HOKKAIDO UNIVERSITY

Title	Accurate numerical method to solve flux distribution of Poisson's equation
Author(s)	Hirokami, Arata; Heshmat, Samia; Tomioka, Satoshi
Citation	Mathematics and computers in simulation, 190, 329-342 https://doi.org/10.1016/j.matcom.2021.05.028
Issue Date	2021-12
Doc URL	https://hdl.handle.net/2115/89321
Rights	© <2021>. This manuscript version is made available under the CC-BY-NC-ND 4.0 license http://creativecommons.org/licenses/by-nc-nd/4.0/
Rights(URL)	https://creativecommons.org/licenses/by-nc-nd/4.0/
Type	journal article
File Information	DFM.pdf

Hokkaido University Collection of Scholarly and Academic Papers : HUSCAP

Highlights

Accurate Numerical Method to Solve Flux Distribution of Poisson's Equation

Arata Hirokami, Samia Heshmat, Satoshi Tomioka

- Proposed DFM can solve fluxes with high accuracy.
- Proposed FastDFM makes DFM faster by reducing variables.
- DFM is slower than FDM and FVM but more accurate.
- FastDFM solves fluxes as accurately as DFM and almost as fast as FDM and FVM.

Accurate Numerical Method to Solve Flux Distribution of Poisson's Equation

Arata Hirokami[a,*], Samia Heshmat[b], Satoshi Tomioka[c]

[a] *Graduate School of Engineering, Hokkaido University, Sapporo-shi, 060-8628, Japan*
[b] *Faculty of Engineering, Aswan University, Aswan, 81542, Egypt*
[c] *Faculty of Engineering, Hokkaido University, Sapporo-shi, 060-8628, Japan*

Abstract

This paper proposes an accurate numerical method, the direct flux method (DFM), to solve fluxes directly for Poisson's equation. In DFM, fluxes are the variables to be solved in the system equations, where a flux is defined as an integral of the flux density across a certain finite-sized cross section. The system equation of the DFM is derived from two equations: an integral form of Poisson's equation obtained by using Gauss's divergence theorem and an integral form of the rotation-free nature of any scalar field from Stokes' theorem. In the numerical approach, no discretization error arises from Gauss's divergence theorem because it is represented as a sum of fluxes. Therefore, the discretization error is caused only by the integral form of the rotation-free nature. From the comparison between DFM, the finite difference method (FDM), and the finite volume method (FVM), we show that the accuracy of DFM is superior to that of FDM and FVM. However, DFM generally requires larger computational resources than other methods because the number of equations in DFM is more than that in other methods. To overcome this drawback, we also propose a faster algorithm than DFM, called FastDFM, which can reduce the number of equations without changing the accuracy. Hence, the proposed FastDFM produces results with the same accuracy as the DFM and with computation time almost the same as that of FDM and FVM.

Keywords: flux, Poisson's equation, Gauss's divergence theorem, Stokes' theorem, rotation free nature

---

[*] Corresponding author

Email addresses: `hirokami.arata.w0@elms.hokudai.ac.jp` (Arata Hirokami), `samia.heshmat@aswu.edu.eg` (Samia Heshmat), `tom@qe.eng.hokudai.ac.jp` (Satoshi Tomioka)

1. Introduction

It is well known that boundary value problems are used in various fields of science and engineering[8]. In most cases, these problems cannot be solved analytically. Therefore, numerical methods have been studied extensively for many years to derive accurate and fast solutions for these boundary value problems.

The methods to solve the boundary value problems are mainly classified into two groups. The first group of methods uses a mesh, such as the finite difference method (FDM)[15, 21], finite volume method (FVM)[5], and finite element method (FEM)[10]. The second group that does not use the mesh includes the boundary element method (BEM)[2, 13], the method of fundamental solutions (MFS)[9, 6], and and spectral methods[14].

Both groups generally solve the distribution of either scalar potential (e.g., velocity potential, electric potential), vector potential (e.g., magnetic potential), or flux density (e.g., mass flux density, electric flux density). These are quantities per unit volume or area, which are useful for determining the distribution of quantities in the volume or area of interest. However, another accurate quantity called flux is sometimes required in product design. The flux is the total amount of flow crossing a particular cross-section in unit time, such as mass/volume flow, electric current, heat flow, and magnetic flux. It is noted that the flux is sometimes defined as the flux density in transport phenomena. However, in this paper, we refer to the flux density as the value of the quantity per unit cross section and the flux as the area integral of the flux density over the cross section.

Table 1: Examples of fluxes and flux densities

flow	flux	flux density
mass (kg)	mass flow rate (kg/s)	mass flux (kg/($\mathrm{m}^2 \cdot \mathrm{s}$))
volume (m^3)	volume flow rate (m^3/s)	velocity (m/s)
heat (J)	heat flow ($\mathrm{J\,s^{-1}}$)	heat flux (J/($\mathrm{m}^2 \cdot \mathrm{s}$))
electric charge (C)	current (A)	current density (A/m^2)
	electric flux (C)	electric flux density (C/m^2)
	magnetic flux (Wb)	magnetic flux density (Wb/m^2)

Table 1 shows examples of fluxes and flux densities. In hydraulic engineering, the flux is used to design a relation between the rotation speed of pumps and flow rates in radial flow pumps[11] or gear pumps[12] and valves[20]. In applications of electromagnetism, magnetic flux is important because its time derivative represents the electromotive force of electric induction, which is used in many products such as electrical generators, electrical transformers, and induction motors[18]. Furthermore, heat flow[1] and electric current[17] are also important in engineering.

However, there are no methods to solve fluxes directly. In the general approach, the flux is obtained by several steps. The potential is obtained first. Then, the flux density is obtained by taking the gradients of the potential, and finally the flux is evaluated by taking the integral of the flux density. In this approach, the accuracy becomes worse because an additional error other than the error of the potential is included. It arises from the evaluation step of flux from potentials. If the flux were evaluated directly, these errors could be reduced. To reduce this error, this paper proposes a new method, the direct flux method (DFM), which takes the flux itself as a variable, and directly solves the flux without the evaluations of the potential and the flux density.

We formulate the DFM by focusing on Poisson's equation with Dirichlet or Neumann boundary conditions, which are defined as

$$\nabla \cdot \nabla u = f, \tag{1}$$

where u is a scalar potential and f is a given source term. The flux density is considered as $\nabla u \cdot \boldsymbol{n}$, where $\boldsymbol{n}$ represents a normal unit vector of a cross section of interest, and the flux is the area integral of the flux density. The purpose of this paper is to introduce the DFM algorithm to solve the flux distribution in Poisson's equation directly and to demonstrate its validity through a comparison of the accuracy with the FDM. However, DFM requires more computational effort than the other methods, such as FDM, as the number of variables of the fluxes in a system equation of DFM is almost twice and thrice that of the other methods in two-dimensional (2-D) problems and three-dimensional (3-D) problems, respectively. To accelerate DFM, a technique called FastDFM is proposed in this paper to reduce the number of variables.

There is a method similar to DFM, which takes the integral of flux density, called 'Finite Integration Technique (FIT)[19, 3]'. FIT can solve Maxwell's equations in the time domain, which is similar to the well-known finite-difference time-domain (FDTD) method[16]. However, the integral domains of FIT and DFM are different. FIT takes the line integral of the flux density along an edge of a 3-D unit cell as a variable, whereas DFM integrates over the cross section of the unit cell. Moreover, FIT does not consider higher order derivatives, whereas DFM does.

