

ニューラル陰関数場を用いた境界面捕捉型流体解析

Flow Analysis with an Interface-Capturing Technique Using Neural Implicit Functions

根本琢巳¹⁾ 馬込望²⁾ 塚本顕成³⁾ Hsu, Ming-Chen⁴⁾ 三目直登⁵⁾

Takumi Nemoto, Nozomi Magome, Akinari Tsukamoto, Ming-Chen Hsu and Naoto Mitsume

¹⁾筑波大学 システム情報工学研究群 (〒 305-8573 茨城県つくば市天王台 1-1-1, E-mail: nemoto.takumi.tkb_ef@u.tsukuba.ac.jp)

²⁾筑波大学 システム情報工学研究群 (〒 305-8573 茨城県つくば市天王台 1-1-1, E-mail:

magome.nozomi.sw@alumni.tsukuba.ac.jp)

³⁾筑波大学 システム情報工学研究群 (〒 305-8573 茨城県つくば市天王台 1-1-1, E-mail:

tsukamoto.akinari.qw@alumni.tsukuba.ac.jp)

⁴⁾Ph.D. Iowa State University, Professor(Ames, IA, 50011, USA, E-mail: jmchsu@iastate.edu)

⁵⁾博(工) 筑波大学 システム情報系 助教 (〒 305-8573 茨城県つくば市天王台 1-1-1, E-mail:mitsume@kz.tsukuba.ac.jp)

In this study, we develop a interface-capturing finite element method that incorporates neural implicit functions as an approach to moving boundary problems assisted by photogrammetry techniques. In the proposed method, a neural implicit functions representing the geometry of an object surface are distributed over the computational domain, and boundary conditions are imposed using a penalty method. To validate the proposed approach, we conducted numerical simulations of incompressible viscous fluid flows involving fixed boundaries.

Key Words : Neural implicit functions, SUPG/PSPG method, Penalty method

1. 序論

バイオミメティクスは、生物の構造や機能、生産プロセスなどから着想を得て、新しい技術の開発やものづくりに活かそうとする科学技術である。これに関連し、数値解析を用いて鳥類や昆虫の羽ばたき飛翔の原理解明を試み、これに基づく羽ばたき飛翔する微小飛行体の開発について研究されている [1,2]。こういった羽ばたき飛翔する微小飛行体の設計に際し、羽ばたき飛翔を模倣する仮想空間の構築が有用である。羽ばたき飛翔を仮想空間上で模倣するには、飛翔環境および生物の視覚機能の再現と、翼と空気流動との移動境界問題にもとづく自動制御系の構築が必要となる。

生物の構造を含む飛翔環境や視覚機能を仮想空間上で再現するにあたり、周囲の環境や障害物等の形状データが必要になる。それに加え、実際の機体が飛行中に周囲の環境を認識して自身の機構を自動制御するために、これらの形状データ取得においてリアルタイム性が求められる。そこで、写真や動画に映る物体の形状や風景を仮想空間上で再現するフォトグラメトリ技術の応用が考えられる。近年では、機械学習を用いて取得したニューラル陰関数場で物体表面の形状を表現する手法の研究が進められている。例えば、NeuS[3]は写真や動画などの直接的な視覚情報を入力としたニューラルネットワークから、物体表面の位置を陰的に表す量として空間内に分布する符号付き距離関数 (signed distance function : SDF) を出力し、3D モデルを自動生成する。

NeuS から得られる SDF は移動境界問題の解析に応用できる。移動境界問題に対する代表的な数値解法の一つに境界面捕捉法 [4] がある。境界面捕捉法は固定された解析メッシュを用いる手法であり、前述の SDF な

どを用いて、間接的に境界位置を表現する。移動境界を間接的に表現するため高精度化には高解像度メッシュを要するが、複雑な移動境界に対してメッシュの再構築が不要であり、メッシュ作成の技術的制約が排除される。SDF は空間内の各点から物体表面までの距離を表すため、解析領域内に SDF 値を分布させて境界の位置を定義できる。NeuS から、解析メッシュに対応する SDF 場を取得することで、境界面捕捉法による解析モデルの作成をスムーズに行うことが可能となる。

一般に、境界面捕捉法では移動境界と要素境界が常に一致するとは限らないため、境界上の節点値を直接固定する従来の Dirichlet 境界条件処理が困難である。このように Dirichlet 境界条件を直接扱えない場合の処理として、埋め込み境界 (immersed boundary : IB) 法 [5] が用いられる。IB 法は Euler 座標系での離散化による流体解析スキームを残して Lagrange 記述される構造物と流体の連成解析を行う手法である。IB 法において境界条件を課す方法の一つにペナルティ法 [6,10] がある。ペナルティ法では境界上の関数値が指定された値から逸脱する場合に罰則として、関数値を拘束する弾性力を附加して近似的に Dirichlet 境界条件を満足させる。

