

論文 / 著書情報
Article / Book Information

題目(和文)	保存型CIP法を用いた二相流構造・非構造格子数値計算モデルおよび衛生陶器流れへの適用
Title(English)	
著者(和文)	池端昭夫
Author(English)	Akio Ikebata
出典(和文)	学位:博士(工学), 学位授与機関:東京工業大学, 報告番号:甲第10849号, 授与年月日:2018年3月26日, 学位の種別:課程博士, 審査員:肖 鋒,奥野 喜裕,青木 尊之,末包 哲也,長崎 孝夫
Citation(English)	Degree:Doctor (Engineering), Conferring organization: Tokyo Institute of Technology, Report number:甲第10849号, Conferred date:2018/3/26, Degree Type:Course doctor, Examiner:,,,,
学位種別(和文)	博士論文
Category(English)	Doctoral Thesis
種別(和文)	要約
Type(English)	Outline

平成 29 年度
創造エネルギー専攻 博士論文

保存型 CIP 法を用いた
二相流構造・非構造格子数値計算モデル
および衛生陶器流れへの適用
(要約)

指導教員 肖 鋒 教授

東京工業大学大学院
創造エネルギー専攻
池端 昭夫

目次

第 1 章	序論	1
1.1	背景	1
1.2	研究の目的	4
1.3	論文の構成	5
第 2 章	VOF 法に基づく既存の気液界面捕獲計算手法	6
2.1	VOF/PLIC 法	6
2.2	MARS 法	7
2.3	CICSAM 法	7
2.4	MULES 法	9
2.5	THINC 法	9
2.6	CLSVOF 法	10
2.7	各既存手法の比較	11
2.8	提案手法の概要	11
第 3 章	直交格子の気液二相流体計算新手法	12
3.1	移流計算手法 UTI-VSIAM3	12
3.1.1	計算手法の詳細	12
3.1.2	UTI-VSIAM3 の移流計算ベンチマークテスト	18
3.2	気液界面捕獲手法 STAA	20
3.2.1	計算手法の詳細	20
3.2.2	STAA 法の移流計算ベンチマークテスト	22
3.3	表面張力の定式化	26
3.4	フラクショナルステップ法による気液二相流非圧縮性流体解析手法	27
3.4.1	手法の詳細	27
3.4.2	二次元ダムブレイク問題による計算精度検証	29
3.4.3	三次元気泡上昇問題による検証	31



3.4.4	住宅設備機器におけるシミュレーション例	31
3.5	第 3 章のまとめ	36
第 4 章	大規模計算対応のための GPGPU 化および並列化	37
4.1	多相流計算における計算効率化の方策	37
4.1.1	直交格子有限体積法におけるデータ構造の効率化	37
4.1.2	CPU コードの GPU 化による加速化	39
4.1.3	境界不整合領域分割法を用いた計算並列化	39
4.1.4	オーバーラップ通信による通信時間の隠蔽手法	44
4.2	加速化手法による計算効率向上の検証	48
4.2.1	GPU 化による加速効果	48
4.2.2	オーバーラップ通信による効果	48
4.2.3	GPU 化および並列化によるスケーラビリティの検証	48
4.3	大規模計算による衛生陶器シミュレーション結果	52
4.4	第 4 章のまとめ	56
第 5 章	運動量保存性の改良	57
5.1	多相流における運動量保存性の問題	57
5.2	提案計算手法の詳細	59
5.2.1	運動量保存式の定式化	59
5.3	ベンチマークテストによる計算精度検証	62
5.3.1	移動液滴のベンチマークテスト	62
5.3.2	強制対流中の静止液滴のベンチマークテスト	62
5.3.3	三次元のダム崩壊ベンチマークテスト	65
5.4	衛生陶器シミュレーションでの検証	67
5.5	第 5 章のまとめ	69
第 6 章	任意多面体非構造格子の気液二相流体計算手法	70
6.1	VSIAM3 による移流計算手法の多面体格子への拡張	70
6.1.1	計算手法の詳細	70
6.1.2	移流計算精度のベンチマークテスト検証	72
6.1.3	計算量の評価	76
6.2	VSIAM3 を用いた界面捕獲手法の多面体格子への拡張	78
6.2.1	STAA 法のためのレベルセット関数構築	78
6.2.2	多面体格子における STAA 法の実装	79
6.2.3	気液界面捕獲精度のベンチマーク検証	84