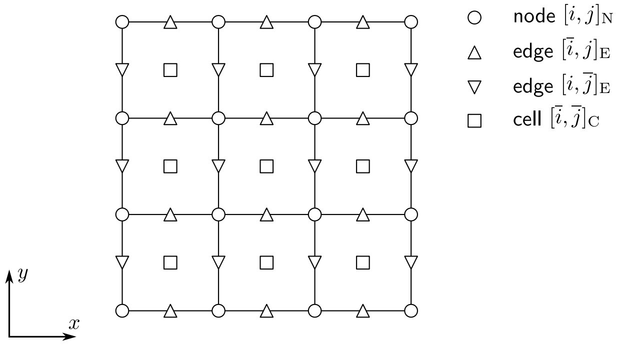

Figure 1: Nodes, edges and cells of a grid

This paper presents the basic theory and characteristics of DFM for the 2-D Poisson's equation. The outline of the paper is as follows. The grid definition used in this study is described in Section 2. Section 3 presents the mathematical theory of DFM in a square grid. Section 4 explains the acceleration of DFM using vector potential (FastDFM). Section 5 describes the accuracy and efficiency of DFM by comparing the results of DFM with that of FDM. Finally, the conclusions are presented in Section 6.

2. Grid definition

In this study, we use a square grid system with spacing Δ. For simplicity, the coordinates of a point $(x, y) = (\xi\Delta, \eta\Delta)$ is denoted as $[\xi, \eta]$, where the variables in the square bracket represent that the coordinates are normalized by Δ.

Each intersection of the grid is referred to as a node. The node at the i-th column from the left-hand side and the j-th row from the bottom of the grid is denoted as node $[i, j]_\mathrm{N}$. It should be noted that the expressions of $[i, j]$ and $[i, j]_\mathrm{N}$ are distinguished by the coordinate or the node number, although the coordinates of node $[i, j]_\mathrm{N}$ are $[i, j]$. Therefore, i and j in $[i, j]_\mathrm{N}$ do not take values other than integer values.

Each segment between adjacent nodes along the x- or y- direction is defined as an edge. The edge connecting nodes between $[i, j]_\mathrm{N}$ and $[i + 1, j]_\mathrm{N}$ is denoted as edge $\left[\bar{i}, j\right]_\mathrm{E}$. Similarly, the edge between $[i, j]_\mathrm{N}$ and $[i, j + 1]_\mathrm{N}$ is denoted as edge $\left[i, \bar{j}\right]_\mathrm{E}$. The overlined symbol "$\bar{\ }$" represents that the coordinate of the variable with the overline is varying along the edge.

Each area surrounded by four edges is defined as a cell, which is denoted by using their node number on the bottom left such as $\left[\bar{i},\bar{j}\right]_{\mathrm{C}}$. Figure 1 shows a representation of the grid, where circular, triangular, and square symbols represent nodes, edges, and cells, respectively.

The integrals along the edges and the integral over the cell are denoted by using the following notations.

$$\int_{\left[\bar{i},j\right]_{\mathrm{E}}} F\,ds = \int_{[i,j]}^{[i+1,j]} F\,dx, \tag{2}$$

$$\int_{\left[i,\bar{j}\right]_{\mathrm{E}}} F\,ds = \int_{[i,j]}^{[i,j+1]} F\,dy, \tag{3}$$

$$\int_{\left[\bar{i},\bar{j}\right]_{\mathrm{C}}} F\,dA = \int_{[i,j]}^{[i,j+1]} \int_{[i,j]}^{[i+1,j]} F\,dx\,dy. \tag{4}$$

3. Formulation of DFM

In this section, the formulation of the 2-D DFM for Poisson's equation (Eq.(1)) using Gauss's divergence theorem and the rotation-free nature of the scalar field in Stokes' theorem is presented.

Although the flux in 3-D problems is defined as the area integral of $\nabla u \cdot \boldsymbol{n}$, the flux in 2-D problems is defined as a line integral as follows:

$$\psi_L = \int_L \nabla u \cdot \boldsymbol{n}\,ds, \tag{5}$$

where L is an integral path directed in the x- or y- direction. There are two directions of $\boldsymbol{n}$ for each direction. For simplicity, $\boldsymbol{n}$ directs to the positive direction of 2-D Cartesian coordinates, that is, $\boldsymbol{n} = +\boldsymbol{e}_x$ or $\boldsymbol{n} = +\boldsymbol{e}_y$. The variables in DFM are the fluxes on the edges, $\left[\bar{i}, j\right]_{\mathrm{E}}$ and $\left[i, \bar{j}\right]_{\mathrm{E}}$, which are represented as

$$\psi_{\left[\bar{i},j\right]_{\mathrm{E}}} = \int_{\left[\bar{i},j\right]_{\mathrm{E}}} \nabla u \cdot \boldsymbol{n}\,ds = \int_{[i,j]}^{[i+1,j]} \nabla u \cdot \boldsymbol{e}_y\,dx, \tag{6a}$$

$$\psi_{\left[i,\bar{j}\right]_{\mathrm{E}}} = \int_{\left[i,\bar{j}\right]_{\mathrm{E}}} \nabla u \cdot \boldsymbol{n}\,ds = \int_{[i,j]}^{[i,j+1]} \nabla u \cdot \boldsymbol{e}_x\,dy. \tag{6b}$$

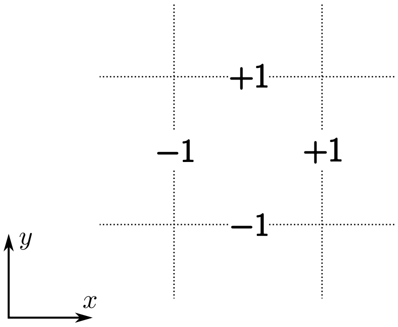

Figure 2: DFM's stencil by Gauss' equation

DFM utilizes two mathematical formulas. One is an integral form of Poisson's equation, which is obtained by applying the Gauss's divergence theorem to the 2-D Poisson's equation.

$$\oint_{\partial A} \nabla u \cdot \boldsymbol{n} \, ds = \int_A f \, dA, \tag{7}$$

where A is any area, and ∂A is the boundary of A. The other formula is an integral form of rotation-free nature, which is obtained by applying the Stokes' theorem to the vector identity of $\nabla \times \nabla u = 0$.

$$\oint_C \nabla u \cdot \boldsymbol{\tau} \, ds = 0, \tag{8}$$

where C is an arbitrary closed path, and $\boldsymbol{\tau}$ denotes the tangential unit vector along C.

3.1. Discretization of Gauss's divergence theorem

When A in Eq.(7) is taken as a cell, the left-hand side of Eq.(7) is simply discretized as follows:

$$\psi_{[i+1,\bar{j}]_\mathrm{E}} + \psi_{[\bar{i},j+1]_\mathrm{E}} - \psi_{[i,\bar{j}]_\mathrm{E}} - \psi_{[\bar{i},j]_\mathrm{E}} = \int_{[\bar{i},\bar{j}]_\mathrm{C}} f \, dA. \tag{9}$$

Figure 2 represents the stencil of Eq.(9), where the values on the edges are the coefficients of the flux on each edge. It should be noted that this equation does not contain any error.

The number of equations obtained by the Gauss's divergence theorem is equal to the number of cells. The number of variables, ψ, is equal to the number of edges, which is more than that of cells in general. Because the number of variables is less than the number of equations, the simultaneous equations cannot be solved.

To address this problem, other equations based on the rotation-free nature of the scalar field in Stokes' theorem, Eq.(8), are added; their details are shown in the next subsection.

