Oral presentation | Numerical methods
Numerical methods-VII
Thu. Jul 18, 2024 2:00 PM - 4:00 PM Room A

[11-A-03] Direct Poisson Solver Combining Domain Decomposition and Influence Matrix Methods and its application to DNS of Oscillating Grid Turbulence

*Toru Yamada[1], Yuki Ohno[1], Yohei Morinishi[1] （1. Nagoya Institute of Technology）

Keywords: Oscillating Grid Turbulence, Direct Numerical Simulation, Direct Poisson Solver

©Retained by Authors

**Twelfth International Conference on
Computational Fluid Dynamics (ICCFD12),
Kobe, Japan, July 14-19, 2024**

Direct Poisson Solver Combining Domain Decomposition and Influence Matrix Methods and its application to DNS of Oscillating Grid Turbulence

T. Yamada*, Y. Ohno*, and Y. Morinishi*
Corresponding author: yamada.toru@nitech.ac.jp
* Graduate School of Engineering, Nagoya Institute of Technology, Gokiso-cho, Showa-ku, Nagoya 466-8555, Japan.

Abstract: This study presents the application of a direct Poisson solver combining domain decomposition and influence matrix methods to the Direct Numerical Simulation (DNS) of oscillating grid turbulence (OGT). Solving the pressure Poisson equation is one of the major challenges in computational fluid dynamics. Traditional direct methods are accurate but difficult to be applied to large-scale problems, while iterative methods can suffer from slow convergence. The hybrid approach enables the use of the direct methods for the problem with complex computational geometry. The results show that this direct solver significantly reduces computation time compared to the iterative BiCGSTAB method. The DNS results are validated against experimental data, demonstrating good agreement in the vertical distribution of velocity fluctuation intensity. Therefore, the efficiency of this direct method for simulating the OGT turbulent flows is confirmed.

Keywords: Oscillating Grid Turbulence, Direct Numerical Simulation, Direct Poisson Solver.

1 Introduction

Solving the pressure Poisson equation is the most time-consuming part in computational fluid dynamics (CFD) for incompressible flows. The Poisson equation arises from the divergence-free constraint on the velocity field, which is essential for maintaining incompressibility. There are two main categories of methods to solve the Poisson equation, i.e. direct methods and iterative methods. The choice of method depends on various factors including the nature of the Poisson equation to be solved, computational resources, the desired accuracy, and computation time.

Direct methods for solving the Poisson equation, such as Gaussian elimination, LU decomposition, and direct solvers based on the Fast Fourier Transform (FFT), aim to find an exact solution in a finite number of operations [1]. The former two methods are typically very efficient for small to moderately sized problems and provide consistent accuracy. However, their computational cost and memory requirements grow rapidly with the size of the problem. The FFT-based solvers are very efficient but limited for problems with periodic boundary conditions. Despite these limitations, direct methods are valued for their robustness and precision in solving linear systems.

On the other hand, iterative methods such as the Jacobi method, Gauss-Seidel method, Successive Over-Relaxation (SOR) method, and Conjugate Gradient method, solve the Poisson equation by iteratively refining an initial guess until convergence tolerance is achieved [1]. These methods are generally more scalable than direct methods and can handle very large systems more efficiently in terms of memory usage. Iterative methods are particularly useful for problems with complex geometries and boundary conditions. However, they may require a large number of iterations to achieve the desired accuracy, leading to long computation times. Preconditioning techniques are often employed to improve the convergence rate of iterative solvers.

For flow fields involving structures and their movement, iterative methods are generally more applicable due to their flexibility in handling complex and dynamic boundaries. One specific example of such a flow field is oscillating grid turbulence (OGT), which is characterized by a grid moving periodically to generate turbulence. Based on previous studies [2], it is obvious that the DNS (Direct Numerical Simulation) of OGT requires high spatial resolution to accurately capture the turbulent structures, leading to excessive computation times and difficulties when iterative methods are used. The high resolution increases the number of grid points, thereby increasing the size of the linear system to be solved, which can significantly slow down the convergence of iterative methods.

Therefore, this study aims to apply a direct method to the DNS of OGT. The proposed approach combines the domain decomposition method with the influence matrix method. Domain decomposition breaks the global problem into smaller subproblems that can be solved independently, while the influence matrix method ensures the consistency and accuracy of the solution across the subdomains. This hybrid approach was first introduced by Schumann and Benner [3] who successfully applied this method to the simulations involving fluid-structure interactions. Meanwhile, it has been applied to only a couple of studies to date [4, 5].

The computational efficiency of this combined method is discussed in this study based on computation time by comparing it with an available iterative method, i.e. BiCGSTAB (Bi-Conjugate Gradient Stabilized) method [6]. Furthermore, the OGT flow is analyzed in terms of velocity fluctuation intensity.

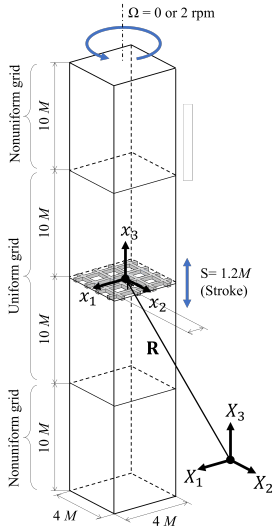


Figure 1: Schematic of computational domain: $(x_1, x_2.x_3)$ is the coordinate of the non-inertial frame, (X_1, X_2, X_3) is the coordinate of inertial frame, and $\mathbf{R}$ is the position vector from the inertial to the moving frames.