本研究では、境界面が移動および変形しない固定境界を含む非圧縮性粘性流体の解析を行う。SDF を用いた境界面捕捉法にペナルティ法を導入した境界面捕捉型流体ソルバーを開発し、境界面と要素境界の位置関係によらず Dirichlet 境界条件を近似的に考慮できることを示した。非圧縮性粘性流体の解析には Streamline upwind/Petrov–Galerkin (SUPG) 項および Pressure stabilized/Petrov–Galerkin (PSPG) 項を導入した SUPG/PSPG 法 [7] にもとづく有限要素法を用いた。

2. SUPG/PSPG 法による支配方程式の離散化

非圧縮性粘性流体の支配方程式となる運動方程式と連続式をそれぞれ式(1),(2)に、Dirichlet境界条件およびNeumann境界条件をそれぞれ式(3),(4)に示す。

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}, p) = \mathbf{0} \text{ in } \Omega \quad (1)$$

$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \quad (2)$$

$$\mathbf{u} = \hat{\mathbf{u}} \text{ on } \Gamma_g \quad (3)$$

$$\boldsymbol{\sigma}(\mathbf{u}, p) \mathbf{n} = \hat{\mathbf{h}} \text{ on } \Gamma_h \quad (4)$$

ρ, \mathbf{u}, p はそれぞれ密度、流速ベクトル、圧力である。 Ω は境界 Γ で囲まれた解析領域とする。 Γ_g と Γ_h はそれぞれ Dirichlet 境界条件と Neumann 境界条件が与えられる境界であり、 $\Gamma = \Gamma_g + \Gamma_h$ が成り立つものとする。 $\hat{\cdot}$ 付きの量は既知の量、 \mathbf{n} は境界外向き単位法線ベクトルを表す。また、 $\boldsymbol{\sigma}$ は式(5)に定義される Cauchy の応力テンソルである。

$$\boldsymbol{\sigma}(\mathbf{u}, p) = -p\mathbf{I} + 2\mu\boldsymbol{\varepsilon}(\mathbf{u}) \quad (5)$$

ここで、 \mathbf{I} は 2 階の単位テンソル、 μ は粘性係数である。 $\boldsymbol{\varepsilon}$ は式(6)に定義される変形速度テンソルである。

$$\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2} \left\{ \nabla \mathbf{u} + (\nabla \mathbf{u})^T \right\} \quad (6)$$

支配方程式に対して重み付き残差法を適用し、弱形式化を行うことで弱形式(7)が得られる。

$$\begin{aligned} & \int_{\Omega} \mathbf{w} \cdot \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) d\Omega \\ & + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}(\mathbf{u}, p) d\Omega \\ & + \int_{\Omega} q \nabla \cdot \mathbf{u} d\Omega - \int_{\Gamma_h} \mathbf{w} \cdot \hat{\mathbf{h}} d\Gamma = 0 \end{aligned} \quad (7)$$

ここで、 \mathbf{w} は運動方程式に対する重み関数ベクトル、 q は連続式に対する重み関数である。

この弱形式(7)を構成するそれぞれの関数に対して、有限要素法にもとづいて形状関数による関数近似を導入し、SUPG/PSPG 法における重み関数を用いることで安定化項を追加する。まず、 n_{node} 個の節点を用いて、解析領域 Ω を n_{elem} 個の要素領域 Ω_e ($e = 1, 2, \dots, n_{\text{elem}}$) に分割する。次に、各節点 α に対応する形状関数 N_α ($\alpha = 1, 2, \dots, n_{\text{node}}$) を用意し、節点 α における流速ベクトル \mathbf{u}_α と圧力 p_α 、運動方程式の重み関数値ベクトル \mathbf{w}_α と連続式の重み関数値 q_α を用いて、近似式(8),(9),(10),(11)を定義する。

$$\mathbf{u} \simeq \mathbf{u}^h = \sum_{i=1}^{n_{\text{node}}} \mathbf{u}_\alpha N_\alpha \quad (8)$$

$$p \simeq p^h = \sum_{i=1}^{n_{\text{node}}} p_\alpha N_\alpha \quad (9)$$

$$\mathbf{w} \simeq \mathbf{w}^h = \sum_{i=1}^{n_{\text{node}}} \mathbf{w}_\alpha N_\alpha \quad (10)$$

$$q \simeq q^h = \sum_{i=1}^{n_{\text{node}}} q_\alpha N_\alpha \quad (11)$$

SUPG/PSPG 法では重み関数 \mathbf{w} として、式(10)に安定化項を加えた式(12)を用いる。

$$\mathbf{w} \simeq \mathbf{w}^h + \tau_M \mathbf{u}^h \cdot \nabla \mathbf{w}^h + \tau_M \frac{\nabla q^h}{\rho} \quad (12)$$