3.2. Discretization of Stokes' theorem

Let the closed path C of Eq.(8) be a rectangular path $C_{[i,j]_\mathrm{N}}$ centered on nodes, $[i,j]_\mathrm{N}$ as described below.

$$
\begin{aligned}
C_{[i,j]_\mathrm{N}} &: C^{\{1\}}_{[i,j]_\mathrm{N}} \longrightarrow C^{\{2\}}_{[i,j]_\mathrm{N}} \longrightarrow C^{\{3\}}_{[i,j]_\mathrm{N}} \longrightarrow C^{\{4\}}_{[i,j]_\mathrm{N}}, \\
C^{\{1\}}_{[i,j]_\mathrm{N}} &: [i+\xi, j-\eta] \longrightarrow [i+\xi, j+\eta], \\
C^{\{2\}}_{[i,j]_\mathrm{N}} &: [i+\xi, j+\eta] \longrightarrow [i-\xi, j+\eta], \\
C^{\{3\}}_{[i,j]_\mathrm{N}} &: [i-\xi, j+\eta] \longrightarrow [i-\xi, j-\eta], \\
C^{\{4\}}_{[i,j]_\mathrm{N}} &: [i-\xi, j-\eta] \longrightarrow [i+\xi, j-\eta].
\end{aligned}
$$

If ξ and η are between 0 and 1, this path is across the four edges connected to the node $[i,j]_\mathrm{N}$. In the Stokes' theorem, as shown in Eq.(8), the tangential component of ∇u is integrated along the path $C_{[i,j]_\mathrm{N}}$, the direction of which is normal to the edges, where each direction is the same as that of the flux. However, it crosses the edge at a point on the edge, and it is not integrated along the edges. Therefore, the integral shown in Eq.(8) does not include the flux. To express the fluxes, we apply a double integral with respect to ξ and η in Eq.(8) so that the integrals of the normal components of ∇u of the edges are included.

$$
\int_0^1 \int_0^1 \oint_{C_{[i,j]_\mathrm{N}}} \nabla u \cdot \boldsymbol{\tau} \, ds \, d\xi \, d\eta = 0. \tag{10}
$$

The left-hand side of Eq.(10) is divided into the sum of four terms.

$$
\int_0^1 \int_0^1 \oint_{C_{[i,j]_\mathrm{N}}} \nabla u \cdot \boldsymbol{\tau} \, ds \, d\xi \, d\eta = \sum_{k=1}^{4} \int_0^1 \int_0^1 \int_{C^{\{k\}}_{[i,j]_\mathrm{N}}} \nabla u \cdot \boldsymbol{\tau} \, ds \, d\xi \, d\eta = \sum_{k=1}^{4} \Psi^{\{k\}}_{[i,j]_\mathrm{N}}. \tag{11}
$$

When $k = 1$, the equation can be written as

$$\begin{aligned}\Psi^{\{1\}}_{[i,j]_\mathrm{N}} &= \int_0^1 \int_0^1 \int_{C^{\{1\}}_{[i,j]_\mathrm{N}}} \nabla u \cdot \boldsymbol{\tau}\, ds\, d\xi\, d\eta \\ &= \int_0^1 \int_0^1 \int_{(j-\eta)\Delta}^{(j+\eta)\Delta} \left.\frac{\partial u}{\partial y}\right|_{x=(i+\xi)\Delta} dy\, d\xi\, d\eta \\ &= \int_0^1 \int_0^1 \int_{j-\eta}^{j+\eta} \left.\frac{\partial u}{\partial \hat{y}}\right|_{\hat{x}=i+\xi} d\hat{y}\, d\xi\, d\eta, \end{aligned} \tag{12}$$

where $\hat{x} = x/\Delta$ and $\hat{y} = y/\Delta$. By exchanging the order of integration, it can be rewritten as follows.

$$\begin{aligned}\Psi^{\{1\}}_{[i,j]_\mathrm{N}} &= \int_{j-1}^{j+1} \int_0^1 (1 - |\hat{y} - j|) \left.\frac{\partial u}{\partial \hat{y}}\right|_{\hat{x}=i+\xi} d\xi d\hat{y} \\ &= \int_{j-1}^{j+1} (1 - |\hat{y} - j|) \int_0^1 \left.\frac{\partial u}{\partial \hat{y}}\right|_{\hat{x}=i+\xi} d\xi d\hat{y}, \end{aligned} \tag{13}$$

Here, we introduce the following functions, X_j and Y_i, the dimensions of which are the same as those of fluxes.

$$Y_i(\hat{y}) = \int_0^1 \left.\frac{\partial u}{\partial \hat{y}}\right|_{\hat{x}=i+\xi} d\xi, \tag{14a}$$

$$X_j(\hat{x}) = \int_0^1 \left.\frac{\partial u}{\partial \hat{x}}\right|_{\hat{y}=j+\eta} d\eta. \tag{14b}$$

When $\hat{y}$ or $\hat{x}$ is an integer, these functions show the flux on the edge, for example, $Y_i(j) = \psi_{[\bar{i},j]_\mathrm{E}}$ and $X_j(i) = \psi_{[i,\bar{j}]_\mathrm{E}}$. By substituting Eq.(14a) to the internal integral on the right-hand side of Eq.(13), and by replacing $\hat{y} + j$ to $\hat{y}$, Eq.(13) is reduced to

$$\Psi^{\{1\}}_{[i,j]_\mathrm{N}} = \int_{-1}^{1} (1 - |\hat{y}|) Y_i(\hat{y} + j) d\hat{y}. \tag{15a}$$

Similarly, when $k = 2, 3, 4$, the following equations are obtained.

$$\Psi^{\{2\}}_{[i,j]_\mathrm{N}} = \int_{1}^{-1} (1 - |\hat{x}|) X_j(\hat{x} + i) d\hat{x}, \tag{15b}$$

$$\Psi^{\{3\}}_{[i,j]_\mathrm{N}} = \int_{1}^{-1} (1 - |\hat{y}|) Y_{i-1}(\hat{y} + j) d\hat{y}, \tag{15c}$$

$$\Psi^{\{4\}}_{[i,j]_\mathrm{N}} = \int_{-1}^{1} (1 - |\hat{x}|) X_{j-1}(\hat{x} + i) d\hat{x}. \tag{15d}$$

If these integrals are expressed by $Y_i(j)$ and $X_j(i)$, they can be represented by the fluxes.

To evaluate the integral in Eq.(15a), let us consider the Taylor series expansion around j.

$$Y_i(\hat{y}+j) = \sum_{l=0}^{\infty} \frac{1}{l!} \left. \frac{d^l Y_i}{d\hat{y}^l} \right|_{\hat{y}=j} \hat{y}^l. \tag{16}$$

Substituting this equation into Eq.(15a), we obtain

$$\Psi^{\{1\}}_{[i,j]_{\mathrm{N}}} = \sum_{l=0}^{\infty} \frac{2}{(2l+2)!} \left. \frac{d^{2l} Y_i}{d\hat{y}^{2l}} \right|_{\hat{y}=j} = \sum_{l=0}^{\infty} \frac{2\Delta^{2l}}{(2l+2)!} \left. \frac{d^{2l} Y_i}{dy^{2l}} \right|_{y=j\Delta}. \tag{17}$$

In this evaluation, the odd order terms of $\hat{y}$ in Eq.(16) are canceled because $1-|\hat{y}|$ in Eq.(15a) is an even function, and a chain rule of the derivative is applied in the translation from the second equation to the last equation so that the derivative becomes independent of Δ.

