equilibrium patches (REP). *International Journal for Numerical Methods in Engineering* 1997; **40**:137–164.
15. Wiberg N-E, Abdulwahab F, Ziukas S. Enhanced superconvergent patch recovery incorporating equilibrium and boundary conditions. *International Journal for Numerical Methods in Engineering* 1994; **37**:3417–3440.
16. Wiberg N-E, Li XD. Superconvergent patch recovery of finite element solution and L_2 norm error estimate. *Communications in Numerical Methods in Engineering* 1994; **10**:313–320.
17. Blacker T, Belytschko T. Superconvergent patch recovery with equilibrium and conjoint interpolant enhancements. *International Journal for Numerical Methods in Engineering* 1994; **37**:517–536.
18. Kvamsdal T, Okstad KM. Error estimation based on superconvergent patch recovery using statically admissible stress yields. *International Journal for Numerical Methods in Engineering* 1998; **42**:443–472.
19. Pelletier D, Ilinca F, Hétu J-F. Adaptive finite element method for convective heat transfer with variable fluid properties. *AIAA Journal of Thermophysics and Heat Transfer* 1994; **8**:687–694.
20. Pelletier D, Ilinca F. An adaptive finite element method for mixed convection. *AIAA Journal of Thermophysics and Heat Transfer* 1995; **9**(4):708–714.
21. Pelletier D, Ignat L, Ilinca F. Adaptive finite element method for conjugate heat transfer. *Numerical Heat Transfer—Part A* 1997; **32**(3):267–289.
22. Pelletier D, Ilinca F, Turgeon É. An adaptive finite element method for forced convection. *International Journal for Numerical Methods in Fluids* 1997; **25**(7):803–823.
23. Pelletier D, Ilinca F. Adaptive remeshing for the k–ε model of turbulence. *AIAA Journal* 1997; **35**(4):640–646.
24. Ilinca F, Pelletier D, Arnoux-Guisse F. An adaptive finite element scheme for turbulent free shear flows. *International Journal of Computational Fluid Dynamics* 1997; **8**:171–188.
25. Ilinca F, Pelletier D, Garon A. An adaptive finite element method for a two-equation turbulence model in wall-bounded flows. *International Journal for Numerical Methods in Fluids* 1997; **24**:101–120.
26. Ignat L, Pelletier D, Ilinca F. Adaptive computation of turbulent forced convection. *Numerical Heat Transfer, Part A* 1998; **34**:847–871.
27. Ilinca F, Pelletier D, Ignat L. Adaptive finite element solution of compressible turbulent flows. *AIAA Journal* 1998; **36**:2187–2194.
28. Turgeon E, Pelletier D, Ilinca F. Compressible heat transfer computations by an adaptive finite element method. *International Journal for Thermal Sciences* 2002; **41**:721–736.
29. Turgeon E, Pelletier D. Verification and validation in CFD using an adaptive finite element method. *Canadian Aeronautics and Space Journal* 2002; **48**:219–231.
30. Pelletier D, Turgeon E, Lacasse D, Borggaard J. Adaptivity, sensitivity and uncertainty: towards standards of good practice in computational fluid dynamics. *AIAA Journal* 2003; **41**:1925–1933.
31. Pelletier D, Turgeon E, Tremblay D. Verification and validation of impinging jet simulations using an adaptive FEM. *International Journal for Numerical Methods in Fluids* 2004; **44**:737–763.
32. Duvigneau R, Pelletier D. On accurate boundary conditions for a shape sensitivity equation method. *International Journal for Numerical Methods in Fluids* 2006; **50**:147–164.
33. Duvigneau R, Pelletier D, Borggaard J. An improved continuous sensitivity equation method for optimal shape design in mixed convection. *Numerical Heat Transfer, Part B* 2006; **50**(1):1–24.
34. Duvigneau R, Pelletier D. A sensitivity equation method for fast evaluation of nearby flows and uncertainty analysis for shape parameters. *International Journal of Computational Fluid Dynamics* 2006; **20**(7):497–512.
35. Carey GF. Derivative calculation from finite element solutions. *Computer Methods in Applied Mechanics and Engineering* 1982; **35**:1–14.
36. Carey GF, Chow SS, Seage MK. Approximate boundary flux calculations. *Computer Methods in Applied Mechanics and Engineering* 1985; **50**:107–120.
37. Wheeler MF. A Galerkin procedure for estimating the flux for two-point boundary-value problems. *SIAM Journal on Numerical Analysis* 1974; **11**:764–768.

DOI: 10.1002/nme

38. Hughes TJR, Engel G, Mazzei L, Larson MG. The continuous Galerkin method is locally conservative. *Journal of Computational Physics* 2000; **163**:467–488.
39. Kvamsdal T. Variationally consistent postprocessing. In *Proceedings for Computational Mechanics*, *New Trends and Applications*, Idelsohn S, Oñate E, Dvorkin E (eds), CIMNE, Barcelona, Spain, 1998.
40. Melbo H, Kvamsdal T. Goal oriented error estimators for Stokes equations based on variationally consistent postprocessing. *Computer Methods in Applied Mechanics and Engineering* 2003; **192**:613–633.
41. Tezduyar TE, Shih R, Mittal S, Ray SE. Incompressible flow using stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Research Report UMSI 90/165*, University of Minnesota/Supercomputer Institute, Minneapolis, 1990.
42. Franca LP, Frey SL. Stabilized finite element methods. II. The incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**(2–3):209–233.
43. Pehlivanov AI, Lazarov RD, Carey GF, Chow SS. Superconvergence analysis of approximate boundary-flux calculations. *Numerische Mathematik* 1992; **63**:483–501.

DOI: 10.1002/nme