τ_M は安定化の程度を操作する安定化パラメータである。式(12)の右辺第 2 項は移流による数値不安定性を回避する役割を持つ SUPG 項である。右辺第 3 項は非圧縮条件による数値不安定性を回避する役割を持つ PSPG 項である。弱形式(7)の重み関数 \mathbf{w} を式(12)で置き換えると、式(13)が得られる。

$$\begin{aligned} & \int_{\Omega} \mathbf{w}^h \cdot \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) d\Omega \\ & + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}^h) : \boldsymbol{\sigma}(\mathbf{u}^h, p^h) d\Omega \\ & + \int_{\Omega} q^h \nabla \cdot \mathbf{u}^h d\Omega - \int_{\Gamma_h} \mathbf{w}^h \cdot \hat{\mathbf{h}} d\Gamma \\ & + \sum_{e=1}^{n_{\text{elem}}} \int_{\Omega_e} \tau_M^e \left(\mathbf{u}^h \cdot \nabla \mathbf{w}^h + \frac{\nabla q^h}{\rho} \right) \cdot \mathbf{r}_M(\mathbf{u}^h, p^h) d\Omega \\ & = 0 \end{aligned} \quad (13)$$

ここで、 \mathbf{r}_M は運動方程式の残差であり、式(14)で表される。

$$\mathbf{r}_M(\mathbf{u}^h, p^h) = \rho \left(\frac{\partial \mathbf{u}^h}{\partial t} + \mathbf{u}^h \cdot \nabla \mathbf{u}^h \right) - \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}^h, p^h) \quad (14)$$

安定化項について、安定化パラメータが要素ごとに異なるため、要素ごとに積分したものの総和で表している。本研究では、要素代表長さ h_e および安定化パラメータ τ_M^e はそれぞれ式(15),(16)で定義する。

$$h_e = \sqrt[3]{\Delta x_e \Delta y_e \Delta z_e} \quad (15)$$

$$\tau_M^e = \left\{ \left(\frac{2}{\Delta t} \right)^2 + \left(\frac{2\|\mathbf{u}^h\|}{h_e} \right)^2 + \left(\frac{4\mu}{\rho h_e^2} \right)^2 \right\}^{-\frac{1}{2}} \quad (16)$$

ここで、 $\Delta x_e, \Delta y_e, \Delta z_e$ はそれぞれ要素 e の x, y, z 軸方向の長さであり、 h_e は要素 e の体積の立方根を意味する。

重み付き残差法における重み関数の任意性から式(13)を離散化し、時間に関する非線形常微分方程式(17),(18)を得られる。

$$(\mathbf{M} + \mathbf{M}_S) \frac{d\mathbf{U}_i}{dt} + (\mathbf{A}(\mathbf{U}_j) + \mathbf{A}_S(\mathbf{U}_j)) \mathbf{U}_i \quad (17)$$

$$-(\mathbf{G}_i - \mathbf{G}_{S,i}) \mathbf{P} + \mathbf{D}_{ij} \mathbf{U}_j = \mathbf{F}_i$$

$$\mathbf{C}_j \mathbf{U}_j + \mathbf{M}_{Pj} \frac{d\mathbf{U}_j}{dt} + \mathbf{A}_{Pj}(\mathbf{U}_k) \mathbf{U}_j - \mathbf{G}_P \mathbf{P} = \mathbf{0} \quad (18)$$

ここで、 \mathbf{U}_i, \mathbf{P} はそれぞれ i 軸方向流速、圧力の全体の節点値ベクトルである。 (\mathbf{U}_j) は移流項の計算に使われる移流速度を表す。 $\mathbf{M}, \mathbf{A}, \mathbf{G}, \mathbf{D}, \mathbf{C}$ はそれぞれ、時間微分項、移流項、圧力項、粘性項、連続式の項に対応する係数行列であり、添字 S, P はそれぞれ SUPG 項、PSPG 項由来の行列であることを表す。式(17),(18)に、Crank–Nicolson 法による時間方向の離散化および

Adamas–Bashforth 法による移流項の線形化を行い、式(19),(20)の離散化式が得られる。ここでは、式(4)において $\hat{\mathbf{h}} = \mathbf{0}$ とする齊次 Neumann 境界条件が課される場合を考え、右辺項ベクトル \mathbf{F}_i は無視する。

$$\begin{aligned} & \left[\frac{\mathbf{M} + \mathbf{M}_S}{\Delta t} + \frac{\mathbf{A}(\bar{\mathbf{U}}_j) + \mathbf{A}_S(\bar{\mathbf{U}}_j)}{2} \right] \mathbf{U}_i^{n+1} \\ & + \frac{\mathbf{D}_{ij}}{2} \mathbf{U}_j^{n+1} - [\mathbf{G}_i - \mathbf{G}_{Si}] \mathbf{P}^{n+1} \end{aligned} \quad (19)$$