When we truncate the terms with $l \geq 1$ on the right-hand side of Eq.(17), $\Psi^{\{1\}}_{[i,j]_{\mathrm{N}}}$ is approximated to $\psi_{[i,\bar{j}]_{\mathrm{E}}}$, the error order of which is Δ^2.

$$\Psi^{\{1\}}_{[i,j]_{\mathrm{N}}} = \psi_{[\bar{i},j]_{\mathrm{E}}} + \mathcal{O}(\Delta^2). \tag{18}$$

In the case where the terms with $l \geq 2$ are truncated, the terms with the second-order derivative emerge. When the second-order derivative is expressed by a second-order central difference, $\Psi^{\{1\}}_{[i,j]_{\mathrm{N}}}$, the 4th order accuracy is represented by the fluxes on the three neighboring edges as follows.

$$\Psi^{\{1\}}_{[i,j]_{\mathrm{N}}} = \frac{1}{12} \left(\psi_{[\bar{i},j-1]_{\mathrm{E}}} + 10\psi_{[\bar{i},j]_{\mathrm{E}}} + \psi_{[\bar{i},j+1]_{\mathrm{E}}} \right) + \mathcal{O}(\Delta^4). \tag{19}$$

In the cases where $k = 2, 3, 4$, the discretization can be similarly obtained. However, the sign of each term is different from Eq.(18) or (19) because the range of the integral is different between Eq.(15a)-(15d). Therefore, after substituting $\Psi^{\{k\}}_{[i,j]_{\mathrm{N}}}$ in Eq.(10), the number of terms on the left-hand side of Eq.(10) in the 2nd and 4th order accuracies are 4 and 12, respectively.

Figure 3 shows the stencils in Eq.(18) and (19). The sum of the factors shown in Figure 3 is identical to zero in each case.

It should be noted that the above formula requires that the higher-order terms that include higher-order derivatives of u are negligibly small. If there are singular points in the cells, special treatments are required.

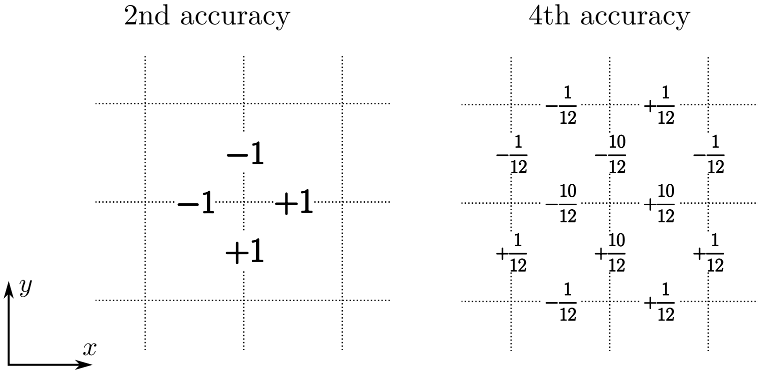

Figure 3: DFM's stencil by Stokes' equation

3.3. Boundary conditions

In this section, we formulate the Neumann and the Dirichlet boundary conditions in the DFM. For simplicity, the calculation area is a rectangle, where $\left[\bar{i}, 0\right]_{\mathrm{E}}$ are the edges on the bottom of the boundary and $\left[0, \bar{j}\right]_{\mathrm{E}}$ are the edges on the left boundary.

The Neumann boundary condition is given as follows.

$$\nabla u \cdot \boldsymbol{n} = g_1, \tag{20}$$

where $\boldsymbol{n}$ is the unit normal vector to the boundaries of the calculation domain, and g_1 is a known function defined on the boundary. This equation is formulated by integrating over the edges on the boundaries; for example, at the bottom of the boundary

$$\psi_{\left[\bar{i},0\right]_{\mathrm{E}}} = \int_{\left[\bar{i},0\right]_{\mathrm{E}}} g_1 \, dx. \tag{21}$$

In the DFM, the Neumann boundary condition can be formulated without any error.

The Dirichlet boundary condition is more complicated than the Neumann boundary condition. It is given by

$$u = g_2, \tag{22}$$

where g_2 is a known function defined on the boundaries.

First, we describe the case where $g_2 = 0$. The line integral value of the gradient ∇u along any path D is expressed as follows:

$$\int_D \nabla u \cdot \boldsymbol{\tau} ds = u(P) - u(Q), \tag{23}$$

where P and Q are the start and end points of any path D, respectively. For each node $[i,0]_{\mathrm{N}}$ at the bottom of the boundary, the path D is set as described here.

$$\begin{aligned}
&D_{[i,0]_{\mathrm{N}}} : D^{\{1\}}_{[i,0]_{\mathrm{N}}} \longrightarrow D^{\{2\}}_{[i,0]_{\mathrm{N}}} \longrightarrow D^{\{3\}}_{[i,0]_{\mathrm{N}}},\\
&D^{\{1\}}_{[i,0]_{\mathrm{N}}} : [i+\xi,0] \longrightarrow [i+\xi,\eta]\,,\\
&D^{\{2\}}_{[i,0]_{\mathrm{N}}} : [i+\xi,\eta] \longrightarrow [i-\xi,\eta]\,,\\
&D^{\{3\}}_{[i,0]_{\mathrm{N}}} : [i-\xi,\eta] \longrightarrow [i-\xi,0]\,.
\end{aligned}$$

If $g_2 = 0$, the following equation holds from Eq.(22).

$$\int_{D_{[i,0]_{\mathrm{N}}}} \nabla u \cdot \boldsymbol{\tau} ds = \sum_{k=1}^{3} \int_{D^{\{k\}}_{[i,0]_{\mathrm{N}}}} \nabla u \cdot \boldsymbol{\tau} ds = 0. \tag{24}$$

Similar to Eq.(10), this equation is integrated with respect to ξ and η as follows.

$$\sum_{k=1}^{3} \int_0^1 \int_0^1 \int_{D^{\{k\}}_{[i,0]_{\mathrm{N}}}} \nabla u \cdot \boldsymbol{\tau} ds d\xi d\eta = \sum_{k=1}^{3} \Psi^{\{k\}}_{[i,0]_{\mathrm{N}}} = 0. \tag{25}$$

Following the same process as in Eq.(10)-(15d), $\{\Psi^{\{k\}}_{[i,0]_{\mathrm{N}}} | k = 1,2,3\}$ is respectively calculated.

$$\Psi^{\{1\}}_{[i,0]_{\mathrm{N}}} = \int_0^1 (1-\hat{y}) Y_i(\hat{y}) d\hat{y}, \tag{26a}$$

$$\Psi^{\{2\}}_{[i,0]_{\mathrm{N}}} = \int_1^{-1} (1-|\hat{x}|) X_i(\hat{x}+i) d\hat{x}, \tag{26b}$$

$$\Psi^{\{3\}}_{[i,0]_{\mathrm{N}}} = \int_1^0 (1-\hat{y}) Y_{i-1}(\hat{y}) d\hat{y}. \tag{26c}$$

Moreover, using the Taylor expansion, Eq.(26b) can be discretized as Eq.(18) or Eq.(19). Equation (26a) can be discretized with 2nd-order and 4th-order accuracy as in the following equation.

$$\Psi^{\{1\}}_{[i,0]_{\mathrm{N}}} = \frac{1}{6}(2\psi_{[\bar{i},0]_{\mathrm{E}}} + \psi_{[\bar{i},1]_{\mathrm{E}}}) + \mathcal{O}(\Delta^2), \tag{27}$$

$$\Psi^{\{1\}}_{[i,0]_{\mathrm{N}}} = \frac{1}{360}(97\psi_{[\bar{i},0]_{\mathrm{E}}} + 114\psi_{[\bar{i},1]_{\mathrm{E}}} - 39\psi_{[\bar{i},2]_{\mathrm{E}}} + 8\psi_{[\bar{i},3]_{\mathrm{E}}}) + \mathcal{O}(\Delta^4). \tag{28}$$

The discretized Eq.(26c) has the same form, except the signs of all the terms are inverted.

