Gregory Volumeを用いた流体シミュレーション

上田卓也　藤澤誠　三浦憲二郎

静岡大学

1　緒言

我々の身の回りには，非常に多くの流体が存在している．例えば，蛇口から流れ落ちる水，カップに注がれる紅茶，砂浜に打ち寄せる波など，数え始めるときりがなく，目を覚ましてから再び眠りに落ちるまで，1日中流体を目にしないことはないであろう．

ナビエ・ストークス方程式の移流項を解く場合の手法として，CIP(Constrained Interpolation Profile)法がよく用いられるようになっている．CIP法とは，メッシュ内における物理量のプロファイルを，各点での物理量のみでなく各方向の物理量の勾配値も用いて，スプライン補間のように滑らかに構成する手法であり，サブグリッド精度まで期待できる．離散化に適応的グリッドを用いる場合，大きいグリッドと小さいグリッドが隣接する場所で計算の精度の低下が懸念されるが，サブグリッド精度まで保障できる手法を用いることにより，精度の低下なく計算が可能になる．我々は，補間値の連続性に関して，CIP法よりもより良い結果を得ることができると考えられるGregoryパッチを移流項での計算に用いる．また，CIP法，B型CIP法，Coonsパッチ，およびGregoryパッチの比較，考察を行う．さらに，Gregoryパッチを3次元に拡張したGregory Volumeを提案する．

2　関連研究

これまで，流体シミュレーションを行うための，様々な手法が提案されてきた．FosterとMetaxasは，コンピュータグラフィックに用いるための，3次元ナビエ・ストークス方程式を解くことによる流体シミュレーションの方法を提案した[4]．Stamは，ナビエ・ストークス方程式の移流項の計算にセミラグランジュ法[10]を用いる，安定流体法を提案した[8][9]．安定流体法は，シミュレーションの時間ステップを大きくしても計算を安定に行うことが可能な手法で，現在ではコンピュータグラフィックに利用される流体シミュレーションに広く用いられている．

Losassoらは，Octreeグリッド上でナビエ・ストークス方程式を解き，シミュレーションを行った[5]．計算グリッドにOctreeを用いることで，急激に速度が変化する領域や，液体表面付近のみ，グリッドを細かくすることができ，結果として計算に必要な時間や必要なメモリ量を節約できた．Songらは，Octreeグリッド上で，ナビエ・ストークス方程式を解き，移流項の計算にCIP法[2][3]を用いたシミュレーションを提案した[7]．

3　ナビエ・ストークス方程式

ナビエストークス方程式を次式に示す．

$$\nabla \cdot \mathbf{u} = 0 \tag{1}$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla)\mathbf{u} + \nu \nabla^2 \mathbf{u} - \frac{1}{\rho} \nabla p + \mathbf{f} \tag{2}$$

ここで，$\mathbf{u}$は流体の速度，νは動粘性係数，ρは流体の密度，pは圧力，$\mathbf{f}$は外力である．式(1)は非圧縮性流体の質量保存を表す連続の方程式である．

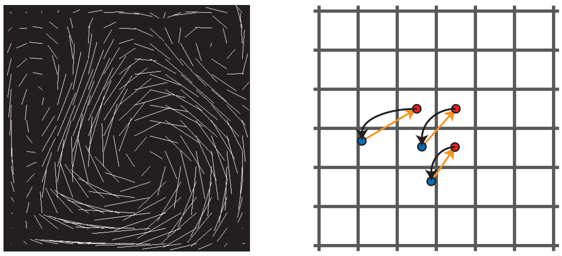
速度場の計算は，式(2)を時間発展させることにより行う．特に，式(2)における右辺第1項の移流項の計算にはセミラグランジュ“ 安定流体 ”法[8][9]を用い，第3項の圧力項の計算にはプロジェクション法を用いた．

