Ninth International Conference on Computational Fluid Dynamics (ICCFD9), Istanbul, Turkey, July 11-15, 2016

ICCFD9-2016-198

Comparative Study of the CTM and SDM-IDC Methods for Diffusive Fluxes Calculation in the CFD Code Based on SIMPLE Algorithm on Highly Skewed Meshes

Pannasit Borwornpiyawat[1], Ekachai Juntasaro[1], Abdul Ahad Narejo[1]
Philippe Traoré[2], Matthias Meinke[3] & Varangrat Juntasaro[4]

[1]The Sirindhorn International Thai-German Graduate School of Engineering (TGGS), King Mongkut's University of Technology North Bangkok, Bangkok 10800, Thailand.
[2] Institut PPRIME, Université de Poitiers, Poitiers 86000, France.
[3] Institute of Aerodynamics, RWTH Aachen University, Aachen 52062, Germany.
[4] Department of Mechanical Engineering, Kasetsart University, Bangkok, 10900, Thailand.

Corresponding author: ekachai.j.mesd@tggs-bangkok.org

Abstract: The performance of two finite-volume methods for diffusive fluxes calculation, i.e. the Coordinate Transformation Method (CTM) and Surface Decomposition Method with Improved Deferred Correction Scheme (SDM-IDC), is analyzed for regular and highly skewed meshes. After the implementation in an in-house CFD code for laminar incompressible flow based on the SIMPLE algorithm, the accuracy, computational efficiency and convergence speed for steady state problems is assessed by simulating various flow problems. The lid-driven skewed cavity case shows that the CTM method can perform better than the SDM-IDC method in terms of robustness and accuracy while the SDM-IDC method is computationally more efficient.

Keywords: Computational Fluid Dynamics, Diffusive Fluxes Calculation, Skewed Cavity Flow, Highly Skewed Meshes.

1 Introduction

The accurate computation of the diffusive fluxes is one of the main concerns in the field of computational fluid dynamics and heat transfer. For that purpose, the CTM method was presented in [1] and the SDM-IDC method was presented in [1] – [3] by Traoré et al in 2009 and 2014. By solving the Poisson equation for a scalar field, it was revealed that the SDM-IDC method is more accurate for determination of the second-order derivatives than the CTM method. However, the question arises what happens if those two methods are used to solve the Navier-Stokes equations, not just the Poisson equation.

In this paper, these two methods are implemented into an in-house solver for the Navier-Stokes equations to analyze in more detail whether the same results with respect to the accuracy are obtained as for the Poisson equation. The CFD code solves the Navier-Stokes equations for 2D steady incompressible laminar flow and is based on the SIMPLE algorithm, in which the detail can be found in [4] and [5]. The performance of the two discretization methods are compared not only in terms of accuracy and convergence but also in computational costs. The 2D lid-driven skewed cavity is used as a test case to demonstrate their performances. For this case, meshes with different skewness ranging from purely orthogonal to angles of 15, 30, 45, 60, 75, and 89 degrees are used, as shown in Figure 1.

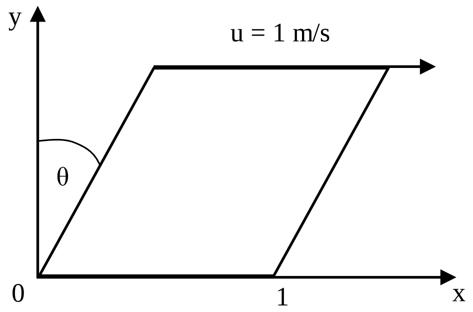

Figure 1: The non-orthogonal angle θ and the computational domain of the skewed cavity.

2 Methods for Diffusive Fluxes Calculation

In order to illustrate the differences between the CTM method and the SDM-IDC method, first consider the governing equations:

$$\rho(\vec{\nabla} \cdot \vec{V}) = 0 \tag{1}$$

$$\rho(\vec{V} \cdot \vec{\nabla})\phi = -\vec{\nabla}P + \mu\vec{\nabla}^2\phi \tag{2}$$

where ϕ represents the velocity components u and v for the x- and y-momentum equations respectively. By integrating Equations (1) and (2) over a cell and applying the divergence theorem, the following discrete forms are obtained:

$$\sum_f F_f = 0 \tag{3}$$

$$\sum_f F_f \phi_f = -\sum_f A_f P_f + \sum_f \mu(\vec{\nabla}\phi)_f \cdot \vec{A}_f \tag{4}$$

