A NODE-CENTERED PRESSURE-BASED METHOD FOR ALL SPEED FLOWS ON UNSTRUCTURED GRIDS

Hyung-Il Choi
Mechanical Engineering and Technology Research Institute, Hanyang University, Seoul, Korea

Dohyung Lee
Department of Mechanical Engineering, Hanyang University, Ansan, Korea

Joo-Sung Maeng
Department of Mechanical Engineering, Hanyang University, Seoul, Korea

This article proposes a node-centered pressure-based method for predicting flows at all speeds on unstructured grids. The compressible version of the SIMPLE algorithm is extended to unstructured grids. For memory and computing time efficiency of arbitrary cell topologies, a node-centered scheme with edge-based data structure is applied to the segregated unstructured-grid algorithm. Convection terms are discretized using the second-order scheme with a deferred-correction approach. Diffusion-term discretization is based on the structured-grid analogy, which can be easily adapted to hybrid unstructured grid solvers. By using the proposed approach, three test cases in the continuum regime are computed. The demonstration of this method is extended to a slip flow problem that has low Reynolds number and compressibility effect. Predictions are compared to other results, which show that the proposed method can improve efficiency in memory usage and computing time without losing any accuracy even in mixed-element grids.

1. INTRODUCTION

Numerical algorithms for simulating fluid flows are categorized as either density-based or pressure-based approaches. The former have been used mostly for computing high-speed compressible flows and the latter for low-speed incompressible flows. However, for resolving convergence problems in the low-speed region incompressible flows or analyzing very-low-speed flow with compressibility effect, an all-speed algorithm that can cover the arbitrary speed flow regime is necessary. Therefore, much research on this theme has been performed by using either the density-based approach or the pressure-based approach.
First, the density-based method [1–3] is amenable for high-speed compressible flows. If this method is used for low-speed incompressible flow problems, it will lose the advantages of the hyperbolic system, because the coupling of density–velocity–pressure deteriorates significantly; that is, the linear system of governing equations is very stiff. As a result, the convergence problem becomes critical. To reduce this difficulty, the artificial compressibility algorithm was developed [4], but this requires the work of finding an optimum artificial compressibility factor [5, 6]. Several researchers have focused on the local preconditioning approach [7, 8], which can improve convergence characteristics, and use the time-marching algorithm.

On the other hand, pressure-based algorithms solve the pressure field instead of the density field. Many workers have developed and utilized various pressure-based methods for calculating incompressible flows on either structured grids [9–11] or unstructured grids [12, 13]. In contrast to the incompressible pressure-based method, most works on the extended pressure-based method for compressible flows have been performed on structured grids [14–16]. These extended formulations can also be used in incompressible flow problems. Karki and Patankar [14] extended the SIMPLER algorithm to a compressible version. They used the power-law scheme for the con-
vection-diffusion discretization and a staggered grid arrangement. The results were well in agreement with other results, but shock smearing was not eliminated. Some years later, Demirdzic et al. [15] proposed an extended SIMPLE method. They used the nonstaggered grid arrangement and second-order central differencing with a deferred-correction approach. The results showed good accuracy and well-resolved shocks. Rincon and Elder [16] discretized convection terms by a high-order scheme with a limiter, the so-called normalized variable formulation (NVF), in order to capture the shocks, and they obtained accurate results in some compressible-flow problems. When the aforementioned structured-grid methodologies are used on complex geometries, generating good structured grids becomes the non-negligible burden of the user. The multiblock approach is one alternative, but this also has a few drawbacks. One is the conservation problem of the flux calculations through the block interfaces. Another is the degenerate convergence rate caused by the interpolation at the block interfaces in an iterative manner. By contrast, the unstructured-grid pressure-based method can become a viable and efficient alternative in complex geometries. Lien [17] recently developed an unstructured-grid pressure-based method for all speed flow regimes. He used the cell-centered scheme and cell-based data structure. However, there is still room for improvement in memory requirements and computing time efficiency in the above cell-data strategy. This is the motivation for the present research.