2 Methodology

A moving coordinate system, involving translation and rotation, is considered in the present study as shown in Figure 1, where the origin of the system is located at the center of oscillating grid. The dimensionless forms of the continuity and the Navier-Stokes equations in this system for incompressible fluids are expressed by the following formulae:

$$\frac{\partial u_i}{\partial x_i} = 0\,, \tag{1}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_j u_i}{\partial x_j} = -\frac{\partial P}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) + \varepsilon_{ijk}\frac{1}{Ro}u_k + \frac{d^2 R_i}{dt^2}\,, \tag{2}$$

where u_j represents the velocity component in the x_j direction. Also, d^2R_i/dt^2 is the transnational acceleration term. The grid oscillates only in the x_3-direction with $R_3 = 0.5S\sin(f_g t + \theta_0)$ where S and θ_0 are the stroke and initial phase of the oscillation. Re and Ro are Reynolds and Rossby numbers, respectively, and they are defined as follows:

$$Re \equiv \frac{f_g M^2}{\nu},\quad Ro \equiv \frac{f_g}{2\Omega}\,. \tag{3}$$

In the above equations, f_g, Ω, M, and ν are frequency of the grid mesh oscillation, angular velocity for the system rotation, grid mesh size, and kinematic viscosity, respectively. The parameters related to the oscillating grid, f_g and S, are chosen so that the DNS assumes our experiments [7] where (f_g, S, M) = (2 [Hz], 60 [mm], 50 [mm]). The Re value is also set to be the same as the experiment, i.e. $Re = 5000$. The Ro values chosen in this study are ∞ and 4.77, corresponding to 0 and 2 [rpm] in the experiment, respectively. The simulations with these values are called the system at rest and the system with rotation, respectively, in the subsequent sections.

The computational domain for the OGT studied in the present study is shown in Figure 1. The lengths in the horizontal directions (L_1, L_2) are (4, 4) and that in the vertical direction L_3 is 40, respectively, where the length scale is normalized by the grid mesh size M as the characteristic length. For the horizontal (x_1, x_2) directions, computational grids are uniformly distributed. For the vertical (x_3) direction, uniform computational grid is adopted for $-10 \le x_3 \le +10$ while non-uniform grid is used for $|x_3| > 10$. The computational grid for this simulation is 200×200×1400. The boundary conditions for the oscillating grid surface, horizontal directions, and vertical direction are no-slip, periodic, and Neumann conditions, respectively. The spatial discretization is performed with second order central finite difference in a staggered grid. The low-storage 3-stage Runge-Kutta (RK) method [8] is used for

the time discretization, and the time increment is set to be $\Delta t = 1.25 \times 10^3$, where the time is normalized by f_g. For further details of computational setup, please refer to Ref.[9].

2.1 Direct Poisson solver

The pressure Poisson equation in this study is discretized based on SMAC method [10]. This equation becomes a large-scale linear system equation (LSE) which is described by the following equation:

$$\mathrm{L}\mathbf{p} = \mathbf{q}\,, \tag{4}$$

where $\mathbf{p}$ is the pressure correction and $\mathbf{q}$ is the source term. Direct solutions cannot be applied this LSE if it is straightforwardly generated. In order to apply direct solutions, such as TDMA and FFT, to solve the equation, the LSE is modified by combining the domain decomposition (DD) [3] and the influence matrix (IM) [4] methods. Briefly describing, the basic LSE is firstly constructed by DD, and then the solution is obtained by solving substitute equations generated by IM to which fast direct solvers can be applied. The details of generating the DD-IM-modified LSE and solving procedure is described in the following subsections.

2.1.1 Prerequisite of typical Poisson problems with Neumann boundaries

The problem to be solved in this study, Eq.(4), is a so-called "Poisson-Neumann" type problem [3], and the problems of this kind require the solution of Poisson equation with Neumann boundary conditions as follows:

$$\frac{1}{\lambda}\mathrm{div}\,\lambda\,\mathrm{grad}\,p = q,\ \ \lambda > 0 \text{ on } R\,, \tag{5}$$

$$\lambda\mathbf{n}\cdot\mathrm{grad}\,p = g_R \ \text{ on } \partial R\,, \tag{6}$$

where λ is a given space-dependent coefficient which is given, and $\mathbf{n}$ is the outwardly normal unit vector on ∂R. The Gauss's divergence theorem gives the following consistency condition for p and g_R to be satisfied,

$$\iiint_R \lambda q\, dV - \oint_{\partial R} g_R dS = 0\,, \tag{7}$$

ensuring a solution of this problem exists. The solution in this case is not unique, and thus

$$p = p' + a \tag{8}$$

is a solution for an arbitrary constant a if p' is a solution of Eq.(5).

2.1.2 Domain decomposition method

We commence with introducing the domain decomposition method using Eq.(4). The coefficient matrix L has a rank deficit of one under the Neumann condition, i.e. $\det(\mathrm{L}) = 0$. This LSE is able to have a solution if L has two eigenvectors, $\mathbf{u} \neq \mathbf{0}$ and $\mathbf{v} \neq \mathbf{0}$, satisfying the following conditions.

$$\mathrm{L}\mathbf{u} = \mathbf{0},\ \ \mathrm{L}^{\mathrm{T}}\mathbf{v} = \mathbf{0} \Leftrightarrow \mathbf{v}^{\mathrm{T}}\mathrm{L} = \mathbf{0}^{\mathrm{T}}\,. \tag{9}$$

Combining the second equation above with Eq.(4) leads to the following relationship, which is necessary for the solution of Eq.(4) to be unique:

$$\mathbf{v}^{\mathrm{T}}\mathrm{L}\mathbf{p} = \mathbf{v}^{\mathrm{T}}\mathbf{q} = 0. \tag{10}$$

Meanwhile, the solution $\mathbf{p}$ can be solved by the following equation:

$$\mathbf{p} = \mathbf{p}' + a\mathbf{u}\,, \tag{11}$$

if $\mathrm{D}\mathbf{p}' = \mathbf{q}$ in which D is the regularized matrix of L, the D and L differ in at least one row. Also, a in the above equation is an additional constant that adjusts the particular solution $\mathbf{p}'$.