where F_f, $\vec{V}_f$ and $\vec{A}_f$ can be expressed as:

$$F_f = \rho \vec{V}_f \cdot \vec{A}_f \tag{5}$$

$$\vec{V}_f = u_f \vec{i} + v_f \vec{j} \tag{6}$$

$$\vec{A}_f = A_f \vec{n}_f = A_x \vec{i} + A_y \vec{j} \tag{7}$$

The diffusion term in Equation (4) can be split into two parts: the primary diffusion along cell centroid direction and the remaining part, i.e. the secondary diffusion, S_f, due to the non-orthogonality of the grid, which are the first and second terms on the right-hand side of Equation (8) respectively.

$$\sum_f \mu(\vec{\nabla}\phi)_f \cdot \vec{A}_f = D_f(\phi_K - \phi_P) + S_f \tag{8}$$

where D_f is defined by:

$$D_f = \frac{\mu \vec{A}_f \cdot \vec{A}_f}{\Delta\xi \vec{A}_f \cdot \vec{e}_\xi} \tag{9}$$

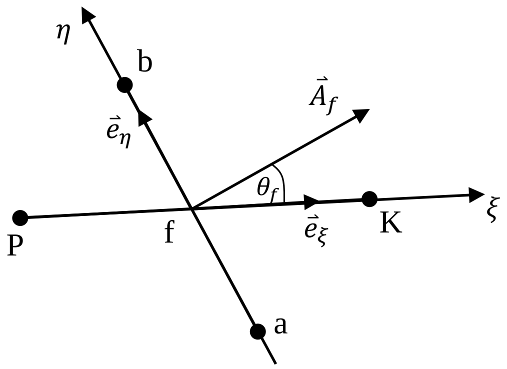

Figure 2: Geometric relationship between two cells on a common face “f”.

When the grid is non-orthogonal, the secondary diffusion term becomes more and more important as the non-orthogonal angle increases. The accurate computation of these secondary fluxes leads to the accurate results.

Two methods are presented in this paper. Both methods use the second-order central differencing scheme in their derivation and can be put in the form of Equation (8). The difference is in the calculation of the secondary diffusion part. The derivation of both methods can be described as follows:

2.1 Coordinate Transformation Method (CTM)

In this method, the diffusion term is defined in the Cartesian coordinate as:

$$\mu(\vec{\nabla}\phi)_f \cdot \vec{A}_f = \mu(\phi_x A_x + \phi_y A_y) \tag{10}$$

where $\phi_x \equiv \dfrac{\partial \phi}{\partial x}$ and $\phi_y \equiv \dfrac{\partial \phi}{\partial y}$.

Equation (10) is then transformed into a local coordinate system $\xi - \eta$ as shown in Figure 2. The following equation is obtained:

$$\begin{aligned} \phi_\xi &= \phi_x x_\xi + \phi_y y_\xi \\ \phi_\eta &= \phi_x x_\eta + \phi_y y_\eta \end{aligned} \tag{11}$$

After re-arranging Equation (11), the expression for ϕ_x and ϕ_y can be written as:

$$\begin{aligned} \phi_x &= \frac{\phi_\xi y_\eta - \phi_\eta y_\xi}{x_\xi y_\eta - x_\eta y_\xi} \\ \phi_y &= \frac{\phi_\eta x_\xi - \phi_\xi x_\eta}{x_\xi y_\eta - x_\eta y_\xi} \end{aligned} \tag{12}$$

where ϕ_ξ, ϕ_η, x_ξ, x_η, y_ξ and y_η can be expressed by the second-order central differencing scheme as:

$$\phi_\xi = \frac{\phi_K - \phi_P}{\Delta\xi} \qquad x_\xi = \frac{x_K - x_P}{\Delta\xi} \qquad y_\xi = \frac{y_K - y_P}{\Delta\xi}$$
$$\phi_\eta = \frac{\phi_b - \phi_a}{\Delta\eta} \qquad x_\eta = \frac{x_b - x_a}{\Delta\eta} \qquad y_\eta = \frac{y_b - y_a}{\Delta\eta} \tag{13}$$

where $\Delta\xi$ is the distance between the two centroids P and K, and $\Delta\eta$ is the distance between the nodes a and b.