6.3	フラクショナルステップ法による気液二相流非圧縮性流体解析手法	87
6.4	GPGPU および並列処理による高速化	89
6.5	数値計算結果	92
6.5.1	キャビティフローテスト	92
6.5.2	ダム崩壊流れテスト	93
6.5.3	衛生陶器シミュレーションへの適用	97
6.6	第 6 章のまとめ	102
第 7 章	結論	104
参考文献		108
謝辞		116



図目次

1.1	The example of a sanitary ware product	1
1.2	The definition of volume of fluid.	3
2.1	Approximation of interface on PLIC.	6
2.2	The definition of surface integrated average for MARS.	7
2.3	The definition of cells for CICSAM reconstruction.	8
3.1	Structured grid	13
3.2	X axis direction flux	15
3.3	Two step SIA calculation method, STEP1(left) and STEP2(right). . . .	16
3.4	Results of 2D advection test with various schemes (after 1 rotation). . .	19
3.5	The definition of neighboring cells.	21
3.6	Result of 2D advection test with UTI-VSIAM3 + STAA (after 1 rotation)	22
3.7	Rudman's problem of notched circle rigid rotation advection (1/2). . . .	23
3.8	Rudman's problem of notched circle rigid rotation advection (2/2). . . .	24
3.9	Result of dam break test	30
3.10	The VOF function of the rising bubble. Displayed are the contours of 0.05, 0.5 and 0.95.	32
3.11	The velocity profile at the top of the bubble.	33
3.12	Result of simulation of a sanitary ware (mesh number = $181 \times 135 \times 267$, CFL=0.48)	34
3.13	Result of simulation of a kitchen (mesh number = $181 \times 223 \times 160$, CFL=0.48)	35
4.1	A example of multiphase simulation of a sanitary ware.	37
4.2	The difference between tetrahedral (upper) and Cartesian (lower) grids.	38
4.3	Mesh numbers of unstructured grids.	38
4.4	Block decomposition method.	40

4.5	Block cyclic decomposition method.	40
4.6	Non-conformal boundary domain decomposition method.	41
4.7	Difference from general domain decomposition method. (upper : general DDM, lower : presented method)	43
4.8	Pseudo-code for message passing of the sleeve meshes.	44
4.9	Packing of send buffer.	45
4.10	Exploring to sleeve meshes from receive buffer.	45
4.11	The algorithm of overlapped communication in GPU code.	47
4.12	The result of measuring load balance of each GPU.	49
4.13	The test model of the sanitary ware (Ex. $2 \times 2 \times 2$ domain decomposition).	50
4.14	The result of performance measurement using TSUBAME supercomputer.	51
4.15	A simulation result of a sanitary ware using TSUBAME supercomputer.	53
4.16	The water level at a probe point. Displayed are the numerical results computed with 1.5mm and 0.5mm grid resolutions respectively in comparison with the experiment measurement.. . . .	54
4.17	Another simulation result of a sanitary ware using TSUBAME supercomputer.	55
5.1	The error of VOF and velocity advection equation. The arrows indicate the velocity field.	58
5.2	The correction of density.	60
5.3	The result of benchmark test of moving particle in the static atmosphere without surface tension. Upper panel is the result of the calculation method where momentum equations are solved by UTI-VSIAM3, lower is that of VSIAM3-CM.	63
5.4	The result of benchmark test of the static particle in the air flow without surface tension. Upper is the result of the calculation method where momentum equations are solved by UTI-VSIAM3, lower is that of VSIAM3-CM.	64
5.5	The result of dam break benchmark test. Left is the result of UTI-VSIAM3. Right is that of VSIAM3-CM.	66
5.6	The result of numerical simulation of a sanitary ware. Upper is the result of UTI-VSIAM3, lower is that of VSIAM3-CM.	68
6.1	Normal and tangential velocity.	72