When domain decomposition is performed, the LSE of Eq.(4) is divided into d equations (or domains) by introducing unknown vector $\mathbf{g}$, which is expressed as follows:

$$\mathrm{L}_i\mathbf{p}_i + \mathrm{G}_i\mathbf{g} = \mathbf{q}_i \,, \tag{12}$$

$$\mathbf{g} + \sum_{j=1}^{d} \mathrm{H}_j\mathbf{p}_j = \mathbf{0} \,, \tag{13}$$

where i is the index of the subdomains ($i = 1, 2, \cdots, d$) and $\mathbf{g}$ is the gradient of $\mathbf{p}$ at the interfaces between adjacent subdomains. G_i is defined to correct the effect of $\mathrm{G}_i\mathbf{g}$ on the original coefficient matrix L. H_j is the operator for generating the pressure gradient at each domain interface.

Each decomposed equation has an eigenvector similar to that of the original LSE. Therefore, applying Eq.(12) to Eq.(10) results in

$$\mathbf{v}^{\mathrm{T}} \left(\mathbf{q}_i - \mathrm{G}_i\mathbf{g} \right) = 0. \tag{14}$$

Also, the following equation holds for each decomposed domain by using the regularized matrix D_i of L_i,

$$\mathrm{D}_i\mathbf{p}'_i + \mathrm{G}_i\mathbf{g} = \mathbf{q}_i \,, \tag{15}$$

$$\begin{cases} \mathbf{p}_i = \mathbf{p}'_i + a_i\mathbf{u}_i & (i < d) \,, \\ \mathbf{p}_i = \mathbf{p}'_i & (i = d) \,. \end{cases} \tag{16}$$

2.1.3 Eigenvectors u and v

In this subsection, the eigenvectors $\mathbf{u}$ and $\mathbf{v}$ for the matrix L are introduced. First of all, we define the volume of (i, j, k)th computational cell as $v_{ijk} = \Delta x_{1,i}\Delta_{2,j}\Delta_{3,k}$, where i, j, and k are indices for the x_1, x_j, and x_3 directions and $\Delta x_{d,i}$ indicates the length of the ith cell for the x_d direction. Next, the cell-volume diagonal matrix, V, is defined using v_{ijk} as follows:

$$\mathrm{V} = \mathrm{diag}\left(v_{ijk}\right), \quad \mathrm{V}^{\mathrm{T}} = \mathrm{V}. \tag{17}$$

Multiplying V with the off-diagonal matrix L results in a diagonal matrix VL, and the following relationship holds.

$$\mathrm{VL} = \left(\mathrm{VL}\right)^{\mathrm{T}} = \mathrm{L}^{\mathrm{T}}\mathrm{V} \,. \tag{18}$$

Also, the gradients of $\mathbf{p}$ (also $\mathbf{p}'$ and $\mathbf{u}$) at wall boundaries are zero, and thus the following vector $\mathbf{u}$ is confirmed to be a eigenvector fulfilling $\mathrm{L}\mathbf{u} = \mathbf{0}$:

$$\mathbf{u} = \left\{u_{ijk}\right\}, \quad u_{ijk} = 1 \,. \tag{19}$$

Additionally, based on Eqs.(18) and (19), the eigenvector, $\mathbf{v}$, satisfying $\mathrm{L}^{\mathrm{T}}\mathbf{v} = \mathbf{0}$ is,

$$\mathbf{v} = \mathrm{V}\mathbf{u} = \left\{v_{ijk}\right\} \,. \tag{20}$$

2.1.4 Additional processing for the influence matrix method

In order to apply the IM method to the present study, an additional process needs to be made to the LSE. As the IM method is described in detail later, a substitute LSE, for which fast direct solvers are applied, is solved twice. The first process correspond to solving the LSE with assuming $\mathbf{g}$ and a_i are equal to $\mathbf{0}$ and 0, respectively, and the second one is with using the values obtained from the first one. In the first process, the consistency condition Eq.(14) is not satisfied with the first estimates $\mathbf{g}^1$ ($= \mathbf{0}$), and therefore, the second estimates $\mathbf{g}^2$ are defined such that Eq.(14) is satisfied as follows:

$$\mathbf{g}^2 = \mathbf{g}^1 + \sum_{j=1}^{d-1} b_j\mathbf{e}_j \,, \tag{21}$$

where b_j ($j = 1, 2, \cdots, d-1$) is coefficients providing $(d-1)$ degrees of consistency, and $\mathbf{e}_j$ is the linearly independent vector ensuring the same consistency to other subdomains. $\mathbf{e}_j$ is linearly independent vectors generated by the modified Gram-Schmidt method. Substituting this equation into Eq.(14) and Eq.(15)