After substituting Equation (13) into Equation (12) and then into Equation (10) with some arrangements, the secondary diffusion term, S_f, in Equation (8) is obtained as:

$$S_f = \mu A_f \frac{\phi_b - \phi_a}{\Delta\eta} \tan\theta_f \tag{14}$$

The values at node a and node b are calculated by the area-weighted averaging method as shown in Equation (15) where N is the number of cells sharing the same node and $\Delta\forall$ is the volume of the cell.

$$\phi_{a,b} = \frac{\sum_{i=1}^{N} \frac{1}{\Delta\forall_i}\phi_i}{\sum_{i=1}^{N} \frac{1}{\Delta\forall_i}} \tag{15}$$

2.2 Surface Decomposition Method with Improved Deferred Correction Scheme (SDM-IDC)

In this method, the surface normal vector is decomposed into two parts:

$$\vec{n}_f = \vec{n}_1 + \vec{n}_2 \tag{16}$$

According to Equation (16), the diffusion term is then decomposed as:

$$\mu(\vec{\nabla}\phi)_f \cdot \vec{A}_f = \mu A_f (\vec{\nabla}\phi)_f \cdot \vec{n}_1 + \mu A_f (\vec{\nabla}\phi)_f \cdot \vec{n}_2 \tag{17}$$

The length of $\vec{n}_1$ is defined, according to the improved deferred correction (IDC) scheme in [1] which shows better results when compared to the standard deferred correction scheme (SDC) in [2] and [3], as:

$$\|\vec{n}_1\| = \frac{\|\vec{n}_f\|}{\cos\theta_f} = \frac{1}{\cos\theta_f} \tag{18}$$

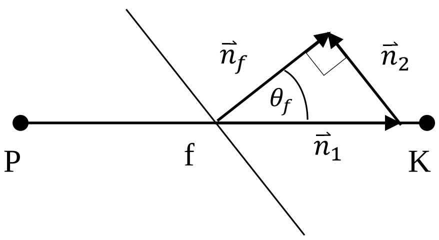

Figure 3: Decomposition of the surface normal vector according to the improved deferred correction (IDC) scheme.

After introducing Equation (18) into Equation (17), the secondary diffusion term, S_f, in Equation (8) is obtained as:

$$S_f = \mu A_f (\vec{\nabla}\phi)_f \cdot \vec{e}_\xi \tan\theta_f \tag{19}$$

However, instead of using Equation (19) to calculate the secondary diffusion in the SDM-IDC method, the deferred correction approach leads to:

$$S_f = \mu(\vec{\nabla}\phi)_f \cdot \vec{A}_f - D_f (\vec{\nabla}\phi)_f \cdot \vec{e}_\xi \Delta\xi \tag{20}$$

Equation (20) is written in terms of the difference between the total diffusion across face "f" and the primary diffusion along the cell centroid direction. The face gradient is calculated from the result in the previous iteration of the cell on both sides of the face using the linear interpolation as follows:

$$(\vec{\nabla}\phi)_f = (\vec{\nabla}\phi)_P + \overline{Pf}\left(\frac{(\vec{\nabla}\phi)_K - (\vec{\nabla}\phi)_P}{\overline{Pf} + \overline{fK}}\right) \tag{21}$$

The gradient at cell center is calculated by the divergence theorem which can be written as:

$$(\vec{\nabla}\phi)_P = \frac{1}{\Delta\forall_P}\sum_f \phi_f \vec{A}_f \tag{22}$$

where ϕ_f in Equation (22) is interpolated in the same manner as the face gradient in Equation (21).

3 Lid-Driven Skewed Cavity Flow

Cavity flow is one of the most common test cases for the validation of the numerical schemes. It has been chosen as a test case in this study due to the ease of controlling the grid skewness and the available existing numerical results for comparison. Several studies have been done on the skewed cavity flow such as Erturk and Dursun in 2007 [6], and Thaker and Banerjee in 2011 [7]. The Reynolds number is set to 1000 in this study. Left, bottom, and right boundaries are prescribed with the stationary no-slip wall condition while the top boundary is a moving wall, as shown earlier in Figure 1.

The orthogonal grid, $\theta = 0$ degree, is used in the validation process of the in-house CFD code because of the absence of the secondary diffusion term. The result obtained from the in-house CFD code is compared to a licensed commercial software, ANSYS Fluent, as shown in Figure 4 where both results are in good agreement and only the X-velocity profile is shown.