$$\begin{aligned} & = \left[\frac{\mathbf{M} + \mathbf{M}_S}{\Delta t} - \frac{\mathbf{A}(\bar{\mathbf{U}}_j) + \mathbf{A}_S(\bar{\mathbf{U}}_j)}{2} \right] \mathbf{U}_i^n + \frac{\mathbf{D}_{ij}}{2} \mathbf{U}_j^n \\ & \left[\mathbf{C}_j + \frac{\mathbf{M}_{Pj}}{\Delta t} + \frac{\mathbf{A}_{Pj}(\bar{\mathbf{U}}_k)}{2} \right] \mathbf{U}_j^{n+1} + \mathbf{G}_P \mathbf{P}^{n+1} \end{aligned} \quad (20)$$

この離散化式を連立一次方程式として、線形ソルバーを用いて解き、解となる節点値ベクトル $\mathbf{U}_i^{n+1}, \mathbf{P}^{n+1}$ を求める。

3. SDF を用いた境界面捕捉型有限要素法

(1) NeuS の概要

本節では、はじめに NeuS のもとになった技術である NeRF(Neural Radiance Fields)[8] を紹介し、相違点に注目して NeuS の概要を述べる。

NeRF は様々な位置や方向から撮影した複数枚の 2 次元画像の入力から、画像に映るシーンを 3D モデルとして仮想空間上に再構成するニューラルネットワーク技術である。NeRF は以下の手順で、ネットワークを最適化させながら、画像入力からシーンを再構成する。

1. 画像を撮影したカメラの位置と視線方向から、カメラの視線直線を示すレイを定め、レイ上に偏りなく分布するサンプル点のセットを作成する。
2. 1 で取得したサンプル点ごとに、注目するサンプル点の位置ベクトル $\mathbf{x} = (x, y, z)$ と、 \mathbf{x} を含むレイの視線方向 $\mathbf{d} = (\theta, \phi)$ の 5 次元入力に対して式(21)の positional encoding $\gamma: \mathbb{R} \rightarrow \mathbb{R}^{2L}$ を施し、高次元空間にマッピングする。

$$\begin{aligned} & \gamma(p) \\ & = (\sin(2^0 \pi p), \cos(2^0 \pi p), \sin(2^1 \pi p), \cos(2^1 \pi p), \dots, \sin(2^{L-1} \pi p), \cos(2^{L-1} \pi p)) \end{aligned} \quad (21)$$

ここで、 p は入力する変数である。 L は分解する高周波成分の数でありユーザが設定する自然数である。

3. positional encoding を通した入力ベクトル $\gamma(\mathbf{x}, \mathbf{d}) = (\gamma(x), \gamma(y), \gamma(z), \gamma(\theta), \gamma(\phi))$ を multi-layer perceptron (MLP) ネットワーク $F_\Theta: (\gamma(\mathbf{x}, \mathbf{d})) \rightarrow (\mathbf{c}, \sigma)$ に与え、空間内のサンプル点において、物体の存在を示す体積密度場 σ と視線に依存する色 $\mathbf{c} = (r, g, b)$ を出力する。
4. 体積密度場 σ の情報をもとに、物体表面が存在する確率の高い位置付近のサンプル点を増やす階層型サンプリング[9]を行い、複数回 2, 3 の処理を繰り返す。

5. レイ上の色と密度を集めて、レイが通過する画像ピクセルの色を計算するボリュームレンダリングにより、元の入力画像を撮影したカメラの撮影位置と方向から、仮想空間上のシーンを疑似的に再撮影した画像を生成する。

6. 真の色である入力画像のピクセルの色と、5 で得られた画像のピクセルの色を比較することで損失を求める。損失が最小になるように勾配降下法によるネットワーク F_Θ の学習を行い、2 に戻る。学習が終了したら、5 で生成されたシーンを出力する。

本研究で活用する NeuS は、NeRF を発展させたニューラルネットワーク技術である。NeuS は NeRF と同様に、様々な位置や方向から撮影した複数枚の 2 次元画像を入力とし、被写体の表面形状を 3D モデルとして仮想空間上に再構成する。ネットワークを最適化させる手順は NeRF と同様であるが、被写体の表面形状の表現に体積密度場ではなく符号付き距離関数 (signed distance function: SDF) 場を出力する点が異なる。3 次元空間 Ω 内のある点 \mathbf{x} における SDF 値 $f_{\text{SDF}}(\mathbf{x})$ は \mathbf{x} から物体表面 $\partial\Omega$ までの垂直距離 $d(\mathbf{x}, \partial\Omega)$ に正負の符号をつけた値であり、式(22)で表される。

$$f_{\text{SDF}}(\mathbf{x}) = \begin{cases} -d(\mathbf{x}, \partial\Omega) & \mathbf{x} \in \Omega_{\text{in}} \\ d(\mathbf{x}, \partial\Omega) & \mathbf{x} \notin \Omega_{\text{in}} \end{cases} \quad (22)$$

