計算流体力学における領域分割法のオブジェクト化に関する研究

Object based approach of DDM in CFD

籠島　高(発表者)　　坪井一洋(指導教官)

Object based approach of Domain Decomposition Method (DDM) in Computational Fluid Dynamics (CFD) is presented in order to perform effective flow simulation, in which we pay our attention to common property of each domain. Design of "Domain class" is described and some results of computation are demonstrated.

1.　はじめに

流れの計算機シミュレーションは計算流体力学(Computational Fluid Dynamics：CFD)と呼ばれ，近年あらゆる産業分野においてその重要性が増してきている．特に，実用的な問題では複雑な解析領域を扱うことが多く，効率的なシミュレーション手法の開発が求められている．このような方法として，すでにいくつかの方法が提案されているが，本研究では領域分割法(Domain Decomposition Method：DDM)に注目する．領域分割法で分割された各部分領域には共通の性質があり，その性質を領域クラスとして定義し，そのインスタンス（領域オブジェクト）を利用することで，領域分割法の構成を効率化できる．

以下では，領域クラスを設計し，Java 言語[1, 2]によって実装した結果について示す．なお，実装に際しては，簡単のため矩形の組み合わせで表現できる解析領域のみに限った．

2.　流れの基礎方程式

粘性流体の運動を表す式は，流速の場に関する非圧縮性ナヴィエ・ストークス方程式（以下，NS 方程式），および連続の式である．擬似圧縮法[3, 4]による NS 方程式と連続の式は，外力を無視し，無次元化すると次のようになる[5].

$$\begin{cases} \dfrac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u}\cdot\mathrm{grad})\boldsymbol{u} = -\mathrm{grad}\,P + \dfrac{1}{\mathrm{Re}}\Delta\boldsymbol{u} \\ \dfrac{\partial P}{\partial t} + \mathrm{div}\ \boldsymbol{u} = 0 \end{cases} \tag{1}$$

ここで，$\boldsymbol{u}$ は流速ベクトル，P は圧力，Re はレイノルズ数と呼ばれる無次元パラメータであり，次式で定義される．

$$\mathrm{Re} = \frac{U_0 L_0}{\nu} \tag{2}$$

ここで，U_0，L_0はそれぞれ速さ，長さの代表値，νは動粘性係数である．したがって，非圧縮性流れは外力がない場合，レイノルズ数に応じて変化する．

式(1)の離散化に物理量の定義点が同一でないスタガードメッシュ（Fig. 1）[6]を用い，空間微分項については２次精度の中心差分，時間微分項については４次精度のルンゲ・クッタ・ジル法[7]を用いた．

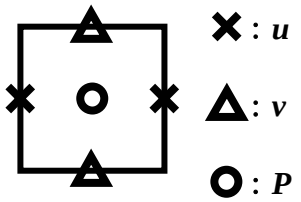

Fig. 1　スタガードメッシュ

3.　領域クラスの設計

オブジェクト指向の中心概念にクラス（class）とインスタンス（instance）がある．インスタンスとは，具体的な「特定のもの」を表し，オブジェクトと呼ぶこともある．すべてのインスタンスはクラスから生成され，それが属しているクラスに共通の性質を持つ[1].

一方，領域分割法の各部分領域は次のような共通の性質を持つことがわかる．

① 流速，圧力を持つ．
② 境界条件が必要である．
③ NS 方程式，連続の式を解く．

したがって，これら共通の性質を持つ「領域クラス」を定義すれば，領域分割法を効率的に構成できると考えられる．

ここでは Domain と WholeDomain という２つのクラスを定義した．Domain は上で述べた領域クラスであり，各部分領域の共通な性質を１つにまとめたもので，流速 $\boldsymbol{u}$ と圧力 P を持ち，境界条件を設定するメソッド，NS 方程式，連続の式を解くメソッドを定義したクラスである．一方，WholeDomain は，Domain クラスのインスタンス（領域オブジェクト）を生成し，領域オブジェクトから計算結果を受け取り，流速ベクトルを表示するメソッドを定義したクラスである．それぞれのオブジェクトの関係を Fig.2 に示す．

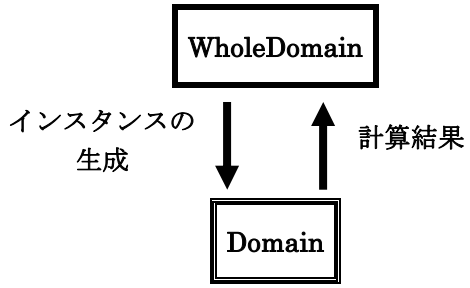

Fig. 2　オブジェクトの関係

また，Domain クラスにはインスタンス生成時の初期化を行うコンストラクタ（constructor）と呼ばれる特殊なメソッドを定義することで，配列の大きさや領域の大きさ，レイノルズ数の異なる様々な領域オブジェクトを作り出すことができる．したがって，解析領域をいくつかの部分領域に分け，その一つ一つに領域オブジェクトを割り当てることで，全領域の解析が可能になる．