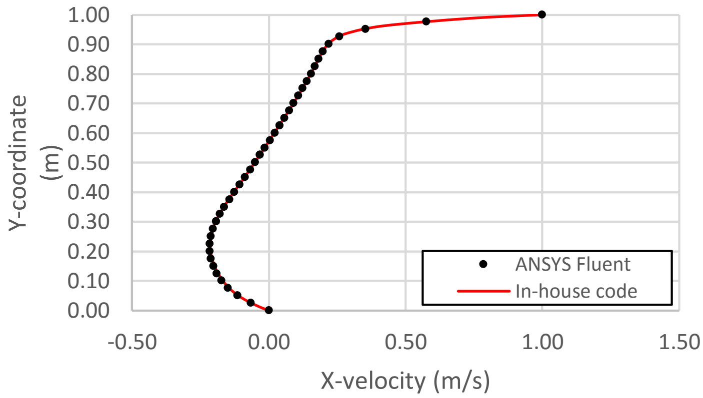

Figure 4: Comparison of the X-velocity profile obtained from the in-house code and ANSYS Fluent at θ = 0 degree.

The 2[nd]-order upwind scheme has been used and the grid resolution has been increased until the result is matched with the reference data taken from [6]. The study shows that the grid resolution of 120 x 120 cells is sufficiently fine as shown in Figure 5 where only the X-velocity profile is shown.

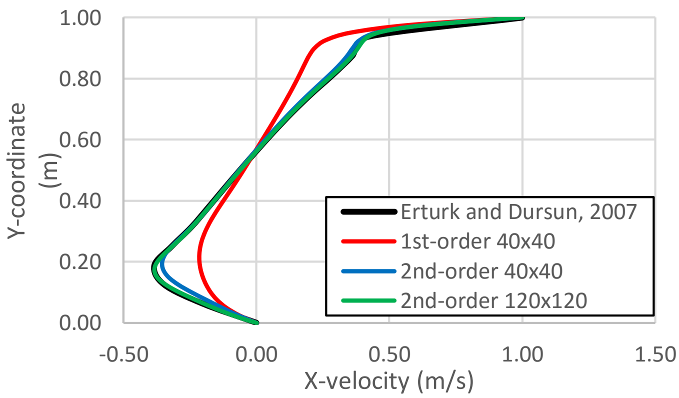

Figure 5: Comparison of the X-velocity profile obtained from different schemes and grid resolutions at θ = 0 degree.

4 Results and Discussion

Since there is no contribution from the secondary diffusion term on the orthogonal grid, both methods show the same result as shown in Figure 4 and Figure 5. The differences can be seen when the grid is non-orthogonal in which the X-velocity and Y-velocity profiles are plotted along the line A-B and C-D as shown in Figure 6 respectively.

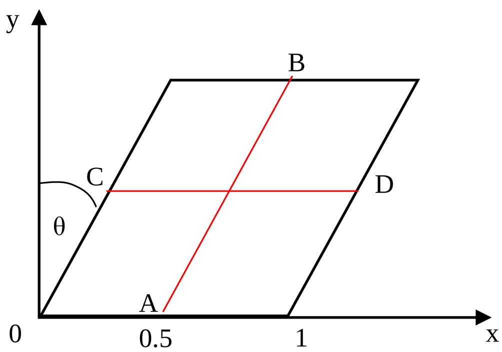

Figure 6: The middle line A-B and C-D in the domain.

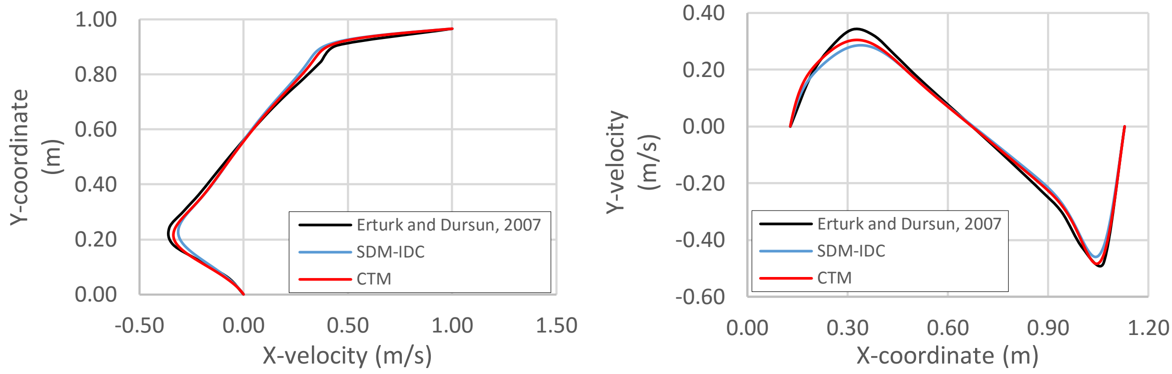

Figure 7: X-velocity profile (Left) and Y-velocity profile (Right) at θ = 15 degrees.