When the node is a corner such as $[0,0]_{\mathrm{N}}$, it is necessary to define the path as follows.

$$\begin{aligned}
D_{[0,0]_{\mathrm{N}}} &: D^{\{1\}}_{[i,0]_{\mathrm{N}}} \longrightarrow D^{\{2\}}_{[i,0]_{\mathrm{N}}}, \\
D^{\{1\}}_{[0,0]_{\mathrm{N}}} &: [\xi, 0] \longrightarrow [\xi, \eta], \\
D^{\{2\}}_{[0,0]_{\mathrm{N}}} &: [\xi, \eta] \longrightarrow [0, \eta].
\end{aligned}$$

This is also calculated as follows:

$$\sum_{k=1}^{2} \int_{D^{\{k\}}_{[0,0]_{\mathrm{N}}}} \nabla u \cdot \boldsymbol{\tau} ds = \Psi^{\{k\}}_{[0,0]_{\mathrm{N}}} = 0, \tag{29}$$

where,

$$\Psi^{\{1\}}_{[0,0]_{\mathrm{N}}} = \int_0^1 (1-\hat{y}) Y_i(\hat{y}) d\hat{y}, \tag{30a}$$

$$\Psi^{\{2\}}_{[0,0]_{\mathrm{N}}} = \int_1^0 (1-\hat{x}) X_i(\hat{x}) d\hat{x}. \tag{30b}$$

Using Eq.(27) or Eq.(28), these equations can be discretized.

When $g_2 \neq 0$, the right-hand side of Eq.(24) is

$$\int_{D_{[i,0]_{\mathrm{N}}}} \nabla u \cdot \boldsymbol{\tau} ds = \sum_{k=1}^{3} \int_{D^{\{k\}}_{[i,0]_{\mathrm{N}}}} \nabla u \cdot \boldsymbol{\tau} ds = u_{[i+\xi,0]} - u_{[i-\xi,0]}. \tag{31}$$

Because the right-hand side is defined as the boundary conditions g_2, by integrating it with respect to ξ and η, we obtain

$$\begin{aligned}
\int_0^1 \int_0^1 (u_{[i+\xi,0]} - u_{[i-\xi,0]}) d\xi d\eta &= \int_0^1 (u_{[i+\xi,0]} - u_{[i-\xi,0]}) d\xi \\
&= \int_{[\bar{i},0]_{\mathrm{E}}} g_2 \, dx - \int_{[\overline{i-1},0]_{\mathrm{E}}} g_2 \, dx,
\end{aligned} \tag{32}$$

where the edges to the integral are connecting to the node $[i,0]_{\mathrm{N}}$. In the case of the corner node, we can obtain the same relation, but the indices of edges are different. Therefore, Eq.(25) for both cases can be rewritten as follows:

$$\sum_{k=1}^{N_k} \Psi^{\{k\}}_{[i,j]_{\mathrm{N}}} = g_{2,[i,j]_{\mathrm{E}+}} - g_{2,[i,j]_{\mathrm{E}-}}, \tag{33}$$

where N_k is 3 for the flat boundary and 2 for the corner, and the two edges connected to the boundary node $[i,j]_{\mathrm{N}}$ are denoted by $[i,j]_{\mathrm{E}-}$ and $[i,j]_{\mathrm{E}+}$ in the counterclockwise order.

3.4. Relationship between number of equations and number of variables

Euler's theorem of planar graphs shows that

$$v - e + f = 1, \tag{34}$$

where v, e, and f are the numbers of vertices, edges, and faces, respectively, in an arbitrary plane graph. In the 2-D DFM grid, the vertices, edges, and faces correspond to the nodes, edges, and cells, respectively.

The equations obtained from Gauss's divergence theorem are formulated for each cell, and those from Stokes' theorem are formulated for each node. The fluxes that are the variables to be solved are defined on the edges. Regarding the boundary condition, Eq.(21) can be formulated for each edge on the boundary for the Neumann boundary condition, and Eq.(33) can be formulated for each node on the boundary for the Dirichlet boundary condition. However, the number of expressions is the same in both cases as the number of nodes on the boundary and the number of edges on the boundary are the same. Therefore, the number of unknown variables, V, is reduced to $V = e - N_{\mathrm{be}}$, where N_{be} represents the number of the boundary edges. At the nodes on the boundary, the equations based on Stokes' theorem cannot be obtained because the equation requires undefined fluxes outside the boundary. As a result, the number of equations based on Stokes' theorem, E_{S}, is reduced to $E_{\mathrm{S}} = v - N_{\mathrm{bn}}$, where N_{bn} is the number of nodes on the boundary. In contrast, the number of equations based on Gauss's divergence theorem, E_{G}, is not reduced, that is, $E_{\mathrm{G}} = f$. Moreover, as the boundary is a closed path, the numbers of nodes and edges are the same, that is, $N_{\mathrm{bn}} = N_{\mathrm{be}}$. By eliminating v, e, f, N_{bn}, and N_{be} from these equations, Eq.(34) can be rewritten as

$$E_{\mathrm{G}} + E_{\mathrm{S}} - 1 = V. \tag{35}$$

For example, for the 3×3 grid shown in Figure 1, $E_{\mathrm{G}} = 9$, $E_{\mathrm{S}} = 4$, and $V = 12$.

Eq.(35) indicates that there is a redundant equation in the set of DFM equations. When all boundaries are the Neumann boundary conditions, one equation derived from Gauss's equation Eq.(9) needs to be removed; otherwise, when the Dirichlet boundary conditions are contained, one equation derived from Stokes' equation Eq.(10) needs to be removed.

4. Acceleration of DFM

DFM handles unknown variables on edges, while FDM, FEM, and FVM handle unknowns on nodes or cells. Therefore, because the number of edges is about twice the number of nodes or cells in 2-D problems, DFM consumes more computational time than the other methods. To reduce the number of unknowns, we propose a method, FastDFM, in which the flux on the edge is represented by variables on the nodes.

Helmholtz's theorem states that any vector field $\boldsymbol{F}$ in 3-D space can be resolved into the sum of a rotation-free vector field, ∇u, and a divergence-free vector field, $\nabla \times \boldsymbol{v}$; namely,

$$\boldsymbol{F} = \nabla u + \nabla \times \boldsymbol{v}, \tag{36}$$

Here, we consider that u satisfies Poisson's equation shown in Eq.(1). Substituting ∇u obtained from Eq.(36) to (7), we obtain the following equation.

$$\oint_{\partial A} \nabla u \cdot \boldsymbol{n}\, ds = \oint_{\partial A} (\boldsymbol{F} - \nabla \times \boldsymbol{v}) \cdot \boldsymbol{n}\, ds = \int_A f\, dA. \tag{37}$$

The second term in the second equation vanishes because this term is a result of an integral of $\nabla \cdot \nabla \times \boldsymbol{v}$ which is identical to 0, that is,

$$\oint_{\partial A} \nabla \times \boldsymbol{v} \cdot \boldsymbol{n}\, ds = 0. \tag{38}$$

However, this integral is useful for representing the flux using nodal variables. When the area A is a unit cell, the closed path is divided into four edges as follows:

$$\oint_{\partial A} \nabla \times \boldsymbol{v} \cdot \boldsymbol{n}\, ds = \sum_{k=1}^{4} \int_{\partial A^{\{k\}}} \nabla \times \boldsymbol{v} \cdot \boldsymbol{n}\, ds, \tag{39}$$

where $\{\partial A^{\{k\}} | k = 1, 2, 3, 4\}$ are the four edges surrounding A. Here, we define an indefinite integral of $\nabla \times \boldsymbol{v} \cdot \boldsymbol{n}$ as

$$\chi = \int \nabla \times \boldsymbol{v} \cdot \boldsymbol{n}\, ds. \tag{40}$$

By using χ, the definite integral along each edge is represented by nodal values as follows:

$$\int_{\partial A^{\{k\}}} \nabla \times \boldsymbol{v} \cdot \boldsymbol{n}\, ds = \chi^{\{k,2\}} - \chi^{\{k,1\}}, \tag{41}$$

where the superscripts of χ, $\{k,1\}$, and $\{k,2\}$ are the beginning and end points of $\partial A^{\{k\}}$, respectively. In contrast to $\nabla\times\boldsymbol{v}\cdot\boldsymbol{n}$, the discretized integral of $\boldsymbol{F}\cdot\boldsymbol{n}$ is defined as ψ^* as follows.