領域分割法では境界条件をどう扱うかが重要になる．各部分領域の境界は上下左右４つあるので，それぞれの境界に対して，

$$\mathrm{set}\begin{Bmatrix}\mathrm{North}\\ \mathrm{South}\\ \mathrm{West}\\ \mathrm{East}\end{Bmatrix}\mathrm{BoundaryCondition}\quad(*)$$

という名前の４つのメソッドを考えた．また，境界条件は壁, 流出, 流入, 接合の４つが考えられ，それぞれの場合に応じて流速や圧力は Table 1 に示すような処理が必要になる．ここで，接合境界とは，別の部分領域が隣接し，それぞれの境界が重なっていることを表す．

Table 1　境界条件の種類

境界	流速	圧力
壁	$\boldsymbol{u}=0$	$P_i = P_{i-1}$ $P_0 = P_1$
流出	$\boldsymbol{u}_i = \boldsymbol{u}_{i-1}$ $\boldsymbol{u}_0 = \boldsymbol{u}_1$	
流入	u ＝一定値 v ＝一定値	$P = 0$
接合	$\left.\begin{matrix}\boldsymbol{u}_i = \boldsymbol{u}_{i-1}\\ \boldsymbol{u}_0 = \boldsymbol{u}_1\end{matrix}\right\}$隣接領域	$\left.\begin{matrix}P_i = P_{i-1}\\ P_0 = P_1\end{matrix}\right\}$隣接領域

したがって，境界条件を設定するには，合計 16 個のメソッドを定義する必要がある．そこで，オーバーロードという機能を利用し，16 種類の処理を（＊）の４個のメソッド名で統一した．

計算例として，(a)キャビティー流れ，(b)Block を越える流れ，(c)角柱周りの流れ，(d)管内の流れを考えた．これらを領域分割法により分割し，それぞれの領域オブジェクトを生成した．これらの例における流速ベクトル図を Fig. 3 に示す．用いた領域オブジェクト数は（a）4 個，（b）5 個，（c）22 個，（d）10 個である．

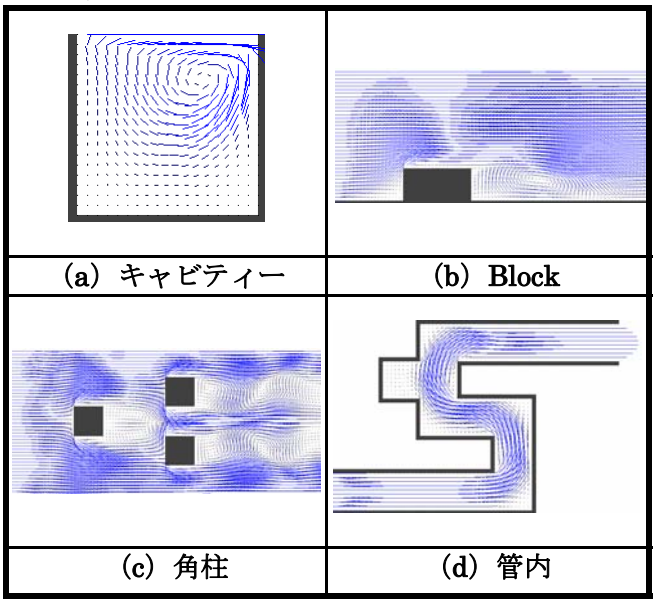

Fig. 3　計算結果

4.　まとめと今後の課題

効率的な流れの数値シミュレーションを行うために，分割された部分領域のもつ共通の性質に着目し，この共通点を領域クラスとして定義することで，領域オブジェクトを生成することができる．必要な領域オブジェクトを生成し，それらを配置することで様々な流れを表現することが可能となった．

今後の課題としては，領域の性質を再検討し，境界条件の扱い等に関して，より抽象化された領域クラスの設計と実装を行いたい．

参考文献

[1] 結城 浩：Java 言語プログラミングレッスン（上）・（下），ソフトバンク パブリッシング，(1999)，第 9 章および第 11 章.

[2] 河西朝雄：Java 入門，技術評論社，(1996)，p.35, pp.359-360.

[3] 日本機械学会 編：流れの数値シミュレーション，コロナ社，(1988)，pp.79-80.

[4] 棚橋隆彦：非圧縮粘性流体の過渡流れ(1)･(2)，機械の研究，第 37 巻，第 3 号および第 4 号，(1985).

[5] 柴田友和：層流における空気の流れの可制御性，平成 8 年度茨城大学工学部システム工学科卒業論文，(1997).

[6] 河村哲也：流体解析 I，朝倉書店，(1996)，p.62.

[7] 梯 鉄次郎，小山英之；数値解析，秀潤社，(1979)，pp.137-138.