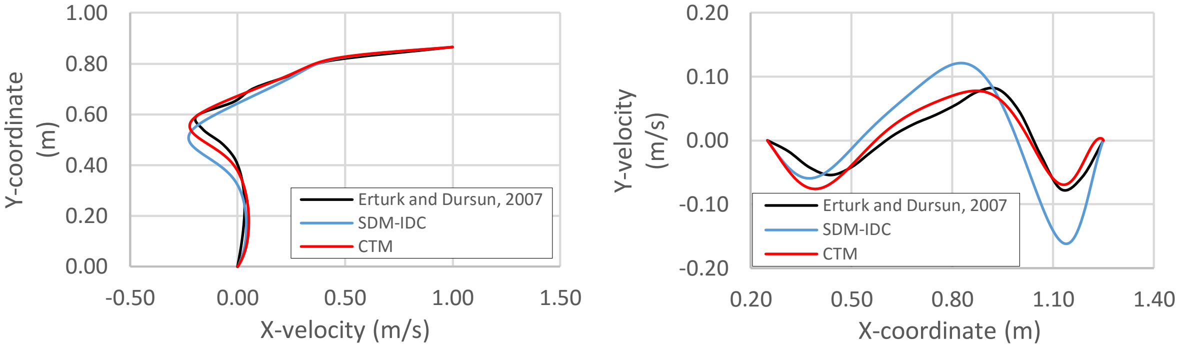

Figure 8: X-velocity profile (Left) and Y-velocity profile (Right) at θ = 30 degrees.

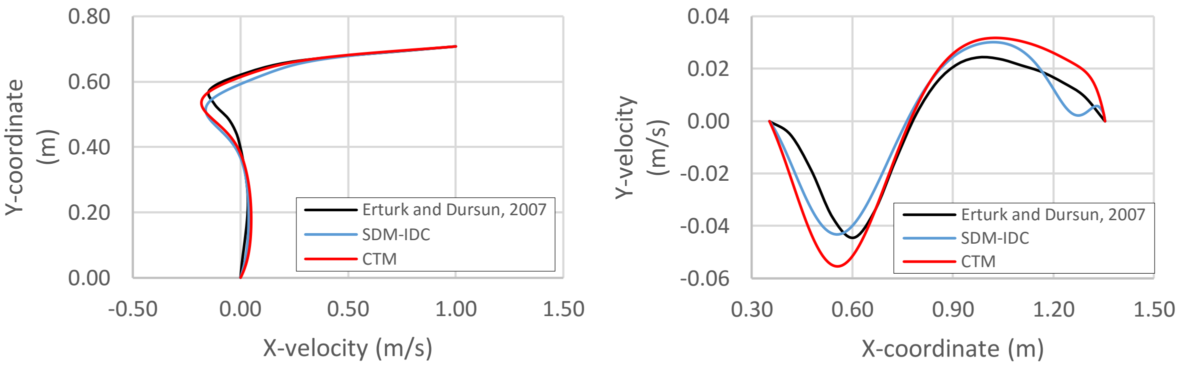

Figure 9: X-velocity profile (Left) and Y-velocity profile (Right) at θ = 45 degrees.