6.2	Initial profile of Rudman benchmark test.	74
6.3	The results of TVD and VSIAM3s after one revolution.	75
6.4	Anti-diffusion vector for STAA in 2D.	80
6.5	Anti-diffusion vector for STAA in 3D.	83
6.6	Searching three neighboring points for 3D-STAA.	83
6.7	Evaluation method of L1 error.	84
6.8	The results of STAA and other existing methods after one revolution. . .	86
6.9	The results of PCG calculation scalability test on a sanitary ware product model. System 1: GeForce GTX 690, 2GPU / node $\times$ 4 nodes. System 2: Tesla K20x, 8GPU / node $\times$ 1 node. System 1 is “Thin node cluster”, while system 2 is “Fat node system”. The numbers of cells, faces and nodes are 2,279,568, 6,925,847 and 2,368,243, respectively.	91
6.10	The results of lid-driven vortex flow benchmark test. Upper and lower panels are the velocity profiles respectively for $Re = 1000$ and 5000 cases.	92
6.11	The initial profile of dam break benchmark test. Three solid lines in upper figure indicate $F = 0.05, 0.5$ and 0.95 contours. Lower figure shows the mesh structure near the corner of liquid.	94
6.12	The results of dam break benchmark test. Three solid lines indicate $F = 0.05, 0.5$ and 0.95 contours. Four panels show the results after 0.10, 0.20, 0.30 and 0.50 second, respectively.	95
6.13	The comparison of results of dam break benchmark test between present and Martin & Moyce’s experiment. Upper is the profile of the front position of water, where $T_h \equiv t\sqrt{2g/a}$, $D \equiv x/a$. Lower is the profile of the height of water column, where $T_h \equiv t\sqrt{2g/(2a)}$, $H \equiv y/(2a)$	96
6.14	The simulation model of a sanitary ware.	97
6.15	The results of thin water film flow on the surface of a sanitary ware. . .	99
6.16	The results of thin water film flow on the surface of a sanitary ware. . .	100
6.17	The comparison of the result of present method (middle) and our existing method using Cartesian grid (lower) with the experimental result (upper). . .	101

表目次

1.1	Comparison of the characteristics of various numerical methods for multiphase flow. “ $\triangle$ ” in this table means “relatively inferior”.	4
3.1	Comparison of L_1 error with other advection calculation methods.	24
4.1	Comparison of the performance between CPU and GPU.	49
6.1	Setting of Rudman benchmark test.	73
6.2	Comparison of L_1 error between TVD and VSIAM3.	75
6.3	Total number of double precision floating operations on TVD code . . .	77
6.4	Total number of double precision floating operations on VSIAM3 code .	77
6.5	Comparison of L_1 error among STAA and various existing methods. . .	86
6.6	Comparison of performance between GPU and CPU calculations. GPU calculation was conducted by one NVIDIA GeForce GTX 690 device, while CPU calculation was conducted by one Intel Xeon 3.6 GHz core with the compile command “nvcc -O3 -arch=sm_20”. The numbers of cells, faces and nodes are 1,447,313, 4,390,325, 1,496,242, respectively.	89



第 1 章

序論

1.1 背景

衛生陶器 (図 1.1) をはじめとする水回りの住宅設備の製品開発において，節水化は大きな設計ポイントの一つとなってきた。衛生陶器の以前の一般的な製品では，一回の洗浄に約 13 リットルを消費していたが，最近の最も節水性の高い製品ではわずか 3.8 リットルと急激な節水化が進んでいる。したがって衛生陶器の設計においては，高い洗浄性能を保持しつつ目標の低洗浄水量を達成するための製品形状および洗浄水量制御の最適化が大きな技術課題となっている。