$$\psi^*_{\partial A^{\{k\}}} \equiv \int_{\partial A^{\{k\}}} \boldsymbol{F}\cdot\boldsymbol{n}\, ds, \tag{42}$$

is defined as a value related to the edge. From the relation between the second and last equations of Eq.(37), we can obtain the following requirement for ψ^* by using Eq.(38).

$$\sum_{k=1}^{4} \psi^*_{\partial A^{\{k\}}} = \int_A f\, dA. \tag{43}$$

By using these definitions, the integral along the edge, $\partial A^{\{k\}}$, is rewritten as

$$\psi_{\partial A^{\{k\}}} = \psi^*_{\partial A^{\{k\}}} + \left(\chi^{\{k,2\}} - \chi^{\{k,1\}}\right). \tag{44}$$

This equation shows that the flux on a certain edge is divided into two terms: 1) ψ^* defined on the edge and 2) the difference in the nodal potentials of χ. Even when ψ^* is a particular solution that satisfies Eq.(9) but does not satisfy Eq.(10) and the boundary conditions, the true flux, ψ, is fixed by tuning χ such that ψ satisfies Eq.(10) and the boundary conditions.

In other words, the fluxes $\psi_{[\bar{i},j]_\mathrm{E}}$, $\psi_{[i,\bar{j}]_\mathrm{E}}$ are replaced as follows.

$$\psi_{[\bar{i},j]_\mathrm{E}} = \chi_{[i+1,j]_\mathrm{N}} - \chi_{[i,j]_\mathrm{N}} + {\psi^*}_{[\bar{i},j]_\mathrm{E}}, \tag{45a}$$

$$\psi_{[i,\bar{j}]_\mathrm{E}} = \chi_{[i,j]_\mathrm{N}} - \chi_{[i,j+1]_\mathrm{N}} + {\psi^*}_{[i,\bar{j}]_\mathrm{E}}, \tag{45b}$$

where $\chi_{[i,j]_\mathrm{N}}$ is χ at the node $[i,j]_\mathrm{N}$, and ${\psi^*}_{[\bar{i},j]_\mathrm{E}}$ and ${\psi^*}_{[i,\bar{j}]_\mathrm{E}}$ are ψ^* on the edge $\left[\bar{i},j\right]_\mathrm{E}$ and $\left[\bar{i},j\right]_\mathrm{E}$. Then, Eq.(45a) and Eq.(45b) are substituted for Eq.(18) or Eq.(19) derived from the rotation-free nature. For example, 2nd-order accuracy:

$$\begin{aligned}\chi_{[i+1,j]_\mathrm{N}} + \chi_{[i,j+1]_\mathrm{N}} + \chi_{[i-1,j]_\mathrm{N}} + \chi_{[i,j-1]_\mathrm{N}} - 4\chi_{[i,j]_\mathrm{N}} \\ = -({\psi^*}_{[\bar{i},j]_\mathrm{E}} - {\psi^*}_{[i,\bar{j}]_\mathrm{E}} - {\psi^*}_{[\overline{i-1},j]_\mathrm{E}} + {\psi^*}_{[i,\overline{j-1}]_\mathrm{E}}).\end{aligned} \tag{46}$$

The number of equations is the total number of nodes, except those on the boundary. The 4th-order accuracy equation can be formulated in the same manner.

In addition, Eq.(45a) and Eq.(45b) are substituted for the boundary condition, that is Eq.(21), Eq.(27), or Eq.(28). The number of equations is the same as the number of nodes on the boundary. After summing them up, the number of equations is the same as the number of nodes. Thereby, if ψ^* were determined as known values, χ is obtained by solving the simultaneous equations.

The remainder of this section describes the method used to determine the particular solution of ψ^*. By using the notation of the cell and the edges shown in section 2, Eq.(43) can be rewritten as follows.

$$\psi^*_{\left[\bar{i},j+1\right]_\mathrm{E}} + \psi^*_{\left[i+1,\bar{j}\right]_\mathrm{E}} - \psi^*_{\left[\bar{i},j\right]_\mathrm{E}} - \psi^*_{\left[i,\bar{j}\right]_\mathrm{E}} = Q_{\left[\bar{i},\bar{j}\right]_\mathrm{C}}, \tag{47}$$

$$Q_{\left[\bar{i},\bar{j}\right]_\mathrm{C}} = \int_{\left[\bar{i},\bar{j}\right]_\mathrm{C}} f\, dA, \tag{48}$$

where $Q_{\left[\bar{i},\bar{j}\right]_\mathrm{C}}$ is a known value because f is given. The system equation to determine ψ^* is composed of Eq.(47) for all cells. The number of variables, ψ^*, is more than the number of equations since the number of edges is almost twice that of the cells. This means that this problem is an underdetermined problem, and the solution cannot be determined uniquely. However, we require only one particular solution in the solutions that satisfy the system equation shown in Eq.(47). To determine the particular solution, we do not need to solve simultaneous equations. The solution can be obtained by a bucket relay scheme of sources in which the sources on the cells in the domain are moved to the edges on the boundary. In this scheme, the flux is evaluated as the accumulation of relayed sources. The algorithm where the edge of the boundary is a certain edge is as follows.

1. Set ψ^* on all edges to 0.
2. Evaluate Q for all cells and set an unprocessed mark to the cells.
3. Set an arbitrary edge as a goal edge.
4. Select a start cell from the population of the unprocessed cells.
5. Draw an arbitrary arrow from the start cell to the goal edge so that the arrow crosses the edges.
6. Add Q of the start cell to ψ^* on the edges crossing the arrow.
7. Repeat step 4 to step 6 until all cells are processed.

Figure 4 exhibits this algorithm for a simple example.

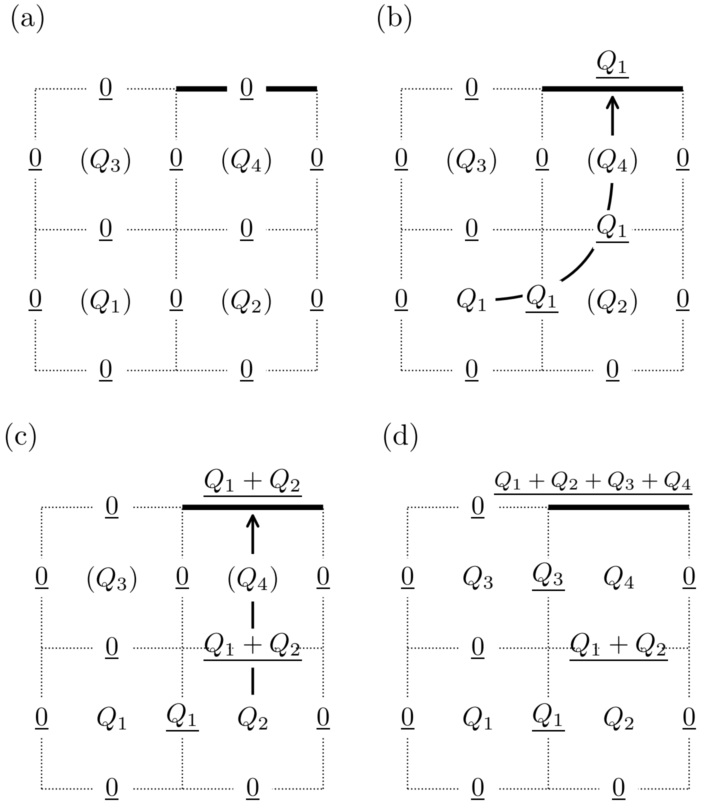

Figure 4: Algorithm to determine a particular flux distribution, ψ^*, on the edges. The thick line is set as the goal edge, and Q_k in the cell is the area integral of f over the k-th cell. The parenthesis of Q_k means that the k-th cell is not processed. The underlined value on the edge is the current value of ψ^*. (a) Initial state. (b) The case where the starting cell is the cell with Q_1. The source value of Q_1 is added to $\psi^*_{[\bar{i},j]_{\mathrm{E}}}$ on the edges that cross the arrow from the starting cell to the goal edge. By this process, the cell of Q_1 satisfies Gauss's divergence theorem. (c) The case where the starting cell is the cell with Q_2. (d) After all the cells are processed.