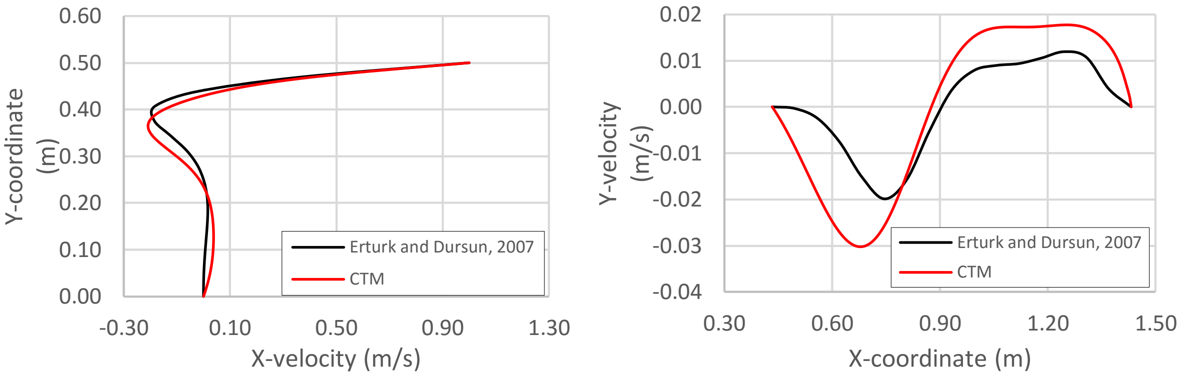

Figure 10: X-velocity profile (Left) and Y-velocity profile (Right)
at θ = 60 degrees.

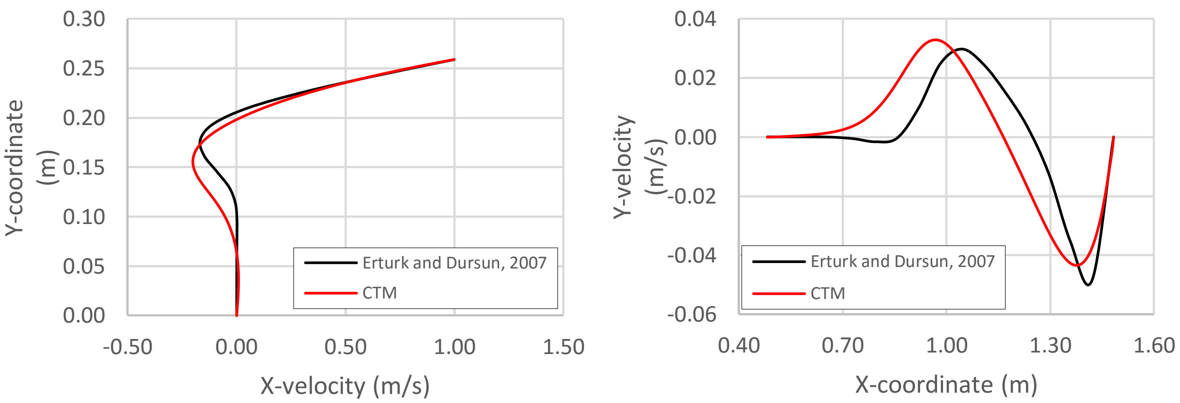

Figure 11: X-velocity profile (Left) and Y-velocity profile (Right)
at θ = 75 degrees.

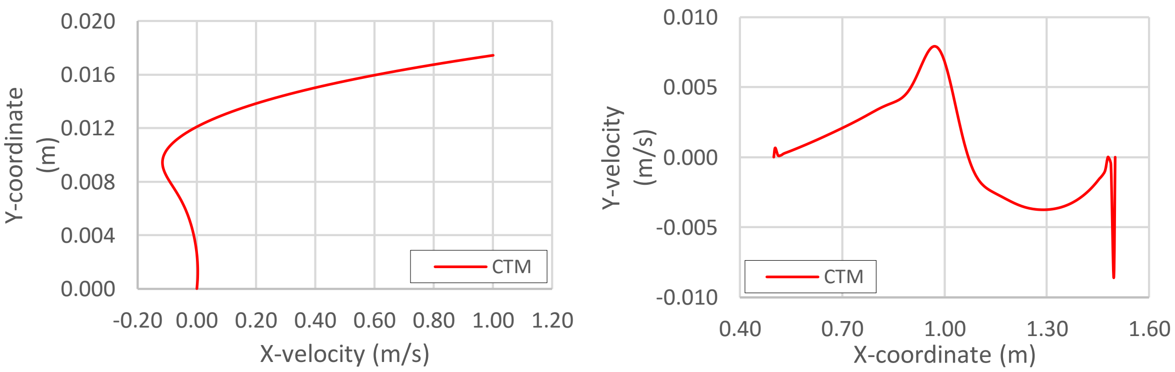

Figure 12: X-velocity profile (Left) and Y-velocity profile (Right)
at θ = 89 degrees.