The aim of the present work is to devise an efficient pressure-based method for all speed flows on unstructured grids with arbitrary cell topologies, i.e., mixed elements. We apply the node-centered scheme with edge-based data structure [18, 19] to an unstructured-grid pressure-based algorithm. The memory requirement and the computing efficiency of this cell-data strategy are elaborately investigated. Then a discretization process is proposed, which is appropriate for hybrid unstructured grids. For the pressure-correction step, the extended SIMPLE algorithm is employed. The momentum interpolation by Rhie and Chow [20] is applied to avoid checkerboard oscillation in the collocated grid arrangement. To obtain linear solution profiles in each control volume, the unweighted least-square reconstruction [19] is used. Noble space-centered interpolation is implemented in a deferred manner for the convection-term discretization. Diffusion-term discretization is based on a structured-grid analogy [21] that can be very flexible for hybrid grids. By using the proposed approach, three validation cases of incompressible and compressible flows in the continuum flow regime are demonstrated. Moreover, a microchannel flow problem [22–26] in the slip flow regime is computed that has a very low Reynolds number and compressibility effect. Then, memory requirements and computing efficiencies of the three examples are examined.

In this article, the system of the governing equations is described in the next section. Section 3 explains the current finite-volume discretization as well as the pressure-correction procedure. The results of continuum and slip flow simulations are described in Section 4, and the concluding remarks are in the next.

2. GOVERNING EQUATIONS

The dimensionless integral forms of two-dimensional steady-state conservation equations for mass, momentum, and energy are given by
\[ \int_S \mathbf{F} \cdot \mathbf{n} \, dS - \int_S \mathbf{G} \cdot \mathbf{n} \, dS = 0 \]  

where \( \mathbf{F} \) is the inviscid flux vector and \( \mathbf{G} \) is the viscous flux vector. Each of them can be rewritten in Cartesian coordinates as follows:

\[
\mathbf{F} = f \mathbf{i} + g \mathbf{j} \quad \mathbf{G} = \frac{1}{Re} (f \mathbf{i} + g \mathbf{j})
\]

where

\[
\begin{align*}
    f &= \begin{pmatrix}
        \rho u \\
        \rho u^2 + p \\
        \rho uv \\
        \rho uh_0
    \end{pmatrix},
    g &= \begin{pmatrix}
        \rho v \\
        \rho v^2 + p \\
        \rho uv \\
        \rho vh_0
    \end{pmatrix},
    f_v &= \begin{pmatrix}
        0 \\
        \tau_{xx} \\
        \tau_{xy} \\
        \tau_x - q_x
    \end{pmatrix},
    g_v &= \begin{pmatrix}
        0 \\
        \tau_{xy} \\
        \tau_{yy} \\
        \tau_y - q_y
    \end{pmatrix}
\end{align*}
\]

\[
\begin{align*}
    \tau_{xx} &= \mu \left( 2u_x - \frac{2}{3} \nabla \cdot \mathbf{V} \right), \\
    \tau_{yy} &= \mu \left( 2v_y - \frac{2}{3} \nabla \cdot \mathbf{V} \right), \\
    \tau_{xy} &= \mu (u_x + u_y).
\end{align*}
\]

In the above equation set, \( \rho \) is the density of the fluid and \( p \) is the static pressure; \( u \) and \( v \) are Cartesian velocity components in the \( x \) and \( y \) directions, respectively; \( h_0 \) is the total enthalpy; \( \mu \) is the dimensionless viscosity; \( Re \) represents the Reynolds number; \( \tau \) is the viscous dissipation; and \( q \) is the heat diffusion. Equation (1c) expresses the components of viscous stresses, viscous dissipation, and heat conduction, respectively. If the pressure-correction approach is adopted, the first row of the equation set (1), the mass conservation equation, will be transformed to the equation for pressure correction. Provided the perfect gas law is satisfied, the relation for the closure is

\[ p = \rho RT \]  

3. FINITE-VOLUME PROCEDURES

3.1. Generic Transport Equation

In a segregated pressure-based approach, a system of equations, Eq. (1), is solved sequentially. The momentum and energy conservation equations each can be rearranged in canonical form as

\[ \int_S \rho \mathbf{V} \phi \cdot \mathbf{n} \, dS - \int_S \lambda_{\phi} \nabla \phi \cdot \mathbf{n} \, dS = S_{\phi} \]  

where \( \phi \) stands for \( u, v, h_0 \). \( \lambda_{\phi} \) represents the diffusion coefficient for each equation.
The first term on the left-hand side in Eq. (3) describes convection effect, and the second the diffusion effect. The right-hand side is treated as the source term. In the momentum equations, this term contains gradient terms for velocities, a pressure force term, and another body force term. The continuity equation can be also recast into the form of Eq. (3) without diffusion. It will be used to derive an equation for pressure correction.