3.1　セミラグランジュ“ 安定流体 ”法

車の周りの空気の流れや，管内の水の流れなどのシミュレーションなどは流体の解析が第一の目的であるので，正確さが重要で，低速であってもさほど問題ではない．しかし，コンピュータグラフィックスやゲームの分野においては，もっともらしく見え，かつ高速であるシミュレーションであることが重要

である．よって，タイムステップを大きくしても安定，即ち，計算が発散しないセミラグランジュ“ 安定流体 ”法を用いる．

移流項の計算においてx，y方向それぞれの速度を定義する点をひとつの粒子と考える．ある点での次のタイムステップでの速度を得るために，図1(a)の速度場をもとに図1(b)の赤丸から青丸まで，黒矢印のようにバックトレースする．そして，そのバックトレースした位置での速度を周りの点における値から補完によって求め，それを次のタイムステップにおける図1(b)の赤丸の位置での速度とする．



(a) 速度場　　(b)バックトレース

図 1: セミラグランジュ“ 安定流体 ”法

3.2　プロジェクション法

シミュレーションにおいて非圧縮性流体を考える場合，タイムステップにおいて式(1)の連続の方程式を満足させなければならない．本研究では，圧力を独立に求める方法であるプロジェクション法を用いる．

まずナビエ・ストークス方程式を時間発展させ仮の速度$\mathbf{u}^*$を求める．そして，連続の式を満足させ体積流量を一定に保つために，$\mathbf{u}^*$を式(3)で定義されるポアソン方程式に代入し圧力pを求める．

$$\nabla^2 p = \frac{\nabla \cdot \mathbf{u}^*}{\Delta t} \tag{3}$$

こうして求めた圧力pを式(4)に代入し速度$\mathbf{u}$を求める．

$$\mathbf{u} = \mathbf{u}^* - \Delta t \nabla p \tag{4}$$

4　補間法の連続性

セミラグランジュ法において，バックトレースした点での内挿法について考察する．CIP法やCIP法を拡張したB型CIP法[3]は，1次精度のバイリニア，トリリニア補間と異なり，3次精度で計算を行い，サブグリッド精度まで保障できるため，現在では多くの流体シミュレーションに用いられている．しかし，補間値の連続性に目を向けてみると，CIP法やB型CIP法では計算点での速度値の連続性は保障されているが，2つの計算点を結ぶ辺上では連続性が保障されていない．我々は，形状処理工学における自由曲面内挿法であるCoonsパッチとGregory[1]パッチを応用する．特に，速度ベクトルの勾配の連続性を容易に保障できるGregoryパッチに着目し，3次元の流体シミュレーションにもこの考えを応用し，Gregoryパッチを3次元に拡張したGregory Volumeを提案する．

4.1　CIP法

CIP法とは，速度のみでなく速度勾配も移流させる方法で，関数のプロファイルを維持しながら移流させることが可能となる．

2次元のシミュレーションでは，点(x_i, y_j)周りの4点間のプロファイルは式(5)のような3次式で表される．

$$\begin{aligned} F_{i,j}(x,y) = \ & a_1X^3 + a_2X^2 + a_3X + a_4 + a_5Y^3 \\ & + a_6Y^2 + a_7Y + a_8X^2Y + a_9XY + a_{10}XY^2 \end{aligned} \tag{5}$$

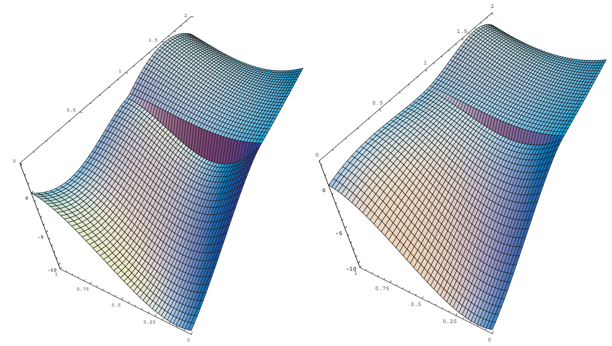
ここで，$X = x - x_i$，$Y = y - y_j$である．点(x_i, y_j)周りの4点での物理量fと，そのx，y各方向の勾配値g^x，g^yより10個の未知数を決定する．