例えば、 Ω が物体表面 $\partial\Omega$ を境界として物体内部領域 Ω_{in} と物体外部領域 Ω_{out} に分けられるとする。ある点 $\mathbf{x} \in \Omega$ が $\mathbf{x} \in \Omega_{\text{in}}$ を満たすなら負の値を持ち、境界 $\partial\Omega$ に近づくにつれ値が増加して、 $\mathbf{x} \in \partial\Omega$ を満たす位置で 0 になる。 $\mathbf{x} \in \Omega_{\text{out}}$ を満たすなら正の値を取り、境界 $\partial\Omega$ から離れるにつれ値は増加し続ける。ここで、正の値を内部、負の値を外部とする規約が採用されることもある。また f_{SDF} はほとんど至るところで微分可能であり、式(23)のアイコナール方程式を満たす。

$$|\nabla f_{\text{SDF}}| = 1 \quad (23)$$

これは、SDF 場の法線方向を表す勾配ベクトルの大きさが常に 1 であることを表す。SDF は定義上、点 \mathbf{x} から境界 $\partial\Omega$ までの最短距離を示すため、注目する点を移動した場合における $\partial\Omega$ までの距離の増加率が常に一定となることから自然に満たされる性質である。NeuS のネットワークの最適化では、得られた f_{SDF} が式(23)を満足するか評価する損失項を追加する。

(2) ペナルティ法による境界条件の付与

SDF 等を用いて境界を陰的に表現する境界面捕捉法では、境界と要素境界が常に一致するとは限らないため、境界上の節点における解の値を直接固定する従来の Dirichlet 境界条件処理が困難である。このように、条件を課そうとする境界と要素境界が一致していない場合に用いられる手法としてペナルティ法[10,6]がある。ペナルティ法は、等式制約条件の式に対応する項を弱形式に追加する形式で定式化される。 $\hat{\mathbf{u}}$ を物体境界 $\partial\Omega$ 上で与えられる既知関数ベクトルとして、 $\partial\Omega$ 全体に $\mathbf{u} = \hat{\mathbf{u}}$ の Dirichlet 境界条件が課されるとき、等式制約条件として式(24)を定義する。

$$g(\mathbf{u}) = \mathbf{u} - \hat{\mathbf{u}} = \mathbf{0} \text{ on } \partial\Omega \quad (24)$$

このとき、式(7)の弱形式にペナルティ法にもとづく追加項を導入した式(25)を得る。

$$\begin{aligned} & \int_{\Omega} \mathbf{w} \cdot \rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) d\Omega + \int_{\Omega} \boldsymbol{\varepsilon}(\mathbf{w}) : \boldsymbol{\sigma}(\mathbf{u}, p) d\Omega \\ & + \int_{\Omega} q \nabla \cdot \mathbf{u} d\Omega - \int_{\Gamma_h} \mathbf{w} \cdot \hat{\mathbf{h}} d\Gamma \\ & + \bar{p} \int_{\partial\Omega} \mathbf{w} \cdot (\mathbf{u} - \hat{\mathbf{u}}) d\Gamma = 0 \end{aligned} \quad (25)$$

式(25)の左辺第5項はペナルティ項と呼ばれ、解である未知関数ベクトル \mathbf{u} を既知関数ベクトル $\hat{\mathbf{u}}$ に拘束するような弾性を意味する。ペナルティ係数 \bar{p} は弾性係数に相当するので、大きな値であるほど、より強い拘束を意味するため等式制約条件をより厳密に満たす。

ここで、境界 $\partial\Omega$ は要素境界と常に一致するとは限らないため、面積分が困難である。そこで、表面に働く力を体積力に変換する continuum surface force (CSF) モデル [11] を用いて体積積分を行う。CSF モデルの概略図を図 1 に示す。

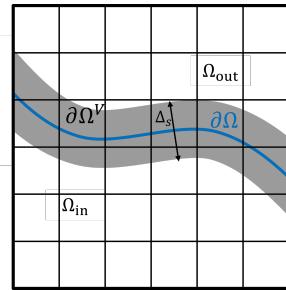


図-1 CSF モデルの概略図

図 1において、 Ω_{in} は物体内部領域、 Ω_{out} は物体外部領域、 $\partial\Omega$ は物体表面、 $\partial\Omega^V$ は境界面 $\partial\Omega$ を跨ぐ厚さ Δ_s の界面領域とする。境界面上の座標 $s \in \partial\Omega$ における表面力を $f^s(s)$ としたとき、この表面力 f^s に対応する体積力 \mathbf{F}^s について、式(26)が成り立つ。

$$\lim_{\Delta_s \rightarrow 0} \int_{\partial\Omega^V} \mathbf{F}^s(\mathbf{x}) d\Omega = \int_{\partial\Omega} f^s(s) d\Gamma \quad (26)$$

f^s の面積分は式(27)により体積積分で表される。

$$\int_{\partial\Omega} f^s(s) d\Gamma = \int_{\partial\Omega^V} f^s(\mathbf{x}) \delta(|\mathbf{x} - s|) d\Omega \quad (27)$$