### 3.2. Domain Discretization and Cell Strategy

The solution domain is subdivided into a finite number of nonoverlapped arbitrary control volumes in unstructured-grid methodology. The methods for constructing control volumes can be classified into cell-centered schemes and node-centered schemes. The node-centered scheme is flexible in integrating control volumes regardless of element shape. When the grids are highly skewed, the flux calculations at the boundary grids in the node centered scheme are more accurate in comparison to those in the cell-centered scheme [27]. The node-centered scheme can impose the Dirichlet boundary condition directly, while the cell-centered scheme requires special treatment. The disadvantage of the node-centered scheme is that the local truncation error can be slightly larger than that of the cell-centered scheme [27].

To estimate the memory requirements, the number of control volumes may be considered, since it becomes a dominant factor. The node-centered scheme has considerably fewer control volumes compared to the cell-centered one. For example, if we assume that all elements are triangular and boundary effects are neglected, there are twice as many control volumes for the cell-centered scheme as many for the node-centered scheme. This means that the memory requirement of the real variable for the cell-centered scheme is twice that of the node-centered one. In three-dimensional cases, provided that all grids are composed of well-generated tetrahedral elements, the number of primary grids (elements) is about six times as much as the number of nodes. Therefore, the node-centered scheme is much more economical in three-dimensional cases.

For more elaborate consideration of a segregated algorithm, the governing parameter for estimating the amount of memory requirement can be derived from the total number of dependent variables and the number of nonzero elements of the linearized sparse matrix system. For a two-dimensional case, Table 1 represents a comparison between the memory requirement for the cell-centered scheme and that of the node-centered scheme. In Table 1, \( a_1 \) is the number of faces in each primary grid and \( a_2 \) is the number of neighboring vertices connected to each vertex. \( N \) represents number. For example, one case problem is evaluated as follows. All primary grids are well-generated triangular shapes and the boundary effects are neglected. The number of flow variables is three. For this case, \( a_1, a_2, N(\text{variable}), \) and \( N(\text{element}) \) are 3, 6, 3, \( 2 \times N(\text{vertex}) \), respectively. As a result, the memory

<table>
<thead>
<tr>
<th></th>
<th>Cell-centered scheme</th>
<th>Node-centered scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse matrix</td>
<td>((a_1 + 1) \times N(\text{element}))</td>
<td>((a_2 + 1) \times N(\text{vertex}))</td>
</tr>
<tr>
<td>Variables</td>
<td>(N(\text{variable}) \times N(\text{element}))</td>
<td>(N(\text{variable}) \times N(\text{vertex}))</td>
</tr>
</tbody>
</table>
requirement for the cell-centered scheme is $14 \times N(\text{vertex})$ and that for the node-centered scheme is $10 \times N(\text{vertex})$. This estimation demonstrates that the node-centered scheme requires less memory. As the number of flow variables increases, the memory savings increase significantly. Therefore, the present study utilizes the node-centered scheme.

Figure 1a shows the control volume around the vertex $o$ in the node-centered scheme. Vertices $o$ and $j$ are two points that specify one edge. The cell boundaries are constructed as median dual type [18, 19]. The line segment, composed of the centroid of one adjacent primary grid and the median point of the edge, becomes one fraction of the cell boundary. Then, all these line segments construct the control surface boundaries of each control volume. These control surfaces are shown as dashed lines in Figure 1a. To calculate the flux through the control surface, one-point Gaussian quadrature is used at each corresponding edge. Figure 1b represents one median dual-cell surface. Fluxes through this surface can be calculated by using variables at the median point of the edge. The representative surface normal vector used in one-point quadrature can be easily derived by the surface area averaging between two vector segments, i.e., $S_L$ and $S_R$. Then, every flux calculation at each edge comprises all flux calculations on the solution domain. This is similar to the convective flux calculation of the popular explicit density-based approach [18]. This type of procedure is known as a very efficient way provided that the order of spatial accuracy of the numerical scheme is within second-order [18]. Therefore, the combination of node-centered scheme and edgewise flux calculation can considerably reduce memory usage and computing time in a segregated solution algorithm.