4.2　B型CIP法

CIP法では，式(5)の未知数を10個の拘束条件から求めたが，拘束条件となり得る点(x_i, y_j)周りの4点での物理量fと，x，y各方向の勾配値g^x，g^yは全部で12個あり，全ての拘束条件を満たしてはいない．これを改良するために式(5)に項を2つ加えた式(6)で表されるのがB型CIP法である．

$$\begin{aligned} F_{i,j}(x,y) = \ & a_1X^3 + a_2X^2 + a_3X + a_4 + a_5Y^3 \\ & + a_6Y^2 + a_7Y + a_8X^2Y + a_9XY + a_{10}XY^2 \\ & + a_{11}X^3Y + a_{12}XY^3 \end{aligned} \tag{6}$$

隣接する2つのグリッド上でCIP法とB型CIP法を用いて作成した曲面を図2に示す．



(a) CIP法　　(b) B型CIP法

図 2: 隣接する2つのグリッド上でのプロファイル

4.3　Coonsパッチ

本節では，形状処理工学における自由曲面内挿法を応用するCoonsパッチ[1]を用いたプロファイルの作成について検討する．自由曲面を用いることで，計算セル間で連続なプロファイルが作成可能となる．ここでは，パッチの4つの角における速度値と速度勾配値を用いて補間が可能なCoonsパッチについて考える．

パッチの4角における値と勾配値が与えられれば，双3次のCoonsパッチは以下のように定義できる．

$$\mathbf{P}(u,v) = \mathbf{F}(u)\ \mathbf{M}\ \mathbf{F}(v)^T \tag{7}$$

$$(0 \le u \le 1, 0 \le v \le 1)$$

$$\mathbf{M} = \left[\begin{array}{cc|cc} \mathbf{P}_{00} & \mathbf{P}_{01} & \mathbf{P}^v_{00} & \mathbf{P}^v_{01} \\ \mathbf{P}_{10} & \mathbf{P}_{11} & \mathbf{P}^v_{10} & \mathbf{P}^v_{11} \\ \hline \mathbf{P}^u_{00} & \mathbf{P}^u_{01} & \mathbf{P}^{uv}_{00} & \mathbf{P}^{uv}_{01} \\ \mathbf{P}^u_{10} & \mathbf{P}^u_{11} & \mathbf{P}^{uv}_{10} & \mathbf{P}^{uv}_{11} \end{array}\right]$$

$$\mathbf{F}(t) = [\ f_0(t)\ \ f_1(t)\ \ f_2(t)\ \ f_3(t)\ ]$$

$$\begin{cases} f_0(t) = 2t^3 - 3t^2 + 1 \\ f_1(t) = -2t^3 + 3t^2 \\ f_2(t) = t^3 - 2t^2 + t \\ f_3(t) = t^3 - t^2 \end{cases}$$

ここで，$\mathbf{P}^u_{ij}$と$\mathbf{P}^v_{ij}$は各方向の速度勾配であり，$\mathbf{P}^{uv}_{00}$，$\mathbf{P}^{uv}_{01}$，$\mathbf{P}^{uv}_{10}$，および$\mathbf{P}^{uv}_{11}$はツイストベクトルであり，我々はツイストベクトルの値として0を与える．