In contrast to the earlier study by Traoré et al in [1], the CTM method is more accurate and converges even on the extremely skewed mesh where θ = 89 degrees while the SDM-IDC method fails when the angle is greater than 45 degrees.

In terms of the computational cost, the SDM-IDC method requires lower memory of the computer and also needs less time per iteration than the CTM method (observed during the iterations) due to the fact that the SDM-IDC method directly uses the variables at the cell center whereas the CTM method needs to calculate the values at the vertices in every iteration. The numbers of required iterations of both methods to converge to the limit of 1.0e-6 are summarized in Table 1.

θ (degrees)	Number of iterations	
	SDM-IDC	CTM
0	7,998	7,998
15	10,408	13,067
30	29,409	41,647
45	11,394	9,821
60	-	16,136
75	-	20,089
89	-	195,304

Table 1: Summary of the number of iterations.

From the programming point of view, the accuracy and computational cost of both methods can be explained by the cell stencil in Figure 13. In 2D structured grid, the CTM method requires all 9 cells in order to calculate the total fluxes in the considered cell “P” which is the reason why this method is more accurate than the SDM-IDC method which requires only 5 cells.

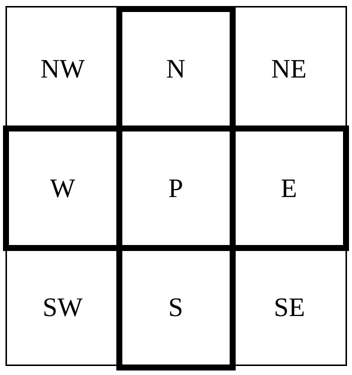

Figure 13: Cell stencil in case of 2D structured grid.

However, the SDM-IDC method can be easily extended to 3D grid since there are always two cells connected to a face, hence Equation (20) can be directly applied to 3D problems. On the other hand, Equation (14) is resulted from the fact that the tangential direction can be uniquely defined in 2D problems. In 3D problems, a face changes from a 2D-line to a 3D-surface, and hence the tangential direction can be defined in numerous ways, which make the CTM method becomes trickier to extend to 3D grid.

5 Conclusion

It can be concluded that the CTM method can perform better than the SDM-IDC method when applied to solve the Navier-Stokes equations due to the higher accuracy and ability to converge on the extremely skewed mesh. However, the advantage of the SDM-IDC method is the lower requirement of computational cost and lower time per iteration.

Acknowledgement

This research work is financially supported by National Metal and Materials Technology Center (MTEC), National Science and Technology Development Agency (NSTDA), Thailand.

The first author (P. Borwornpiyawat) would like to gratefully thank Univ.-Prof. Dr.-Ing. Wolfgang Schröder for the invitation and his guidance during the internship/research at Institute of Aerodynamics (AIA), RWTH Aachen University, Aachen, Germany, and also Dr. Jian Wu for his helpful discussion.

References

[1] J. Wu, P. Traoré. *Similarity and comparison of three finite-volume methods for diffusive fluxes computation on nonorthogonal meshes*. Numerical Heat Transfer: Part B, 64: 118 – 146, 2014.

[2] P. Traoré, Y. Ahipo, C. Louste. *A robust and efficient finite volume scheme for the discretization of diffusive flux on extremely skewed meshes in complex geometries*. Journal of Computational Physics. 228: 5148 – 5159, 2009.

[3] Y. Ahipo, P. Traoré. *A robust iterative scheme for finite volume discretization of diffusive flux on highly skewed meses*. Journal of Computational and Applied Mathematics. 231: 478 – 491, 2009.

[4] J.Y. Murthy, S.R. Mathur. *Numerical methods in heat, mass and momentum transfer*. Draft Notes: ME 608, School of Mechanical Engineering, Purdue University, 2002.

[5] J.H. Ferziger, M. Perić. *Computational methods for fluid dynamics*, 3rd revised edition. Springer, 2002.

[6] E. Erturk, B. Dursun. *Numerical solutions of 2-D steady incompressible flow in a driven skewed cavity*. ZAMM - Journal of Applied Mathematics and Mechanics, Vol. 87, 2007, pp. 377-392.

[7] J.P. Thaker, B.J. Banerjee. *Numerical simulation of flow in lid-driven cavity using OpenFOAM*. International Conference on Current Trends in Technology: NUiCONE-2011, Institute of Technology, Nirma University, Ahmedabad, 2011.