### 3.3. Reconstruction

In the node-centered finite-volume discretization, the variables of vertex $o$ are representative values of that control volume. If first-order spatial accuracy is assumed, variables inside the cell are constant. In order to increase the spatial accuracy, a reconstruction process is needed [18, 19, 21]. In the present algorithm, the solution distribution is considered as piecewise linear for its simplicity, and an unweighted least-square reconstruction method is employed [19]. This gives second-order spatial accuracy.

The variables of neighboring nodes can be expressed to the value of vertex $o$ and its gradients as follows (Figure 1c):

$$
\phi_j = \phi_o + \phi_{xo}(x_j - x_o) + \phi_{yo}(y_j - y_o)
$$

where $\phi_{xo}$ and $\phi_{yo}$ are Cartesian components of the solution gradient in control volume $o$. Rewriting this with all neighboring nodes, the matrix form is

$$
\begin{bmatrix}
\Delta x_{o1} & \Delta y_{o1} \\
\Delta x_{o2} & \Delta y_{o2} \\
\vdots & \vdots \\
\Delta x_{oN} & \Delta y_{oN}
\end{bmatrix}
\begin{bmatrix}
\phi_{xo} \\
\phi_{yo}
\end{bmatrix}
=
\begin{bmatrix}
\phi_{j1} - \phi_o \\
\phi_{j2} - \phi_o \\
\vdots \\
\phi_{jN} - \phi_o
\end{bmatrix}
$$

Thus, the matrix equation is solved for the unknown variables at the cell boundaries. This type of procedure is consistent with the convective flux calculation of the popular explicit density-based approach [18]. Therefore, the combination of node-centered scheme and edgewise flux calculation can considerably reduce memory usage and computing time in a segregated solution algorithm.
Figure 1. Representation of the proposed node-centered scheme: (a) node-centered control volume; (b) median dual-cell representation; (c) nodes for the least-square reconstruction of variables.
where $\Delta x$ and $\Delta y$ are Cartesian components of the distance vector from vertex $o$ to $j$. This is the overdetermined linear system of equations, i.e., $Ax = b$. To solve Eq. (5), we adopt the modified Gram–Schmidt orthogonalization, which is known to be a numerically efficient procedure [28].

$$A = QR$$

where $Q$ is the orthogonal matrix and $R$ is the upper triangular matrix. Therefore, the solution gradient can be obtained from the following form:

$$x = R^{-1}Q^Tb$$

### 3.4. Discretization of Convection Term

The convection term is represented as the sum of convective fluxes through the cell surfaces of each control volume. Each convective flux at one cell surface can be approximated as the product of the mass flux multiplied by the dependent variable. This is given by

$$\int_S \rho \nabla \phi \cdot n \, dS \approx \sum_{sj} F_{sj} \phi_{sj}$$

The first-order upwind scheme is stable but is not satisfactory due to its false diffusion and low-order spatial accuracy. To increase the accuracy, second-order space-centered interpolation is used that is similar to Muzafjerija’s [21] cell-centered approach. The cell face value is approximated as the averaged value of the left and right cell face values. The left and right cell face values are reconstructed from the vertex data. This interpolation can be written as

$$F_{sj} \phi_{sj} = F_{sj} \cdot \frac{1}{2} \left\{ \left[ \phi_o + \nabla \phi_o(x_{sj} - x_o) \right] + \left[ \phi_j + \nabla \phi_j(x_{sj} - x_j) \right] \right\}$$

To obtain a stable convergence property, Eq. (9) is implemented by using the deferred correction and the blending factor [21]. In this study, the blending factor, $\gamma$, is chosen to be from around 0.9 to 1.0, which means the convective scheme uses weak upwinding. This convective scheme is simple but gives both reasonable accuracy and geometric flexibility even in mixed-element grids.