4.4　Gregoryパッチ

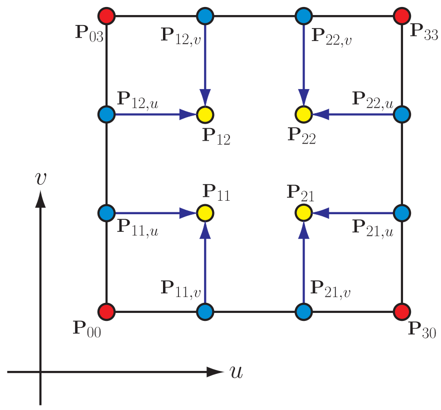


図 3: Gregoryパッチ

本論文ではCoonsパッチでのツイストベクトルを0とするが，ツイストベクトルに0以外の値を与えることによって，本来Coonsパッチが持つ柔軟な内挿が可能となる．その場合，ツイストベクトルにはu，vによる微分の順序に依存しない関数を用いる必要がある．しかし，シミュレーションにおいては各ステップごとに速度値，速度勾配値が変化するため，最適なツイストベクトルは各ステップごとに異なるものになるが，各ステップごとに適切なツイストベクトルを求めるためには，計算領域全域における対極的な決定法を用いる必要がある．これは，計算量が多く，実用的ではない．そこで，Bezier曲面を拡張した，ツイストベクトルを局所的に決定できるGregoryパッチを用いることを考える．本手法はBezier曲面の持つ良い性質を持ち，かつ立式も簡潔である．Gregoryパッチとは，図3のようにパッチの辺上と内部に16個の点を置き，その点の情報を元に内挿する方法であり，Bezier曲面とは異なり$\mathbf{P}_{11}$，$\mathbf{P}_{12}$，$\mathbf{P}_{21}$，および$\mathbf{P}_{22}$の値を混ぜ合わせにより求める．また，本手法を流体シミュレーションのソルバに組み込むために，パッチ式にさらに修正を加える．

Gregoryパッチは二項係数を用いると以下の式(8)で表すことができる．

$$\mathbf{P}(u,v) = \sum_{i=0}^{3}\sum_{j=0}^{3}\binom{3}{i}u^i(1-u)^{3-i}\binom{3}{j}v^j(1-v)^{3-j}\mathbf{P}_{ij}$$

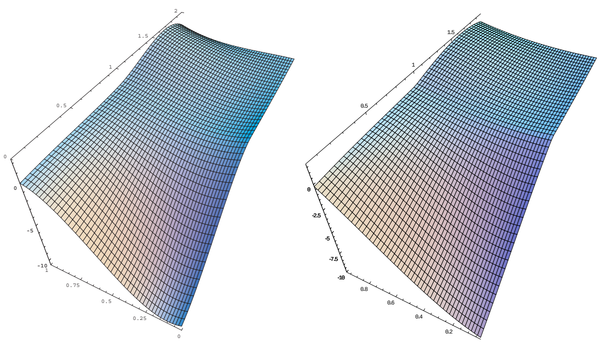
$$(0 \le u, v \le 1) \tag{8}$$

計算セルから与えられる情報は図3での点$\mathbf{P}_{00}, \mathbf{P}_{30}$, $\mathbf{P}_{03}, \mathbf{P}_{33}$における速度と各方向の速度勾配のみである．従って，その他の点における速度を与える必要がある．

パッチの辺上の点については3次曲線で補間する．パッチ内部の点については以下のように考える．まず，パッチ辺上の点における，辺に鉛直方向の速度勾配成分は辺に沿って線形に変化すると考える．すると，パッチ辺上の点における速度勾配が決まるので，図3のように，パッチ内部の各点に最も近傍のパッチ辺上の点での速度値$\mathbf{P}_{ij}$，速度勾配値$\partial \mathbf{P}_{ij}/\partial u$，$\partial \mathbf{P}_{ij}/\partial v$，そして距離$u$，$v$の関係から仮の速度値$\mathbf{P}_{i'j',u}$，$\mathbf{P}_{i'j',v}$を求める．これらを以下の式で混ぜ合わせることにより，パッチ内部の4点での速度が求まる．

$$\mathbf{P}_{11} = \frac{v\mathbf{P}_{11,u} + u\mathbf{P}_{11,v}}{u+v}$$
$$\mathbf{P}_{21} = \frac{v\mathbf{P}_{21,u} + (1-u)\mathbf{P}_{21,v}}{(1-u)+v}$$
$$\mathbf{P}_{12} = \frac{(1-v)\mathbf{P}_{12,u} + u\mathbf{P}_{12,v}}{u+(1-v)}$$
$$\mathbf{P}_{22} = \frac{(1-v)\mathbf{P}_{22,u} + (1-u)\mathbf{P}_{22,v}}{(1-u)+(1-v)}$$

隣接する2つのグリッド上でCoonsパッチとGregoryパッチを用いて作成した曲面を図4に示す.