ここで、 δ は式(28)の Dirac のデルタ関数である。

$$\delta(x) = \begin{cases} \infty & (x = 0) \\ 0 & (x \neq 0) \end{cases} \quad (28)$$

$|\mathbf{x} - s|$ は境界面上の座標 s から空間座標 \mathbf{x} までの距離を表すので式(29)が成り立つ。

$$|\mathbf{x} - s| = |f_{SDF}(\mathbf{x})| \quad (29)$$

Dirac のデルタ関数は任意の $x \in \mathbb{R}$ について $\delta(x) = \delta(-x)$ が成り立つことを考慮すれば、式(30)が得られる。

$$\delta(|\mathbf{x} - s|) = \delta(f_{SDF}(\mathbf{x})) \quad (30)$$

式(26),(27),(30)より、式(31)が成り立つ。

$$\lim_{\Delta_s \rightarrow 0} \mathbf{F}^s(\mathbf{x}) = f(\mathbf{x}) \delta(f_{SDF}(\mathbf{x})) \quad (31)$$

数値計算するにあたり、 $\Delta_s = 0$ すなわち厚さがゼロの界面領域は扱えないので、 Δ_s は有限量で近似する必要がある。そこで、Dirac のデルタ関数値が非ゼロの値を取る幅を有限幅にする近似を取る。ただし、近似デルタ関数 δ_s は式(32)を満たすように定める。

$$\int_{-\infty}^{\infty} \delta_s(x) dx = 1 \quad (32)$$

式(32)を満たす関数として、式(33)に表される近似デルタ関数を用いる。また、式(33)が表す関数を図 2 に示す。

$$\delta_s(f_{SDF}(\mathbf{x})) = \begin{cases} \frac{1 + \cos\left(\frac{2\pi f_{SDF}(\mathbf{x})}{\Delta_s}\right)}{2\Delta_s} & (|f_{SDF}(\mathbf{x})| \leq \Delta_s/2) \\ 0 & (|f_{SDF}(\mathbf{x})| > \Delta_s/2) \end{cases} \quad (33)$$

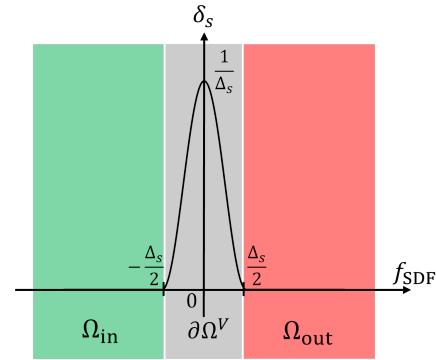


図-2 近似デルタ関数

式(33)を用いて、式(25)のペナルティ項を体積力表示した式(34)が得られる。

$$\begin{aligned} & \bar{p} \int_{\partial\Omega} \mathbf{w} \cdot (\mathbf{u} - \hat{\mathbf{u}}) d\Gamma \\ & \approx \bar{p} \int_{\Omega} (\mathbf{w} \cdot \mathbf{u}) \delta_s(f_{SDF}(\mathbf{x})) d\Omega \\ & - \bar{p} \int_{\Omega} (\mathbf{w} \cdot \hat{\mathbf{u}}) \delta_s(f_{SDF}(\mathbf{x})) d\Omega \end{aligned} \quad (34)$$

式(34)を(25)に代入し、式(8),(10)の近似関数をペナルティ項にも適用して離散化することで、運動方程式の離散化式(35)が得られる。

$$\begin{aligned} & \left[\frac{\mathbf{M} + \mathbf{M}_S}{\Delta t} + \frac{\mathbf{A}(\bar{\mathbf{U}}_j) + \mathbf{A}_S(\bar{\mathbf{U}}_j)}{2} + \mathbf{K}_C \right] \mathbf{U}_i^{n+1} \\ & + \frac{\mathbf{D}_{ij}}{2} \mathbf{U}_j^{n+1} - [\mathbf{G}_i - \mathbf{G}_{S,i}] \mathbf{P}^{n+1} \\ & = \left[\frac{\mathbf{M} + \mathbf{M}_S}{\Delta t} - \frac{\mathbf{A}(\bar{\mathbf{U}}_j) + \mathbf{A}_S(\bar{\mathbf{U}}_j)}{2} \right] \mathbf{U}_i^n + \frac{\mathbf{D}_{ij}}{2} \mathbf{U}_j^n + \mathbf{F}_C \end{aligned} \quad (35)$$