### 3.5. Discretization of Diffusion Term

The diffusion term is approximated as the sum of all diffusion fluxes through the cell surfaces of each control volume in the following form:

$$\int_S \lambda \nabla \phi \cdot n \, dS \approx \sum_{sj} (\lambda \nabla \phi^{est} \cdot S)_{sj}$$

The diffusion term discretization has been performed by several researchers using various methods. Anderson et al. [19] used the Gauss theorem in the primary grids.
adjacent to the edge. The Galerkin finite-element formulation has been widely used in the density-based approach [18]. However, these are not easy to implement in the unstructured-grid implicit solver. In particular, applying these methods to edge-based data structure is very complicated. The present procedures use the modified structured-grid analogy [13, 21]. This is easy to implement for the implicit solver using arbitrary grid topologies.

The conservation equations have the rotationally invariant property. Thus, we can split the diffusion flux into two components at each control surface, i.e., the edgewise and normal components, as shown in Figure 2. The edgewise gradient uses central differencing and the normal gradient uses the averaged gradient of two control volumes adjacent to the edge. This is given by the formula

$$\nabla \phi_{sj}^{ext} = \left[ \nabla \phi_{sj} - (\nabla \phi_{sj} \cdot \mathbf{n}_d) \cdot \mathbf{n}_d \right] + \left( \frac{\phi_j - \phi_o}{d_{oj}} \cdot \mathbf{n}_j \right)$$

where $\mathbf{n}_d$ represents the unit vector tangential to the edge and $d_{oj}$ is the length of the edge. $\nabla \phi_{sj}$ is an arithmetic averaging of gradients between left and right cells. Substituting Eq. (11) into Eq. (10), we can approximate diffusive fluxes at all control surfaces. This formulation does not break the diagonal dominance of the discretized linear system of equations.

### 3.6. Pressure-Correction Algorithm

In the segregated pressure-based algorithm, SIMPLE and its variants have been widely used for pressure–velocity coupling of the Navier–Stokes equations. These methods are successfully extended to treat the compressibility. Karki and Patankar [14] and Demirdzic et al. [15] presented solution methods based on the SIMPLE-type procedure. Lien [17] extended this to an unstructured-grid methodology. The present study applies extended SIMPLE to the node-centered framework.

In the extended SIMPLE method, the corrected mass flux through the control-volume face $sj$ can be expressed as [15, 16]

$$F_{sj} = (\rho^p + \rho')_{sj}(\mathbf{V}^* + \mathbf{V}')_{sj} \cdot \mathbf{S}_{sj}$$

![Figure 2. Splitting the gradient on the control surface boundary.](image)
or

$$F_{sj} = (\rho^n \mathbf{V} \cdot \mathbf{S} + \rho^n \mathbf{V}' \cdot \mathbf{S} + \rho' \mathbf{V}^* \cdot \mathbf{S} + \rho' \mathbf{V}' \cdot \mathbf{S})_{sj}$$

(13)

where the superscript $n$ represents the value of the previous iteration. The superscript $'$ means the correction component. The superscript $^*$ means the variable that satisfies the momentum conservation equation only. Then, the correction part of the mass flux is given by

$$F_{sj}' = (\rho^n \mathbf{V}' \cdot \mathbf{V} + \rho' \mathbf{V}^* \cdot \mathbf{S} + \rho' \mathbf{V}' \cdot \mathbf{S})_{sj}$$

(14)

The last term in Eq. (14) can be neglected since it vanishes faster than the other terms in the iterative steady-state solution process. The remaining two terms of the right-hand side are related to the velocity correction and the density correction, respectively. The first term on the right-hand side can be approximated based on the original SIMPLE method [13, 21].

$$(\rho^n \mathbf{V}' \cdot \mathbf{S})_{sj} = -((\rho^n |\mathbf{S}|)_{sj} \frac{\Omega_o}{A_o})_{sj} (\frac{\delta p'}{\delta n})_{sj}$$