(a) Coonsパッチ　(b) Gregoryパッチ

図 4: 隣接する2つのグリッド上でのプロファイル

4.5 Gregory Volume

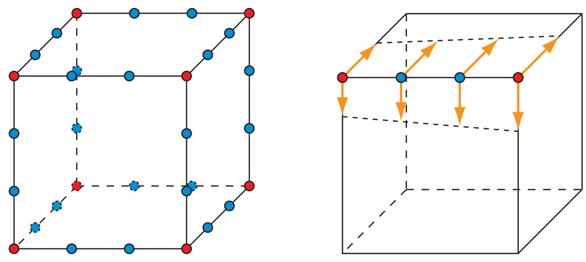
本研究では，3次元流体シミュレーションにおいて連続性の保証された補間法を用いるために，Gregoryパッチを3次元に拡張する．また，これをGregory Volumeと呼ぶこととする．

Gregoryパッチ式(8)をz軸方向に拡張すると以下のGregory Volume式(9)となる．

$$\mathbf{P}(u,v,w) = \sum_{i=0}^{3}\sum_{j=0}^{3}\sum_{k=0}^{3}\binom{3}{i}u^i(1-u)^{3-i} \binom{3}{j}v^j(1-v)^{3-j}\binom{3}{k}w^k(1-w)^{3-k}\mathbf{P}_{ijk}(u,v,w)$$
$$(0 \le u,v,w \le 1) \qquad (9)$$

Gregory VolumeはGregoryパッチをz軸方向に拡張したものであるので，各制御点にあらかじめ与えられている情報は各角の8点での速度と各方向の速度勾配のみである．したがって，他の制御点における速度は補間により求める．まず，各制御点を，セルの辺上の点，セルの面上の点，そしてセル内部の点と3種類に分けて考える．

4.5.1 セル辺上の点



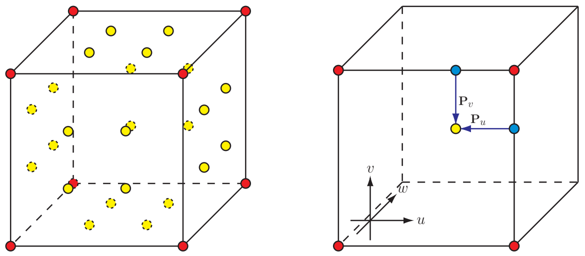
(a)　(b)

図 5: セル辺上の点

セル辺上の点(図5(a))における速度値ついては，辺の両端の点における速度値を元に，3次曲線の式で補間する．速度勾配値については，図5(b)に示すように，辺に対して鉛直方向の成分のみを，線形補間を用いて値を内挿する．

4.5.2 セル面上の点

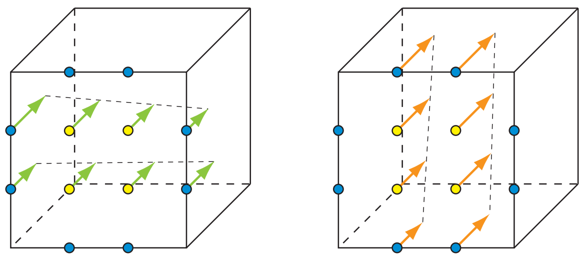
セル面上の点(図6(a))における速度値ついては，各面を2次元として考え，Gregoryパッチの場合と同様に仮の速度を求める．図6(b)に示すように，各点$\mathbf{P}_{i'j'k',u}$に最も近傍のセル辺上の点での速度値$\mathbf{P}_{ijk}$，速度勾配値$\partial \mathbf{P}_{ijk}/\partial u$，$\partial \mathbf{P}_{ijk}/\partial v$，$\partial \mathbf{P}_{ijk}/\partial w$，および距離$u$，$v$，