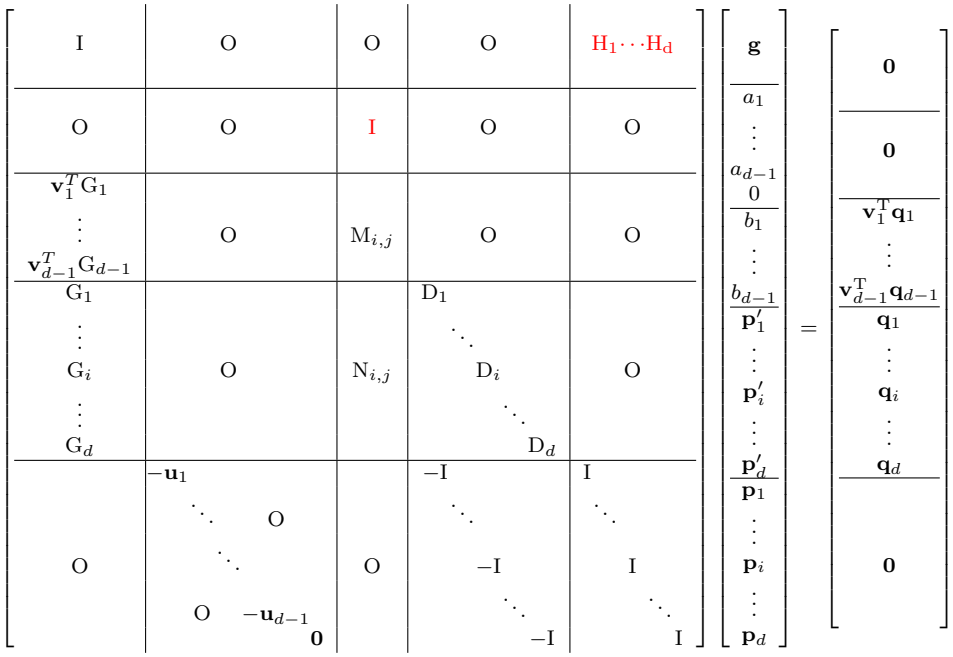


Figure 2: Detailed formula of the A problem, Eq.(24). The elements with red color prevent the application of fast direct solvers.

results in the following formulae:

$$\mathbf{v}_i^{\mathrm{T}}\mathrm{G}_i\mathbf{g} + \sum_{j=1}^{d-1}\mathrm{M}_{i,j}\,b_j = \mathbf{v}_i^{\mathrm{T}}\mathbf{q}_i \quad (i = 1, 2, \cdots, d-1)\ , \tag{22}$$

$$\mathrm{D}_i\mathbf{p}' + \mathrm{G}_i\mathbf{g} + \sum_{j=1}^{d-1}\mathrm{N}_{i,j}\,b_j = \mathbf{q}_i \quad (i = 1, 2, \cdots, d)\ , \tag{23}$$

where $\mathrm{M}_{i,j} = \mathbf{v}_i^{\mathrm{T}}\mathrm{G}_i\mathbf{e}_j$ and $\mathrm{N}_{i,j} = \mathrm{G}_i\mathbf{e}_j$. It should be noted that $b_i = 0$ for $i = 1, 2, \cdots d-1$ if the consistency condition, Eq.(15), is fulfilled in each decomposed domain. Based on Eqs.(13) and (16) together with the additional process, a new A problem is given which provides a solution for the original pressure Poisson equation, Eq.(4), as follows:

$$\mathrm{A}\mathbf{x} = \mathbf{y}\ , \quad \det(\mathrm{A}) \neq 0\ , \tag{24}$$

where the detailed formulae of A, $\mathbf{x}$, and $\mathbf{y}$ are shown in Figure 2. The detailed formula of the substitute B problem is shown in the next section.

2.1.5 Influence matrix method

As can be seen in Figure 2, the fast direct solvers cannot be applied to Eq.(24). In the IM method, in order to give a substitute B problem which is quickly solvable, A ($n \times n$ matrix) and $\mathbf{y}$ (n vector) are first partitioned as follows:

$$\mathrm{A} = \begin{bmatrix} \mathrm{A}_1 \\ \mathrm{A}_2 \end{bmatrix}\ , \quad \mathbf{y} = \begin{Bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{Bmatrix}\ . \tag{25}$$

A_1 and A_2 are $m \times n$ and $(n-m) \times n$ matrices, respectively, where A_1 corresponds to the irregular equations that prevent application of fast direct solvers to the A problem. Based on the above expressions, Eq.(24) can be rewritten as:

$$\begin{bmatrix} \mathrm{A}_1 \\ \mathrm{A}_2 \end{bmatrix}\mathbf{x} = \begin{Bmatrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{Bmatrix}\ , \tag{26}$$