(15)

where $\Omega_o$ is the area of each control volume and $A_o$ represents the diagonal element of the discretized momentum equation corresponding to the vertex $o$. In the second term on the right-hand side in Eq. (14), the density correction can be reexpressed by the pressure correction as

$$\rho' \approx \left(\frac{\partial p}{\partial p}\right)p' = C_\rho p'$$

(16)

Then, substituting Eq. (15) and (16) into Eq. (14) and rearranging, we obtain the approximate form of mass flux correction at one control-volume face given by

$$F_{sj}' = -((\rho^n |\mathbf{S}|)_{sj} \frac{\Omega_o}{A_o})_{sj} (\frac{\delta p'}{\delta n})_{sj} + (C_\rho \mathbf{V}^* \cdot \mathbf{S})_{sj} p'_{sj}$$

(17)

From Eq. (17), the first term and the second term on the right-hand side are diffusion-like and convection-like terms, respectively. Thus, the discretization schemes described in Sections 3.4 and 3.5 are applied. Once pressure corrections at all control-surface boundaries are approximated and summed, we obtain the linear system of equations for pressure correction. After solving the linear system, the pressure, density, and velocity can be corrected by

$$p_o = p^n_o + \beta p'_o$$

(18)

$$\rho_o = \rho^n_o + \left(\frac{\partial p}{\partial p}\right)_o \beta p'_o$$

(19)

$$\mathbf{V}_o = \mathbf{V}^*_o + \mathbf{V}'_o = \mathbf{V}^*_o - \frac{\Omega_o}{A_o} \nabla p'_o$$

(20)

where $\beta$ is the pressure relaxation factor and has a value from around 0.2 to 0.4.
3.7. Boundary Conditions

The boundary conditions used in the present study and the treatment necessary to incorporate them into the discretized equations are similar to the method of Muzaferija [21]. The inlet and outlet boundaries are treated as Dirichlet-type or Neumann-type boundary conditions, depending on the problems that will be solved. The wall boundary condition is treated as two types. In the continuum flow regime, the velocity adjacent to the wall is the wall velocity. That is no-slip boundary condition. However, for slip flow problems, the slip wall condition should be imposed. The well-known Maxwell slip wall condition [23, 25] is applied in the present research. This condition is based on the approximation of surface–gas molecular interaction in the gas kinetic theory. Assuming isothermal flow, the slip condition can be written as

\[
U_g = \sigma \text{Kn} \frac{dU}{dy}
\]  

where subscript \( g \) represents the gas at the wall. \( \sigma \) is the streamwise accommodation coefficient. From Eq. (21), the slip velocity of the wall depends on the streamwise accommodation coefficient, the Knudsen number, and the streamwise velocity gradient. In the present algorithm, the streamwise velocity gradient is reproduced directly from the reconstructed boundary cell gradient. For stable convergence property in the iteration process, relaxation is adopted as

\[
U_{g,\text{new}} = (1 - \omega) U_{g,\text{old}} + \omega U_{g,\text{est}}
\]  

where the relaxation factor, \( \omega \), is chosen to be from around 0.8 to 1.0.

3.8. Data Structure

Since the grid connectivity cannot be expressed implicitly in the unstructured-grid methodology, explicit pointers are needed. The data structure with explicit pointers influences memory usage and computing efficiency. In the present study, the edge-based data structure is applied that has been widely used for density-based computational fluid dynamics (CFD) algorithms [18, 19]. The relation between the discretized linear system and edge data structure in the present segregated solution procedure is shown in Figure 3. As the flux is calculated at each edge, two diagonal elements and two off-diagonal elements of the linearized system are updated. These corresponding elements are expressed as diamond shapes in Figure 3. This process is very efficient because flux calculations are not repeated in the entire solution domain. For an efficient implicit-type solver, compressed sparse row (CSR) matrix format [29] is applied and combined with edge data structure. This combination requires a little additional memory of integer type.