製品形状および洗浄水量の最適化設計では，従来は実験評価に頼っていたが，非常に多くの試作・試験が必要のため多額の試作費用および開発期間の長期化という問題があった。そこで当社では，実機試作試験の代替となる気液二相流シミュレーション技術開発を進めてきた。衛生陶器の二相流においては，水を溜めている便鉢部の陶器表面の薄膜流れを精度良くシミュ



図 1.1 The example of a sanitary ware product

レーションすることが、付着汚物の洗浄評価試験の置き換えを実現するための必要条件である。特に洗浄水量 3.8L 程度の超節水型の衛生陶器製品においては、便鉢部陶器表面の水膜はわずか数ミリ程度となる場合があり、二相流シミュレーションにおける格子解像度の限界から、高精度かつ高効率な混相流シミュレーション技術の開発が必要である。

水と空気の流れをはじめとする気液二相流体解析手法は、近年急速に発展してきている。水と空気の界面を計算する方法として、Smoothed Particle Hydrodynamics (SPH) 法 [1][2][3] や Moving Particle Semi-implicit (MPS) 法 [4][5][6] などの粒子法、格子ボルツマン法 [7][8][9]、界面追跡法 [10][11]、Front-Tracking 法 [12][13][14]、レベルセット法 [15][16][17] および VOF 法 [18][19][20] など様々な数値計算手法が提案されてきている。

有限体積法

有限体積法 [38] は、Navier-Stokes 流体方程式を体積積分形に変形し、各格子における物理量の積分平均値を代表値として保持することにより、積分形 Navier-Stokes 方程式の計算において物理量の保存性を保証する手法である。Navier-Stokes 方程式の移流項、粘性項などに現れる発散項 $\nabla \cdot \mathbf{v}$ は有限体積格子において体積積分することにより以下の式で計算される。

$$\nabla \cdot \mathbf{v}_i \equiv \frac{1}{V_i} \sum_f^{N_i} s_f \mathbf{n}_f \cdot \mathbf{v}_f \quad (1.1)$$

ここで添え字 f は着目している格子 i と隣接格子 j との界面を表し、 s_f は f の面積、 $\mathbf{n}_f$ は f の外向き法線ベクトル、 $\mathbf{v}_f$ は f における流束 (=フラックス) を表す。また V_i は格子 i の体積である。界面 f におけるフラックスを通して隣接格子どうして物理量を交換するため、保存性が保証される。

有限体積法を用いた気液二相流シミュレーションでは、一般に気液界面は各格子において定義される流体率 F によって間接的に捕獲される。図 1.2 に示すように、 F は格子の空間体積における液体体積の占める割合を 0.0 ~ 1.0 の範囲で表したものである。流体率は Volume Of Fluid (VOF) 関数ともよばれる [18]。時々刻々変化する F は、各格子の流速ベクトル $\mathbf{u}$ および時刻 t を用いて、次の移流方程式により計算される。

$$\frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0 \quad (1.2)$$

各格子の流体率 F が、流速 $\mathbf{u}$ に沿って移動していくことを表している。この移流方程式を精度よく解くことが、気液二相流シミュレーションにおける重要な課題である。

VOF 法を用いた有限体積法では、気泡巻き込みなどの複雑な界面変形にも対応できるため、市販流体解析コードなどでも広く利用されてきた。VOF 法における大きな課題として、VOF 関数の計算において気液界面で発生する数値拡散を抑制し精度よく界面を捕獲すること

が挙げられてきたが, Piecewise Linear Interface Calculation (PLIC) 法 [30][41][42], Multi-interface Advection and Reconstruction Solver (MARS) 法 [43], Tangent of Hyperbola Interface Capturing (THINC) 法 [44][45][46] など様々な有力手法が開発され, 格子解像度の範囲で十分信頼性の高い自由界面捕獲計算ができるようになった.