(a) (b)

図 6: セル面上の点

wの関係から仮の速度値$\mathbf{P}_{i'j'k',u}$，$\mathbf{P}_{i'j'k',v}$，$\mathbf{P}_{i'j'k',w}$を求める．面上の点における値はGregoryパッチと同様に2つを混ぜ合わせることにより決定する．



(a) (b)

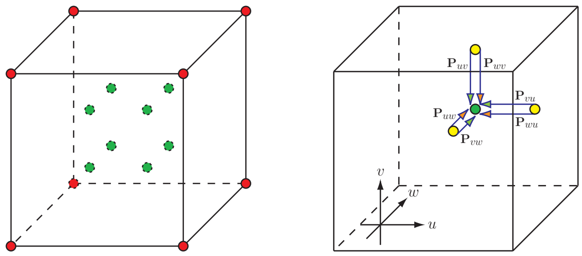
図 7: 面に対して鉛直方向の速度勾配

速度勾配値については，面に対して鉛直方向の成分を線形補間により内挿するが，図7に示すように2通りの方向で線形補間を行う．ここで，x，y，およびz軸方向に沿って補間した速度勾配をそれぞれ，$(\mathbf{P}')_u$，$(\mathbf{P}')_v$，および$(\mathbf{P}')_w$と表す．

4.5.3 セル内部の点

セル内部の点(図8(a))における速度値ついては，図6(b)に示すように，最も近傍の面上の点3つが持つ速度から，セル面上の点に対して行った方法と同様に求める．こうすることでセル内部の各点にそれぞれ6つの仮の速度値が得られる．

次に6つの値を混ぜ合わせることにより，その点での値を決定する．点(1,1,1)と点(2,1,1)における混ぜ合わせの式は以下となる．



(a) (b)

図 8: セル内部の点

$$
\begin{aligned}
\mathbf{P}_{111} &= \frac{vw\mathbf{P}_{111,u} + uw\mathbf{P}_{111,v} + uv\mathbf{P}_{111,w}}{uv + uw + vw} \\
\mathbf{P}_{111,u} &= \frac{w\mathbf{P}_{111,vu} + v\mathbf{P}_{111,wu}}{v + w} \\
\mathbf{P}_{111,v} &= \frac{w\mathbf{P}_{111,uv} + u\mathbf{P}_{111,wv}}{u + w} \\
\mathbf{P}_{111,w} &= \frac{v\mathbf{P}_{111,uw} + u\mathbf{P}_{111,vw}}{u + v} \\
\mathbf{P}_{211} &= \frac{vw\mathbf{P}_{211,u} + (1-u)w\mathbf{P}_{211,v} + (1-u)v\mathbf{P}_{211,w}}{(1-u)v + (1-u)w + vw} \\
\mathbf{P}_{211,u} &= \frac{w\mathbf{P}_{211,vu} + v\mathbf{P}_{211,wu}}{v + w} \\
\mathbf{P}_{211,v} &= \frac{w\mathbf{P}_{211,uv} + (1-u)\mathbf{P}_{211,wv}}{(1-u) + w} \\
\mathbf{P}_{211,w} &= \frac{v\mathbf{P}_{211,uw} + (1-u)\mathbf{P}_{211,vw}}{(1-u) + v}
\end{aligned}
$$

残りの点に関しても同様に混ぜ合わせる．

4.5.4 まとめ

各補間手法で作成したプロファイルの連続性について，図2(a)と図2(b)とを比較すると，B型CIP法ではCIP法よりも拘束条件が2つ増えているため，プロファイルに形成される段が小さくなっているが，端点を除く各セル間での連続性は保証されていない．一方，図4(a)と図4(b)とを比較すると，CoonsパッチとGregoryパッチでは各セル間でのプロファイルの連続性が保証されていることが確認できた．各セル間でのプロファイルの連続性が保証されているということは，補間を行う場合の計算の精度が高くなるということであり，シミュレーションにおいて各セルの速度値が厳しい条件になった場合に，より計算の安定性の向上が期待できる．