3.9. Overall Solution Procedure

By using the proposed discretization procedure, we can obtain the linearized system of equations for momentum, pressure correction, and energy sequentially. Thus, the overall solution procedure is as follows.
0. Initialize all the field variables.
1. Assemble coefficients and sources of the momentum equation using currently available estimates of dependent variables.
2. Solve the linearized momentum equations to obtain velocity components.
3. Assemble the coefficients and sources of the pressure-correction equation.
4. Solve the linearized equation for pressure correction.
5. Correct the pressure, density, velocity, and mass fluxes.
6. Assemble the coefficients and sources of the energy equation.
7. Solve the linearized equation for energy and update other thermodynamic properties such as temperature.
8. If the convergence criterion is not satisfied, return to step (1).

4. RESULTS AND DISCUSSION

In order to validate the unstructured-grid Navier–Stokes program, three test cases in the continuum flow regime are simulated. Furthermore, a microchannel slip flow problem is computed that has a very low Reynolds number and non-negligible compressibility. Next, the efficiency of our approach is exhibited with respect to memory savings and computing time.

4.1. Lid-Driven Cavity Flow Problem

Two-dimensional lid-driven cavity flow is the well-known benchmark problem for Navier–Stokes codes. The rectangular cavity is composed of one moving upper wall with constant velocity and three stationary walls. The unstructured grid used in the present simulation has triangular and quadrilateral elements as shown in Figure 4a. All elements have various aspect ratios and thus it is said that grid quality is not good. Figure 4b represents u-velocity profiles along the vertical line through the geometric center in the case of Re = 1,000. As the number of elements increases, the velocity profile becomes closer to the result of Ghia et al. [30]. The grid with 3,725 elements or more gives excellent agreement. The convergence history of 3,725
elements is shown in Figure 4c. The L2 norm errors of velocity and pressure differences have monotonically decreasing profiles and drop under $10^{-5}$ within 400 iterations. In the second case, grids are composed of 6,400 quadrilateral elements stretched to the wall, and the Reynolds number is 5,000. As shown in Figure 4d, the computed $u$-velocity profile is in good agreement with the results of Ghia et al. [30].
4.2. Incompressible Flow in the T-Shaped Planar Branch

The flow problems of branched geometries have been analyzed by many researchers because of its many application areas, such as biomedical engineering. The two-dimensional T-shaped planar branch is composed of one main branch and one side branch perpendicular to the main one. A schematic of the problem is illustrated in Figure 5a. At the inlet boundary, fully developed velocity profile is imposed and the pressures are extrapolated from the interior domain. The outlet condition is constant atmospheric pressure. A total of 8,100 Cartesian-type quadrilateral grids are generated for the present simulations. Streamlines at $Re = 400$ are illustrated in
Figure 5. Simulations of the flow in the T-shaped planar branch: (a) schematic of the flow split in a branch; (b) streamlines (Re = 400); (c) mass flow fraction at various Reynolds numbers.
4.3. Supersonic Flow over a Circular Arc Bump

The compressible flow over a bump is the well-known compressible flow benchmark problem and is simulated by many authors [15, 16]. The present case is that the ratio of the bump height to channel height is 4%. The grid system has 9,906 triangular elements as shown in Figure 6a. After the grids were generated by the method of Marcum and Weatherill [32], the grid quality improvement procedure, the extended topological clean-up procedure [33], was performed. The free-stream Mach number at the channel inlet is 1.65. All variables at the inlet are prescribed. At the channel exit, all dependent variables such as velocity and pressure are extrapolated from the corresponding interior values. The iso-mach lines are demonstrated in Figure 6b. It is shown that the proposed methodology captures the shock formation and reflections well, without smearing. In Figure 6c, the Mach numbers along the upper and lower boundaries are shown. The present results are in good agreement with the results by Eidelman et al. [34] and give sharper slopes adjacent to the shock. However, slight oscillations are depicted. To reduce these oscillations, local grid refinement or efficient high-resolution convective schemes may be employed [15, 16].
4.4. Slip Flow in the Uniform Microchannel

Microchannel flows in slip flow regime represent the flows in narrow passages of microdevices such as micro shear stress sensors and micro accelerometers. In the present study, two-dimensional, steady, isothermal flow in a uniform microchannel is simulated. The height \( H \) of the channel is 1.04 \( \mu \text{m} \) and the length \( L \) is 31.4 \( \mu \text{m} \). The working fluid is nitrogen. A total of 18,000 quadrilateral grids are generated and stretched to the wall to resolve the large velocity gradient in the vertical direction. The exit Knudsen number is 0.05 and the ratio of inlet to exit pressure is 2.47.