一般に VOF 法は VOF/SLIC(Simple Line Interface Calculation)[19] および VOF/PLIC のような気液界面の幾何的構築の手法を指すことが多いが, ここでは VOF 関数を移流させて気液界面を捕獲する手法全般を VOF 法とよぶ. VOF 法の中で VOF/PLIC などの幾何的再構築手法を幾何的 VOF 法 (= Geometric VOF method) とよび, その他の手法を代数的 VOF 法 (= Algebraic VOF method) とよぶこととする.

次に, 気液二相流では Navier-Stokes 方程式における移流計算にも注意を払う必要がある. 水と空気の 1000 倍の密度差において密度 $\times$ 流速, すなわち運動量に関する移流方程式を精度良く解くことは, 液滴, 気泡および薄膜流れなどの正しい挙動を導くための重要な条件である. 運動量に関する移流方程式は, Total Variation Diminishing (TVD) 条件に基づく二次精度スキーム [47][48] が広く用いられている. 一方, より高次精度の計算手法として Discontinuous Galerkin (DG) 法 [49][50][51] の研究が進められているが, 計算の安定性と計算量の観点からは多くの課題がある. 高次精度かつ安定的な計算手法として Essentially Non-Oscillatory polynomial interpolation (ENO) 法 [52][53] や WENO 法が挙げられるが, 非構造格子への実装性に課題がある.

計算手法の比較

以上の計算手法の長所および短所を表 1.1 にまとめる. 衛生陶器流体シミュレーションにおいては, まず流体の保存性が重要であり, 保存性が満たされない場合に薄膜, 飛沫, 気泡などの消失が発生する懸念がある. また計算格子を細分化して計算精度を向上させるためには計算

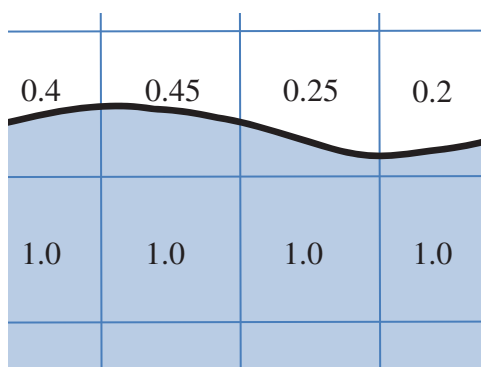


図 1.2 The definition of volume of fluid.

表 1.1 Comparison of the characteristics of various numerical methods for multiphase flow. “△” in this table means “relatively inferior”.

	Conservative- ness	Calculation cost	ρ_l/ρ_g ≥ 1000	Topology change of interface
SPH	-	△	-	-
MPS	-	△	-	-
LGM	-	△	-	-
LBM	-	-	△	-
FEM	△	-	-	Not available
F-T	△	-	-	Not available
LSM	△	-	-	-
FVM (VOF)	-	-	-	-

量がなるべく少ない手法が適している．さらに空気と水の混在する流れであるため，気液の密度比は 1000 倍以上を安定的に計算できることが必要条件である．また巻き込み気泡などの気液界面のトポロジー変化に対応できることが重要である．以上の条件を全て満たす手法として，有限体積法に基づく VOF 法が衛生陶器シミュレーションでは好適と考えられる．格子ボルツマン法も有力な手法であるが，空気と水の密度比 1000 倍に対応するためには現時点では圧力ポアソン方程式を解く必要があり，本来は陽的計算で全ての計算を完結する格子ボルツマン法の特徴を活かせない．高密度比への対応は今後の重要な研究課題といえる．

1.2 研究の目的

複雑化した衛生陶器の製品開発において，節水化などの製品性能向上および開発期間の短縮やコスト削減を目的とし，多相流シミュレーション手法を開発する．これまでの市販ソフトウェアでは，計算精度や計算速度において当社の要求性能を満たしていなかった．そこで衛生陶器の水の流れにおいて，無数の気泡，飛沫および薄膜流れを高精度にシミュレーションするため，1mm 以下レベルの高精細な大容量メッシュを用いて高速に精度良く計算できる有限体積法ベースの多相流計算ソフトウェアを新たに開発する．既存手法に対して計算精度のさらなる高精度化を図るため，気液界面捕獲および Navier-Stokes 移流項の高精度な計算手法を開発する．計算格子については直交格子および非構造格子の双方を検討する．また大容量メッシュを高速に計算するため，GPGPU 技術による加速化とメモリ分散並列による大規模並列計算に対応できるフレームワークを構築する．