$\mathbf{K}_C, \mathbf{F}_C$ はそれぞれペナルティ項を意味する。

4. 数値例

(1) 解析条件

本検証では、3D キャビティ流れの解析を行った。解析領域を図 3 に示す直方体とする。

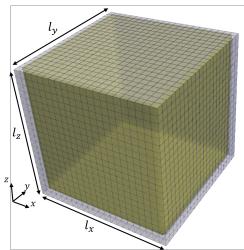


図-3 解析領域

空間の離散化には各座標軸方向に均等に節点を配置して構築される構造格子メッシュを用いた。Lagrange 補間関数を用いた六面体一次要素により離散化している。本検証では、図 3 の直方体領域 Ω 内に 1 辺の長さが 1 の立方体領域 Ω_{in} を置き、 Ω_{in} 内のキャビティ流れの様子を調べる。

ここでは、立方体領域の境界と要素境界の位置関係が異なる場合の解析結果を比較する。解析領域を平面 $y = l_y/2$ で切った断面におけるメッシュの模式図を図 4 に示す。

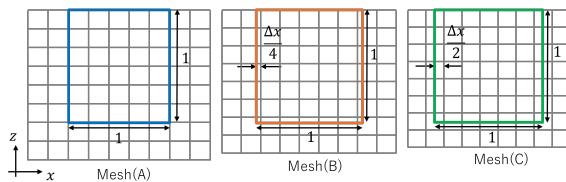


図-4 解析領域断面上のメッシュ

図 4 において、 Δx は六面体要素の x 軸方向の辺の長さを表す。立方体領域の境界と要素境界の位置関係について、すべての Mesh において、一様流速を課す $z = l_z$ の面は立方体領域の境界と要素境界が一致している。Mesh(A) では、立方体領域の他の境界も要素境界と一致している。Mesh(B) では、立方体領域の境界に対し、 $\Delta x/4$ だけ内側の位置に六面体要素の境界が存在する。Mesh(C) では、立方体領域の境界に対し、 $\Delta x/2$ だけ内側の位置に六面体要素の境界が存在する。対称性から、図 4 の断面図は平面 $y = l_y/2$ で切った断面と一致するものとする。ここでは、メッシュを伸縮させることで境界と要素境界の位置関係を操作している。各メッシュに対応する解析モデルの寸法を表 1 に示す。

表-1 解析モデルの寸法

モデル	l_x	l_y	l_z
Mesh(A)	1.0800	1.0800	1.0400
Mesh(B)	1.0693	1.0693	1.0348
Mesh(C)	1.0588	1.0588	1.0297

境界条件について、上面 ($z = l_z$) には一様流速 $\mathbf{u} =$

$(1, 0, 0)^T$ の Dirichlet 境界条件を従来の方法で適用する。他の面には no-slip 条件 $\mathbf{u} = (0, 0, 0)^T$ をペナルティ法を用いて課す。図 5 のように解析領域全体の SDF 分布から境界面の位置を定義し、ペナルティ項の計算に用いる。

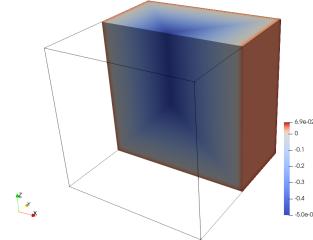


図-5 解析領域内の SDF 分布

ここで、SDF 値の定義には式 (22) を用いる。ペナルティ項の計算に用いる近似デルタ関数の幅 Δ_s は、1 要素あたりの x 軸方向長さを Δx として、 $\Delta_s = 2\Delta x$ と定義する。また、ペナルティ係数 \bar{p} は、 $\bar{p} = \Delta x \times 10^5$ とする [10]。

流体の解析条件として、密度 $\rho = 100.0 \text{ [kg/m}^3]$ 、粘性係数 $\mu = 1.0 \text{ [Pa} \cdot \text{s]}$ の Reynolds 数 100 とする。時間刻み幅は $\Delta t = 5.0 \times 10^{-3} \text{ [s]}$ 、解析終了時間を 10.0 s とする。また、下表に示す分割数を使用する。

表-2 分割数

$l_x/\Delta x$	$l_y/\Delta y$	$l_z/\Delta z$	節点数	要素数
108	108	104	1,247,505	1,213,056

要素内の数値積分は各座標軸方向 3 点の積分点を配置して行う。解析結果について、Jiang et al. [12] の結果と比較する。

(2) 結果および考察

Mesh(C) について解析した結果を例として、Reynolds 数 100 における流速ベクトル図を図 6 に、各 Mesh のパターンについて、参照解と比較した結果を図 7 に示す。