5. Numerical experiments

To validate of the proposed DFM and FastDFM, the numerical error and computational time of the 4th-order DFM and FastDFM were compared with those of the compact 4th-order FDM[15] and 4th-order FVM of Mehrstellen discretization[4].

To estimate the error of the flux, the following normalized root mean square errors (RMSEs), $\hat{\varepsilon}$, were evaluated.

$$\hat{\varepsilon} = \sqrt{\frac{\sum_{e\in\mathcal{E}} \left(\psi_e - \tilde{\psi}_e\right)^2}{\sum_{e\in\mathcal{E}} \tilde{\psi}_e^2}}, \tag{49}$$

where $\mathcal{E}$ is a set of all edges contained in the grid, and ψ_e and $\tilde{\psi}_e$ are the numerical and exact solutions of the flux on the edge e, respectively.

In the computation by FDM, a flux on an edge was evaluated through two steps: (1) the potentials at nodes were evaluated by FDM, and (2) the edge flux was obtained by the following 4th-order accuracy equation. This can be derived by the Taylor expansion[4].

$$\begin{aligned}\psi_{[\bar{i},j]_{\mathrm{E}}} = &\frac{1}{12}\left(u_{[i-\frac{1}{2},j+\frac{1}{2}]} - u_{[i-\frac{1}{2},j-\frac{1}{2}]} + u_{[i+\frac{3}{2},j+\frac{1}{2}]} - u_{[i+\frac{3}{2},j-\frac{1}{2}]}\right)\\ &+ \frac{10}{12}\left(u_{[i+\frac{1}{2},j+\frac{1}{2}]} - u_{[i+\frac{1}{2},j-\frac{1}{2}]}\right) - \frac{\Delta^2}{24}\int_{[\bar{i},j]_{\mathrm{E}}} \frac{\partial f}{\partial y} dx + \mathcal{O}(\Delta^4),\end{aligned} \tag{50a}$$

$$\begin{aligned}\psi_{[i,\bar{j}]_{\mathrm{E}}} = &\frac{1}{12}\left(u_{[i+\frac{1}{2},j-\frac{1}{2}]} - u_{[i-\frac{1}{2},j-\frac{1}{2}]} + u_{[i+\frac{1}{2},j+\frac{3}{2}]} - u_{[i-\frac{1}{2},j+\frac{3}{2}]}\right)\\ &+ \frac{10}{12}\left(u_{[i+\frac{1}{2},j+\frac{1}{2}]} - u_{[i-\frac{1}{2},j+\frac{1}{2}]}\right) - \frac{\Delta^2}{24}\int_{[i,\bar{j}]_{\mathrm{E}}} \frac{\partial f}{\partial x} dy + \mathcal{O}(\Delta^4).\end{aligned} \tag{50b}$$

As FVM discretizes the fluxes into potentials u by Eq.(50a) and Eq.(50b), for formulating simultaneous equations, the fluxes can be obtained with minimum error by resubstituting the obtained u into Eq.(50a) and Eq.(50b).

For solving the simultaneous equations, we used the method of LU decomposition with full pivot by Eigen (version 3.3.7)[7], which is an open source software for linear algebra written in C++. The calculation time was measured using a PC with an Intel Core i7-7700K CPU @ 4.20 GHz and 64.0 GB RAM.

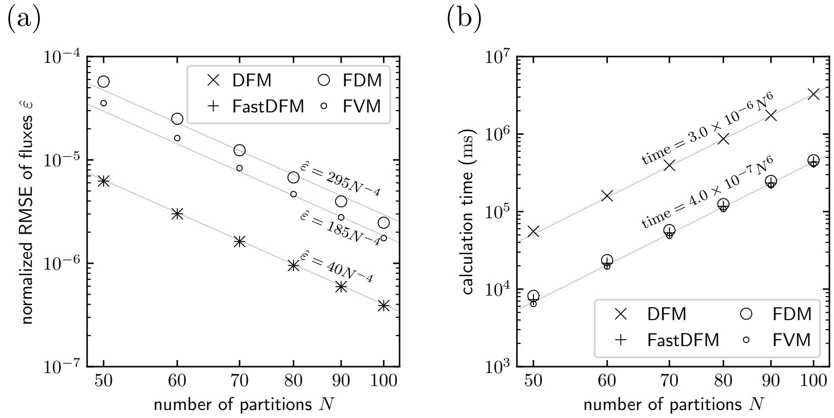

Figure 5: Comparison of the performance of DFM, FastDFM, FDM, and FVM in terms of (a) normalized RMSE and (b) calculation time in Problem 1.

5.1. Problem 1: Neumann boundary condition

The source term of Poisson's equation was set as a trigonometric function as follows,

$$\nabla \cdot \nabla u = \cos k_x x \cos k_y y \quad \text{in } \Omega, \tag{51}$$

where Ω denotes the computational domain and $x, y \in [-1, 1]$. The boundary condition was given as the following Neumann boundary condition.

$$\nabla u \cdot \boldsymbol{n} = 0 \quad \text{on } \partial\Omega, \tag{52}$$

where $\partial\Omega$ is the boundary of Ω. When $k_x = m_x \pi$ and $k_y = m_y \pi$ (m_x and m_y are integers), the exact solutions $\tilde{u}$ of these problems are as follows:

$$\tilde{u} = -\frac{1}{k_x^2 + k_y^2} \cos k_x x \cos k_y y. \tag{53}$$

When the calculation domain is divided into $N \times N$ square grids, the exact flux, $\tilde{\psi}$, at each edge is given as

$$\tilde{\psi}_{[\bar{i},j]_{\mathrm{E}}} = \frac{k_y}{k_x} \frac{1}{k_x^2 + k_y^2} \left[\sin k_x x \sin k_y y\right]_{[i,j]}^{[i+1,j]}, \tag{54a}$$

$$\tilde{\psi}_{[i,\bar{j}]_{\mathrm{E}}} = \frac{k_x}{k_y} \frac{1}{k_x^2 + k_y^2} \left[\sin k_x x \sin k_y y\right]_{[i,j]}^{[i,j+1]}. \tag{54b}$$

We took the parameter as $(k_x, k_y) = (2\pi, \pi)$ for fair comparisons, as the error of DFM happens to be zero when $k_x = k_y$. Figure 5 shows the normalized RMSE and calculation time of DFM, FastDFM, FDM, and FVM. Figure 5 (a) shows that the DFM error is approximately one-seventh of the error of FDM and approximately two-ninth of that of the FVM. The errors of DFM and FastDFM are the same. The dependence of errors on N in all methods is proportional to N^{-4}, which is reasonable because the size of the edges, Δ, is proportional to the reciprocal of N, and the accuracy of the methods has $\mathcal{O}(\Delta^4)$. From Figure 5 (b), we can see that the calculation time of FastDFM is similar to that of FDM and FVM, and it is almost one-eighth that of DFM. Further, we can see that the computational time for all methods exhibits N^6-dependency. The reduction rate of FastDFM and the N^6-dependency can be explained by the same reason. Most of the computational time is required for solving the simultaneous equations, where we employed the solver based on the LU decomposition scheme. In the LU decomposition, the computational time is proportional to the cube of the number of variables. As the number of variables of DFM is almost twice that of FastDFM, DFM requires eight times the computational time of FastDFM. Similarly, as the numbers of variables in all methods are proportional to N^2, the computational cost is proportional to N^6.