$$
\left[\begin{array}{c|c|c|c|c}
\mathrm{I} & \mathrm{O} & \mathrm{O} & \mathrm{O} & \mathrm{O} \\
\hline
\mathrm{O} & \mathrm{I} & \mathrm{O} & \mathrm{O} & \mathrm{O} \\
\hline
\begin{array}{c}\mathbf{v}_1^T\mathrm{G}_1 \\ \vdots \\ \mathbf{v}_{d-1}^T\mathrm{G}_{d-1}\end{array} & \mathrm{O} & \mathrm{M}_{i,j} & \mathrm{O} & \mathrm{O} \\
\hline
\begin{array}{c}\mathrm{G}_1 \\ \vdots \\ \mathrm{G}_i \\ \vdots \\ \mathrm{G}_d\end{array} & \mathrm{O} & \mathrm{N}_{i,j} & \begin{array}{ccc}\mathrm{D}_1 & & \\ & \ddots & \\ & & \mathrm{D}_i \\ & & & \ddots \\ & & & & \mathrm{D}_d\end{array} & \mathrm{O} \\
\hline
\mathrm{O} & \begin{array}{ccc}-\mathbf{u}_1 & & \\ \ddots & & \mathrm{O} \\ & \ddots & \\ \mathrm{O} & & -\mathbf{u}_{d-1} \\ & & \mathbf{0}\end{array} & \mathrm{O} & \begin{array}{ccc}-\mathrm{I} & & \\ & \ddots & \\ & & -\mathrm{I} \\ & & & \ddots \\ & & & & -\mathrm{I}\end{array} & \begin{array}{ccc}\mathrm{I} & & \\ & \ddots & \\ & & \mathrm{I} \\ & & & \ddots \\ & & & & \mathrm{I}\end{array}
\end{array}\right]
\left[\begin{array}{c}
\mathbf{g}^1 \\ \hline
a_1^1 \\ \vdots \\ a_{d-1}^1 \\ 0 \\ \hline
b_1 \\ \vdots \\ b_{d-1} \\ \hline
\mathbf{p}'_1 \\ \vdots \\ \mathbf{p}'_i \\ \vdots \\ \mathbf{p}'_d \\ \hline
\mathbf{p}_1 \\ \vdots \\ \mathbf{p}_i \\ \vdots \\ \mathbf{p}_d
\end{array}\right]
=
\left[\begin{array}{c}
\mathbf{0} \\ \hline
\mathbf{0} \\ \hline
\mathbf{v}_1^{\mathrm{T}}\mathbf{q}_1 \\ \vdots \\ \mathbf{v}_{d-1}^{\mathrm{T}}\mathbf{q}_{d-1} \\ \hline
\mathbf{q}_1 \\ \vdots \\ \mathbf{q}_i \\ \vdots \\ \mathbf{q}_d \\ \hline
\mathbf{0}
\end{array}\right]
$$

Figure 3: Detailed formula of the B problem, Eq.(28). The elements with blue color are the difference from the A problem (see Figure 2) which enable fast direct solvers to be used.

or

$$
\begin{cases}
\mathrm{A}_1\mathbf{x} = \mathbf{y}_1 \,, \\
\mathrm{A}_2\mathbf{x} = \mathbf{y}_2 \,.
\end{cases}
\tag{27}
$$

A quickly solvable B problem is then defined as:

$$
\mathrm{B}\bar{\mathbf{x}} = \bar{\mathbf{y}} \,, \quad \det(\mathrm{B}) \neq 0 \,.
\tag{28}
$$

As with the A problem, partitioning of B ($n \times n$ matrix) and $\bar{\mathbf{y}}$ (n vector) results in

$$
\mathrm{B} = \left[\begin{array}{c} \mathrm{B}_1 \\ \mathrm{A}_2 \end{array}\right] \,, \quad \bar{\mathbf{y}} = \left\{\begin{array}{c} \bar{\mathbf{y}}_1 \\ \mathbf{y}_2 \end{array}\right\} \,,
\tag{29}
$$

where B_1 is a matrix having the size of $m \times n$, chosen so that fast direct solvers can be applied to it. Also, $\bar{\mathbf{y}}_1$ is an m-size vector which can be arbitrary chosen. In this study, $\bar{\mathbf{y}}_1 = \mathbf{0}$ is initially given. Based on the above expressions, Eq.(28) can be rewritten as follows:

$$
\left[\begin{array}{c} \mathrm{B}_1 \\ \mathrm{A}_2 \end{array}\right] \bar{\mathbf{x}} = \left\{\begin{array}{c} \bar{\mathbf{y}}_1 \\ \mathbf{y}_2 \end{array}\right\} \,,
\tag{30}
$$

or

$$
\begin{cases}
\mathrm{B}_1\bar{\mathbf{x}} = \bar{\mathbf{y}}_1 \,, \\
\mathrm{A}_2\bar{\mathbf{x}} = \mathbf{y}_2 \,.
\end{cases}
\tag{31}
$$

The detailed formula of Eq.(28) is shown in Figure 3. It should be noted that solving this equation corresponds to the first process of solving a substitute LSE with assuming $\mathbf{g}$ and a_i are equal to zero, as described in the subsection 2.1.4.

Because the solution for the B problem is $\bar{\mathbf{x}}$, next step is to utilize the B problem to obtain $\mathbf{x}$ without solving the A problem. Assuming that $\mathbf{x}$ can be obtained by solving the following modified B problem:

$$
\mathrm{B}\mathbf{x} = \bar{\mathbf{y}} + \mathrm{W}\mathbf{w} \,,
\tag{32}
$$

where

$$\mathrm{W} = \begin{bmatrix} \mathrm{W}_1 \\ O \end{bmatrix} . \tag{33}$$

W_1 is $m \times m$ matrix and usually $\mathrm{W}_1 = \mathrm{I}$ (I : unit matrix). The following equation is obtained by multiplying both sides by B^{-1} from left.

$$\mathrm{B}^{-1}\mathrm{B}\mathbf{x} = \mathrm{B}^{-1}\left(\bar{\mathbf{y}} + \mathrm{W}\mathbf{w}\right) ,$$

$$\Leftrightarrow \mathrm{B}^{-1}\mathrm{W}\mathbf{w} = \mathbf{x} - \mathrm{B}^{-1}\bar{\mathbf{y}} . \tag{34}$$

By substituting $\bar{\mathbf{x}} = \mathrm{B}^{-1}\bar{\mathbf{y}}$ from Eq.(28) into the above equation, one can get

$$\mathrm{B}^{-1}\mathrm{W}\mathbf{w} = \mathbf{x} - \bar{\mathbf{x}} . \tag{35}$$

Additionally, by multiplying both sides of the above equation by A_1, the equation becomes

$$\mathrm{A}_1\mathrm{B}^{-1}\mathrm{W}\mathbf{w} = \mathrm{A}_1\mathbf{x} - \mathrm{A}_1\bar{\mathbf{x}} . \tag{36}$$

This can be rewritten by substituting $\mathrm{A}_1\mathbf{x} = \mathbf{y}_1$ from Eq.(27) into the above equation.

$$\mathrm{C}\mathbf{w} = \mathbf{y}_1 - \mathrm{A}_1\bar{\mathbf{x}} , \tag{37}$$

where

$$\mathrm{C} = \mathrm{A}_1\mathrm{B}^{-1}\mathrm{W} , \quad \det\left(\mathrm{C}\right) = \det\left(\mathrm{A}\right)\det\left(\mathrm{W}_1\right)/\det\left(\mathrm{B}\right) \neq 0 , \tag{38}$$

is the influence matrix with the size of $m \times m$. The detailed formula of this matrix is shown in the next subsection.