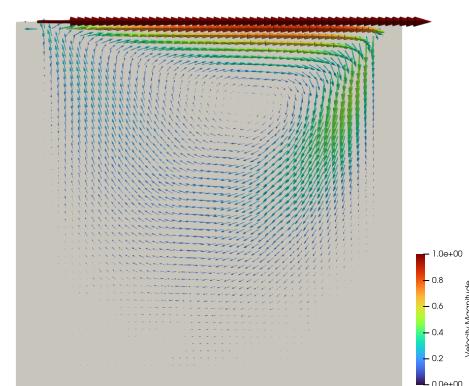


図-6 提案手法による Reynolds 数 100 の 3D キャビティ流れ
流速ベクトル図

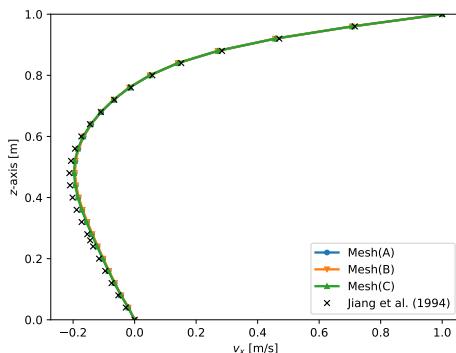


図-7 提案手法による Reynolds 数 100 の 3D キャビティ流れ解析結果

図 6 において、数値振動のような不安定性は見られず、提案手法を用いて安定的に解析できており、解析領域内部にある矩形領域内の循環流が再現されていることが確認できる。

図 7 より、メッシュと境界の位置関係によらず、同程度に参照解に近い結果であるといえる。一方で、 x 軸負の方向における流速の大きさが参照解よりも小さくなっている。これは、ペナルティ法による境界条件の処理により、矩形領域の壁面で流出が生じ、循環流が弱まっていることが原因であると考えられる。

5. 結論

本研究では、SDF で境界面を捕捉し、ペナルティ法を用いて Dirichlet 境界条件を処理する境界面捕捉法を導入した非圧縮性粘性流体解析手法を開発し検証した。提案手法において、解析領域内に SDF 場を分布させることで、解析メッシュの形状とは独立に境界面の位置をモデル化できた。また、ペナルティ法を有限要素法に導入し、境界面と要素境界が一致しない場合においても Dirichlet 境界条件を近似的に課すことができた。以上をもって、NeuS 等のニューラル陰関数ベースのフォトグラメトリ手法と有限要素法を基盤とした流体解析をシームレスに連結する数値解析的基盤を構築した。

謝辞: 本研究は、JST 創発的研究支援事業 JPMJFR215S および JSPS 科研費 23H00475, 24K22288 の支援を受けたものである。ここに記して謝意を表する。

参考文献

- [1] H. Liu, S. Ravi, D. Kolomenskiy, and H. Tanaka. Biomechanics and biomimetics in insect-inspired flight systems. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 371(1704):20150390, 2016.
- [2] K. Kubota and H. Tanaka. Machine learning-based wind classification by wing deformation in biomimetic flapping robots: Biomimetic flexible structures improve wind sensing. *Advanced Intelligent Systems*, page 2400473, 2024.
- [3] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang. NeuS: learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*, NIPS '21, Red Hook, NY, USA, 2024. Curran Associates Inc.
- [4] S. Marella, S. Krishnan, H. Liu, and H.S. Udaykumar. Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations. *Journal of Computational Physics*, 210(1):1–31, 2005.
- [5] L. Zhang, A. Gerstenberger, X. Wang, and W. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193(21–22):2051–2067, 2004.
- [6] Y. Kim and M.-C. Lai. Simulating the dynamics of inextensible vesicles by the penalty immersed boundary method. *Journal of Computational Physics*, 229(12):4840–4853, 2010.
- [7] T.E. Tezduyar, S. Mittal, S.E. Ray, and R. Shih. Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. *Computer Methods in Applied Mechanics and Engineering*, 95(2):221–242, 1992.
- [8] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, and R. Ng. NeRF: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65:99–106, 12 2021.
- [9] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, and R. Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. *arXiv preprint arXiv:2003.08934*, 2020.
- [10] M. Utku and G.F. Carey. Boundary penalty techniques. *Computer Methods in Applied Mechanics and Engineering*, 30(1):103–118, 1982.
- [11] J.U. Brackbill, D.B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335–354, 1992.
- [12] B.-N. Jiang, T. Lin, and L.A. Povinelli. Large-scale computation of incompressible viscous flow by least-squares finite element method. *Computer Methods in Applied Mechanics and Engineering*, 114(3-4):213–231, 1994.