5.2. Problem 2: Dirichlet boundary condition

Next, we compare the performance of the problems of the Dirichlet boundary condition. If the source term is a trigonometric function as in Problem 1, the accuracy of FVM becomes 5th-order accuracy, because the 4th-order error term is zero. As this problem does not have generality, we set the following exact solution:

$$\tilde{u} = (a - \cosh k_x x)(b - \cosh k_y y), \tag{55}$$

where a, b, k_x, and k_y are constants.

This $\tilde{u}$ becomes zero, when

$$a - \cosh k_x x = 0 \text{ or } b - \cosh k_y y = 0. \tag{56}$$

When the computational domain, Ω, is set to $x, y \in [-1, 1]$, and the boundary conditions are set as

$$u = 0 \quad \text{on } \partial\Omega, \tag{57}$$

the boundary condition can be satisfied by the following relationships.

$$a = \cosh k_x, \tag{58a}$$

$$b = \cosh k_y. \tag{58b}$$

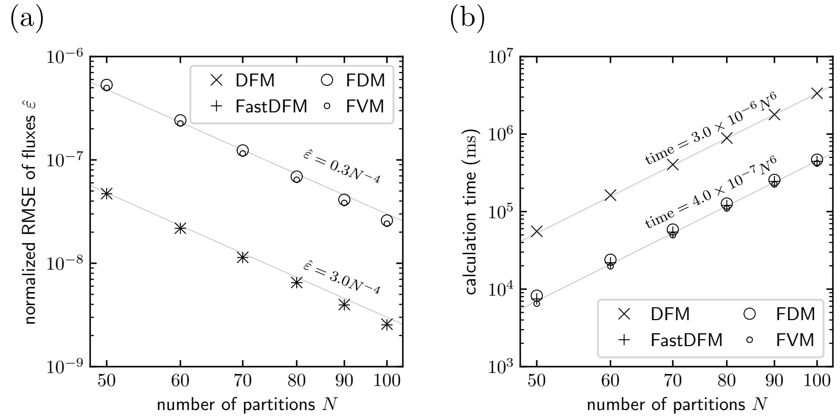

Figure 6: Comparison of the performance of DFM, FastDFM, FDM, and FVM in terms of (a) normalized RMSE and (b) calculation time in Problem 2.

For this solution, Poisson's equation is

$$\nabla \cdot \nabla u = -k_x^2(b - \cosh k_y y)\cosh k_x x - k_y^2(a - \cosh k_x x)\cosh k_y y \quad \text{in } \Omega. \tag{59}$$

Figure 6 shows the normalized RMSE and calculation time of DFM, FastDFM, FDM, and FVM, when $k_x = \cosh^{-1}(2)$ and $k_y = \cosh^{-1}(3)$ in Eq.(59), where $\cosh^{-1}(\cdot)$ is the inverse hyperbolic cosine function. This figure shows that DFM is approximately 10 times more accurate than FDM and FVM, and FastDFM has almost the same speed as the other methods.

From these comparisons, the evaluation of flux by FastDFM is more accurate than that by FDM and FVM without increase in calculation time.

6. Conclusion

We proposed the DFM to compute accurate flux values, directly. A system equation of DFM is formulated based on two mathematical formulas: Gauss's divergence theorem and Stokes' theorem. No errors are included in the equations based on Gauss's divergence theorem. However, the equations based on Stokes' theorem include truncation errors of the Taylor series expansion. In this paper, the 2nd- and 4th-order formulations of DFM are presented. The numerical result of DFM with 4th-order accuracy shows a reasonable error trend. Furthermore, the error of the flux calculated by DFM is smaller than that obtained by FDM and FVM. In terms of computational time, DFM is slower than FDM and FVM because the number of variables of DFM is almost twice that of FDM and FVM in 2-D problems. To reduce the computational time, we also developed a method called FastDFM in which the flux on the edge is represented by a sum of the fictitious flux on the edge and the fictitious potentials on the nodes located at the ends of the edge. Because the fictitious flux is determined with a small computational cost, most of the computational time is exhausted to solve the fictitious potential on the nodes. In this scheme, the number of variables is the same as that of the FDM and FVM. From the numerical results, we can show that the calculation time of FastDFM is similar to that of FDM and FVM. Furthermore, the accuracy of FastDFM is the same as that of DFM. Therefore, for the purpose of evaluating flux, FastDFM is superior to FDM and FVM.

Acknowledgments

This research was partially supported by JSPS KAKENHI grant number 18K04158.

References

[1] A. Bejan, A.D. Kraus, Heat transfer handbook, volume 1, John Wiley & Sons, New Jersey, 2003.

[2] C.A. Brebbia, S. Walker, Boundary element techniques in engineering, Elsevier, Amsterdam, 2016.

[3] M. Clemens, T. Weiland, Discrete electromagnetism with the finite integration technique, Progress In Electromagnetics Research 32 (2001) 65–87.

[4] L. Collatz, The numerical treatment of differential equations, volume 60, Springer Science & Business Media, Berlin, 2012.

[5] R. Eymard, T. Gallouët, R. Herbin, Finite volume methods, Handb. Numer. Anal. 7 (2000) 713–1018.

[6] G. Fairweather, A. Karageorghis, The method of fundamental solutions for elliptic boundary value problems, Adv. Comput. Math. 9 (1998) 69–95.

[7] B.J. Gaël Guennebaud, et al., Eigen v3, http://eigen.tuxfamily.org, 2010.

[8] F.D. Gakhov, Boundary Value Problems, Elsevier, Amsterdam, 2014.

[9] M.A. Golberg, The method of fundamental solutions for Poisson's equation, Eng. Anal. Bound. Elem. 16 (1995) 205–213.

[10] J.M. Jin, The finite element method in electromagnetics, John Wiley & Sons, New Jersey, 2015.

[11] J. Liu, Z. Li, L. Wang, L. Jiao, Numerical simulation of the transient flow in a radial flow pump during stopping period, J. Fluids Engrg. 133 (2011).

[12] M. Rundo, Models for flow rate simulation in gear pumps: a review, Energies 10 (2017) 1261.

[13] S. Sauter, C. Schwab, Boundary Element Methods, Springer Series in Computational Mathematics, Springer Berlin Heidelberg, Heidelberg, 2010.

[14] J. Shen, T. Tang, L.L. Wang, Spectral methods: algorithms, analysis and applications, volume 41, Springer Science and Business Media, Heidelberg, 2011.

[15] W. Spotz, High-order compact finite difference schemes for computational mechanics, Ph.D. thesis, Faculty of the Graduate School of the University of Texas at Austin in Partial Fulfillment of the requirements for the degree of Doctor of philosophy, 1995.

[16] D.M. Sullivan, Electromagnetic simulation using the FDTD method, John Wiley & Sons, New Jersey, 2013.

[17] C. Wadhwa, Electrical power systems, New Ageing International, New Delhi, Delhi, 2006.

[18] B.M. Weedy, B.J. Cory, N. Jenkins, J.B. Ekanayake, G. Strbac, Electric power systems, John Wiley & Sons, New Jersey, 2012.

[19] T. Weiland, Eine methode zur lösung der Maxwellschen gleichungen für sechskomponentige felder auf diskreter basis (German), Electronics and Communication, AEÜ 31 (1977) 116–120.

[20] Q. Yang, Z. Zhang, M. Liu, J. Hu, Numerical simulation of fluid flow inside the valve, Procedia Engrn. 23 (2011) 543–550.

[21] M.U. Zapata, R.I. Balam, High-order implicit finite difference schemes for the two-dimensional Poisson equation, Appl. Math. Comput. 309 (2017) 222–244.