The solution process of the IM method is summarized as follows.

1. Preparation of the influence matrix based on Eq.(38), i.e. $\mathrm{C} = \mathrm{A}_1\mathrm{B}^{-1}\mathrm{W}$, and decomposing C into LU form.
2. Solve $\mathrm{B}\bar{\mathbf{x}} = \bar{\mathbf{y}}$, Eq.(28), for $\bar{\mathbf{x}}$.
3. Solve $\mathrm{C}\mathbf{w} = \mathbf{y}_1 - \mathrm{A}_1\bar{\mathbf{x}}$, Eq.(37), for $\mathbf{w}$.
4. Solve $\mathrm{B}\mathbf{x} = \bar{\mathbf{y}} + \mathrm{W}\mathbf{w}$, Eq.(32), for $\mathbf{x}$.

As can be seen in the above process, the A problem, to which fast solvers cannot be applied, can be alternatively solved by solving two of the B problems by using fast solvers (2 and 4) and solving Eq.(37) with LU-decomposed C (3).

2.1.6 Formula of the matrix C

In this subsection, the detailed procedure of developing C is shown. First of all, Eq.(38) is rewritten as follows:

$$\mathrm{C} = \mathrm{A}_1\tilde{\mathrm{X}} , \tag{39}$$

where $\tilde{\mathrm{X}} = \mathrm{B}^{-1}\mathrm{W}$. Also, defining vectors $\mathbf{w}_m$ and $\tilde{\mathbf{x}}_m$ $(m = 1, 2, \ldots, k+d-1)$ which comprise W and $\tilde{\mathrm{X}}$ matrices, respectively, i.e.

$$\mathrm{W} = \left[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{k+d-1}\right] , \tag{40}$$

$$\tilde{\mathrm{X}} = \left[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{k+d-1}\right] , \tag{41}$$

then Eq.(39) is expressed as

$$\mathrm{C} = \mathrm{A}_1\left[\tilde{\mathbf{x}}_1, \tilde{\mathbf{x}}_2, \ldots, \tilde{\mathbf{x}}_{k+d-1}\right] . \tag{42}$$

Since $\tilde{\mathrm{X}} = \mathrm{B}^{-1}\mathrm{W} \rightarrow \mathrm{B}\tilde{\mathrm{X}} = \mathrm{W}$, $\tilde{\mathbf{x}}_m$ can be obtained by solving a set of the B problems of

$$\mathrm{B}\tilde{\mathbf{x}}_m = \mathbf{w}_m , \tag{43}$$

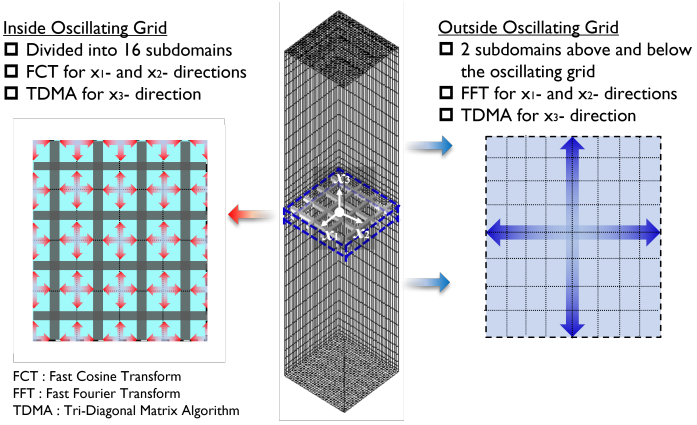


Figure 4: Fast solvers applied for the subdomains.

Table 1: CPU time (sec) for the present direct solver and BiCGSTAB.

Solver	1-RK stage	Generating C
DD-IM	5	65536
BiCGSTAB	24	-

where $\mathbf{w}_m$ is known by Eq.(33). The resultant matrix $\tilde{\mathrm{X}}$ by solving the above B problems is expressed as follows, where δ_{ij} is the Kronecker delta:

$$\tilde{\mathrm{X}} = \left[\begin{array}{c|c|c|c|cc} \mathrm{I} & \mathrm{O} & \tilde{\mathbf{b}}_j & \tilde{\mathbf{p}}' & \tilde{\mathbf{p}} & \tilde{p}_d \\ \hline \mathrm{O} & \mathrm{I} & \mathrm{O} & \mathrm{O} & \mathbf{u}\delta_{ij} & 0 \end{array}\right]^{\mathrm{T}} . \tag{44}$$

Finally, substituting the above matrix and A_1 into Eq.(39) gives the influence matrix C as follows:

$$\mathrm{C} = \left[\begin{array}{c|c} \mathrm{I} + \sum_{j=1}^{d} \mathrm{H}_j \tilde{\mathrm{p}}_j & \sum_{j=1}^{d-1} \mathrm{H}_j \mathbf{u}_j \delta_{ij} \\ \hline \tilde{\mathbf{b}}_j & \mathrm{O} \end{array}\right] . \tag{45}$$