5　シミュレーション結果

2次元でのOctreeグリッド上において行った気泡の変形のシミュレーションの結果を図9に示す．図中の赤い線はその点での速度ベクトルを示している．計算に用いたグリッドの分割レベルは最大で8レベル，最小で6レベルである．

各シミュレーションで変化した気泡のディテールに違いが見受けられる．また，CIP法を用いたシミュレーション(図9(b))では，所々に計算の破綻が発生している．これは，図2(a)に示すように，速度値の連続性が保障されていないために発生する，計算値の振動が原因の1つと考えられる．

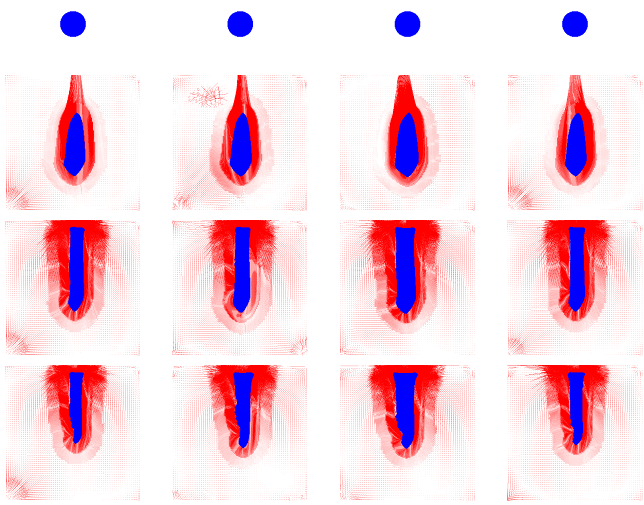


図 9: シミュレーション結果

6　結言

我々は，移流項の計算において，速度勾配の連続性を考慮した補間手法であるGregoryパッチを用いたシミュレーションを行った．また，各種補間法を用いた気泡の変形のシミュレーションを行ったが，どの手法が最も精度が高いかということを，このシミュレーションから解析的に検証することはできない．したがって，今後は解析的に解がはっきりわかっている現象について，Gregoryパッチ，およびGregory Volumeの優位性を検証する必要がある．

参考文献

[1] 千代倉弘明, 工業調査会, ソリッドモデリング CAD/CAMのための基礎技術 (1985)

[2] 姫野武洋, CIP-Level Set 法の考え方 (1999)

[3] 矢部孝, 内海隆行, 尾形陽一, 北森出版, CIP法 (2003)

[4] Foster, N., Metaxas, D., “Realistic Animation of Liquids”, Graphical Models and Image Processing 58, 471-483, 1996.

[5] Losasso, F., Gibou, F., Fedkiw, R., “Simulating Water and Smoke with an Octree Data Structure”, SIGGRAPH 2004, ACM TOG 23, pp.457-462, 2004.

[6] Ogawa, T., “An Efficient Numerical Algorithm for the Treedata Based Flow Solver”, Computational Fluid Dynamics 2000, pp.337-342, 2000.

[7] Song, O., Kim, D., Ko, H., “A Simulation Method for Highly-detailed Fluids”, Proceedings of the International Symposium on Ubiquitous VR 2006, pp.61-64, 2006.

[8] Stam, J., “Real-Time Fluid Dynamics for Games”, Proceedings of the Game Developer Conference, 2003.

[9] Stam, J., “Stable Fluids”, SIGGRAPH 1999, ACM TOG 26, pp.121-128, 1999.

[10] Staniforth, A., Cote, J., “Semi-Lagrangian Integration Schemes for Atmospheric Models - A Review”, Monthly Weather Review 119, 2206-2223, 1991.