**Figure 6.** Simulation of the supersonic flow over a circular arc bump \( (M = 1.65): \) (a) triangular unstructured grids (9,906 elements); (b) Mach contours; (c) Mach number at upper and lower boundaries.
Maxwell slip wall condition is applied in order to predict the slip phenomena at the wall. In Figure 7, the computed pressure distribution along the channel center is compared to the analytic distribution derived by Arkilic et al. [25] and the Direct Simulation Monte Carlo (DSMC) result simulated by Piekos and Breuer [23]. The present result is slightly closer to the analytical result than the DSMC result is. The pressure nonlinearity owing to the large pressure drop is accurately predicted.

### 4.5. Memory and Computing Efficiencies

Finally, memory requirements for real variables and the number of flux calculations are examined. The present node-centered scheme is compared to the widely used cell-centered scheme in Table 2. The memory requirements are evaluated in Table 1. The number of flux calculations is considered to be that of each linearized transport equation. Three cases of the above simulations are considered.

#### Table 2. Memory requirements and amount of flux calculations for three examples

<table>
<thead>
<tr>
<th></th>
<th>Cell-centered and</th>
<th>Node-centered and</th>
<th>Reduction (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>cell-based</td>
<td>edge-based</td>
<td></td>
</tr>
<tr>
<td>Driven-cavity</td>
<td>Mem. 4,340</td>
<td>Mem. 3,824</td>
<td>11.9</td>
</tr>
<tr>
<td>(mixed grids)</td>
<td>Flux calc. 1,980</td>
<td>Flux calc. 1,030</td>
<td>48.0</td>
</tr>
<tr>
<td>T-shaped branch</td>
<td>Mem. 64,800</td>
<td>Mem. 66,604</td>
<td>-2.8</td>
</tr>
<tr>
<td>(quad. grids)</td>
<td>Flux calc. 32,400</td>
<td>Flux calc. 16,500</td>
<td>49.1</td>
</tr>
<tr>
<td>Flow over a bump</td>
<td>Mem. 79,248</td>
<td>Mem. 55,608</td>
<td>29.8</td>
</tr>
<tr>
<td>(tri. grids)</td>
<td>Flux calc. 29,718</td>
<td>Flux calc. 15,019</td>
<td>49.5</td>
</tr>
</tbody>
</table>
First, in the lid-driven cavity problem with 590 mixed elements, the proposed approach reduces 11.9% of the real memory requirement and 48% of the flux calculations. Second, we consider 8,100 quadrilateral grids of the T-shaped planar branch. The real memory requirement increases by 2.8%. This is due to the fact that the number of vertices is slightly larger than that of elements when the elements are all quadrilateral. The third case is 9,906 triangular grids of flow over a bump problem. The reductions of real memory requirement and flux calculations are 29.8% and 49.5%, respectively. These investigations show that the proposed cell data strategy is very economical considering the memory requirement and the computation time.

5. CONCLUDING REMARKS

In this article, an efficient unstructured-grid pressure-based method for all speed flows is proposed. The validations are three continuum flow problems and one slip microscale flow problem. The above simulation results give conclusions as follows.

For the lid-driven cavity flow, accurate results are obtained though the mixed-element grids with various aspect ratios used. The incompressible flow in the T-shaped planar branch is also well predicted by the all-speed algorithm. The supersonic flow over a bump was simulated. The shock formation and reflection due to compressibility are captured with reasonable accuracy. The slip flow in a microchannel that has very low speed and compressibility effects is computed. The pressure nonlinearity is predicted accurately. These incompressible and compressible flow simulations show that the proposed unstructured-grid all-speed algorithm has effective capability even up to the slip flow regime in any grid types.

For the unstructured-grid pressure-based method with arbitrary cell topologies, applying the proposed node-based cell data strategy is proved to be an efficient methodology that reduces the memory requirement and flux calculations without losing any accuracy.

REFERENCES