3 Results

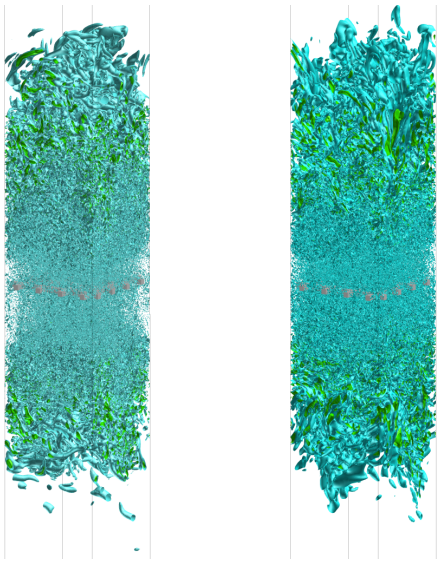
The DNSs of OGT for the system at rest were performed by using the present direct solution and BiCGSTAB [6], and the computational speeds for these cases were compared. Fast direct solvers applied for the subdomains are illustrated in Figure 4. Table 1 shows average CPU times per single RK step for the cases using the direct and iterative methods. This table reveals that the present direct method is about five times faster than BiCGSTAB. For the direct method, an influence matrix has to be preliminary generated before the initiation of simulation. This requires about 65500 sec which is almost the same as 2730 steps of BiCGSTAB.

Next, the effect of spatial resolution on the simulation results are investigated. For sake of comparison, an additional DNS of the system at rest was performed with the computational grid of $300 \times 300 \times 1900$. Figure 5 shows the isosurfaces of the second invariant of velocity gradient tensor, Q, of $Q = 0.01$ (light blue) and 0.04 (light green) of the system at rest obtained by the moderate (Figure5(a)) and high (Figure5(b)) spatial resolutions, where Q value is defined as follows:

$$Q = \frac{1}{2}(W_{ij}W_{ij} - S_{ij}S_{ij}) = -\frac{1}{2}\frac{\partial u_i}{\partial x_j}\frac{\partial u_j}{\partial x_i} . \tag{46}$$

S_{ij} and W_{ij} in the above equation are the symmetric and antisymmetric components of the velocity gradient tensor, respectively. As can be seen in this figure, for both results, fine and coarse vortices exist in the vicinity of and away from the oscillating grids, respectively. This is mainly caused by turbulence diffusion, indicating that the simulation is adequately conducted.

Comparison of it with Figure 5(a) shows a good qualitative correspondence between them, while the result of the higher spatial resolution capture finer vortices especially in the vicinity of the oscillating grid. This indicates that the DNS with the higher spatial resolution is more adequate. In this study, however, the rest of the DNSs results were of the moderate spatial resolution of $200 \times 200 \times 1400$ since our preliminary results show that the effect of the spatial resolution on statistics is approximately negligible (not shown).



(a) (b)

Figure 5: Vortex structures obtained from the present DNS of OGT using computational grid of (a) $200 \times 200 \times 1400$ and (b) $300 \times 300 \times 1900$.

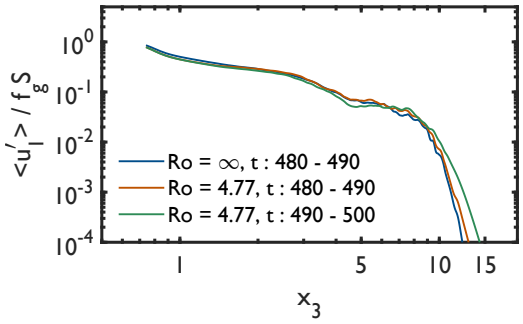


Figure 6: Velocity fluctuation intensity for the systems at rest and with rotation obtained from the present DNS.

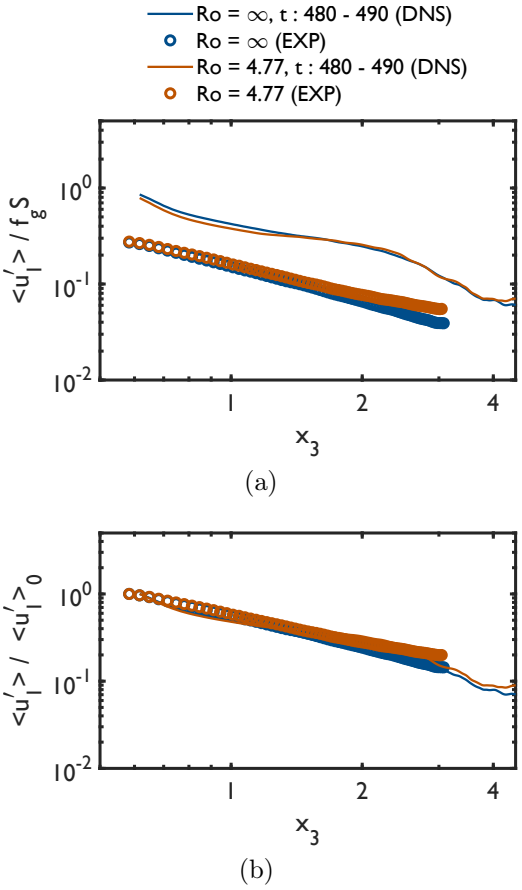


Figure 7: Comparison of velocity fluctuation intensity between the present DNS and experiment for the systems at rest: $\langle u_1' \rangle$ is normalized by (a) $f_g S$ and (b) $\langle u_1' \rangle_0$.

Here, the effect of system rotation on the turbulence intensity in the horizontal direction, $\langle u_1' \rangle$, is evaluated. Figure 6 shows the vertical distribution of turbulence intensity for both the systems at rest and with rotation. The values are averaged over the ranges $480 \leq t \leq 490$ for the system at rest, while these are averaged over the two different ranges $480 \leq t \leq 490$ and $490 \leq t \leq 500$ for the system with rotation. From this figure, it can be seen that, regardless of the presence of rotation, the trend of the decrease in $\langle u_1' \rangle$ is gradual at $x_3 < 10$ and strong at $x_3 > 10$. Also, the values in the vicinity of the oscillating grid are approximately identical for all the cases. On the other hand, for the system with rotation, the turbulence intensity remains higher than those at rest in the region away from the grid. This indicates that the effect of adding rotation becomes more pronounced beyond $x_3 > 10$ for $Ro = 4.77$.

It should be noted that the velocity field of the system with rotation has not reached a statistically steady state yet as shown in Figure 6 demonstrating that $\langle u_1' \rangle$ for $x_3 > 10$ increases as time progresses. Thus, a thorough discussion will be made in future study once the system reaches the steady state.

Figure 7(a) shows the vertical distribution of velocity fluctuation intensity in the present DNS and experiment. The experiments were performed by using the PIV technique [7]. The experimental conditions are approximately coincide with the DNS, where water was used as working fluid. From this figure, the trend of both the DNS and experimental results are approximately the same. Comparing the quantities of the $\langle u_1' \rangle$, it can be observed that the simulation results are larger than the experimental ones for both systems over the entire x_3 range. This discrepancy is possibly caused by the difference in

turbulence generated by the oscillating grid attributed to, for example, the difference in the oscillation motion of the grid in the experiment, where it can be assumed that the oscillation is not perfectly sine wave, and experimental conditions, such as non-periodic boundaries.

In order to minimize these effects, the fluctuation intensity is normalized by the value at the top dead point to which the oscillating grid can reach, $\langle u_1' \rangle_0$, as shown in Figure 7(b). This figure demonstrates that the numerical and experimental results collapse into each other for both systems. This supports our consideration that the discrepancy between the DNS and the experiment is caused by the degree of turbulence generated by the oscillating grid as well as that the fundamental flow behavior is successfully reproduced by the present DNS.

4 Conclusions

This research performed a set of DNSs of oscillating grid turbulence (OGT) using a direct Poisson solver that combines domain decomposition and influence matrix methods.

By decomposing the domain into smaller subproblems and using the influence matrix method, direct solvers are able to be applied even to the problem with complex computational geometry. Our results showed that this method significantly reduced computation times compared to the BiCGSTAB iterative solver while maintaining or improving accuracy.

Validation against experimental data showed that the DNS results accurately replicate the vertical distribution of turbulence intensity in the systems at rest and with rotation. The influence of rotation becomes more pronounced beyond $x_3 > 10$, far from the oscillating grid, highlighting the solver's ability to capture complex flow dynamics.

The ongoing progression observed in the simulation of the system with rotation suggests that further time advancement is necessary to achieve statistical steady states. This highlights areas for future improvement and optimization.

Further details about the flow structures of the OGT and the rotational effects can be seen in our another study [9] in which the proper orthogonal decomposition (POD) and the dynamic mode decomposition (DMD) are applied to the DNS data to analyze the structures.

In conclusion, this study demonstrated the potential of the combined domain decomposition and influence matrix method for advancing DNS of turbulent flows. The method's efficiency and accuracy made it a valuable tool for detailed and extensive studies of complex fluid dynamics.

This research is financially supported by the JSPS KAKENHI Grant Number 24K00802.

References

[1] C. A. J. Fletcher. *Computational Techniques for Fluid Dynamics 1: Fundamental and General Techniques*. Springer-Verlag, Berlin, Heidelberg, 1988.

[2] F. S. Godeferd and F. Moisy. Structure and dynamics of rotating turbulence: a review of recent experimental and numerical results. *Applied Mechanics Reviews*, 67(3):030802, 2015.

[3] U. Schumann and J. Benner. Direct solution of the discretized poisson—neumann problem on a domain composed of rectangles. *Journal of Computational Physics*, 46(1):1–14, 1982.

[4] D. Wilhelm. *Numerical investigation of three-dimensional separation in a forward-facing step flow using a spectral element method*. Doctoral dissertation, ETH Zürich, 2000.

[5] O. Shishkina, A. Shishkin, and C. Wagner. Simulation of turbulent thermal convection in complicated domains. *Journal of Computational and Applied Mathematics*, 226(2):336–344, 2009.

[6] H. A. Van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 13(2):631–644, 1992.

[7] K. Uchida, S. Shibata, T. Yamada, and Y. Morinishi. The effect of small rotation on turbulence structure in oscillating grid turbulence. In *Proceedings of The ASME-JSME-KSME Joint Fluids Engineering Conference (AJKFED2023)*, Osaka, Japan, 4-05-1-01, July 2023.

[8] J. H. Williamson. Low-storage runge-kutta schemes. *Journal of Computational Physics*, 35(1):48–56, 1980.

[9] R. Sumi, T. Yamada, and Y. Morinishi. Mode analysis using dns data of oscillating grid turbulence subjected to system rotation. In *Proceedings of the Twelfth International Conference on Computational Fluid Dynamics (ICCFD12)*, Kobe, Japan, 10-B-02, July 2024.

[10] A. A. Amsden and F. H. Harlow. A simplified mac technique for incompressible fluid flow calculations. *Journal of Computational Physics*, 6(2):322–325, 1970.