1.3 論文の構成

第 2 章では、有限体積法における VOF 法について既存手法を再調査し、それらの問題点を洗い出すことで、新しく提案する計算手法の目標レベルを設定する。

第 3 章では、構造格子においてマルチモーメント法に基づく独自の移流計算手法を核とした有限体積法フレームワークを提案する。移流計算ベンチマークや二・三次元流れ計算テストおよび衛生陶器シミュレーションを通して、提案手法が妥当であることを示す。

第 4 章では、第 3 章で提案したフレームワークを GPGPU 加速化および大規模並列化により大幅に高速化する手法を述べる。新たに提案する領域分割法と GPU 化手法の組み合わせにより衛生陶器大容量メッシュの計算において飛躍的に計算速度を向上できることを実証する。

第 5 章では、衛生陶器シミュレーションにおける洗浄時の飛沫の動きの精度向上を目的とし、保存型運動量解法に基づく移流計算手法の改良方法を提案する。運動量保存性を向上する安定的な移流計算アルゴリズムを見出し、ベンチマークテストおよび衛生陶器シミュレーションへの適用でその有効性を示す。

第 6 章では、衛生陶器表面の薄膜流れの精度向上を目的に、第 3 章の構造格子のフレームワークを任意多面体非構造格子に拡張する新しい移流計算手法を提案する。計算精度や GPGPU 加速効率、並列化効率を検証するとともに、衛生陶器実製品の流れに適用し、本手法の有効性を実証する。



第 2 章

VOF 法に基づく既存の気液界面捕獲計算手法

2.1 VOF/PLIC 法

PLIC 法は，格子ごとに三次元平面方程式で気液界面を近似する手法である．図 2.1 の三次元直交格子において，曲面の気液界面を図のように平面方程式 p

$$p : n_x x + n_y y + n_z z = \rho \quad (2.1)$$

により近似する．この平面 p は，法線ベクトル $\mathbf{n} = (n_x, n_y, n_z)$ を持ち，切片 ρ の平面を表している．PLIC では，まず何らかの方法で法線ベクトル $\mathbf{n}$ を計算した後，平面 p が直交格子を

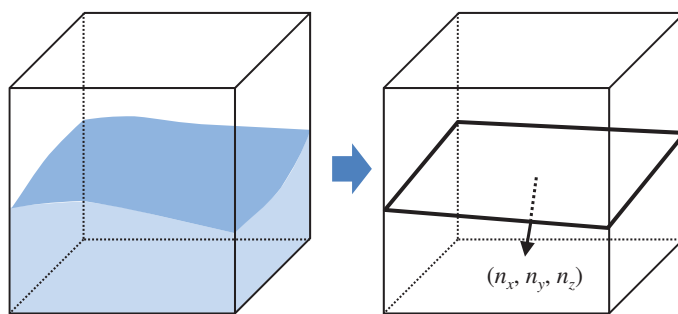


図 2.1 Approximation of interface on PLIC.

切り取る体積 V_r において

$$\frac{V_r}{V_i} = F_i \quad (2.2)$$

が成り立つように切片 ρ を求める． V_i および F_i は式 (1.1) および (1.2) で現れた，格子 i の体積および流体率である．法線ベクトル $\mathbf{n}$ が精度良く求められていれば，図 2.1 よりこの再構築は妥当な近似となることが分かる．この再構築方法，すなわち ρ を計算する有力な手法として，Rider らの手法 [41] と Zaleski らの手法 [42] がある．

2.2 MARS 法

VOF/PLIC の三次元計算量が多いことから，より計算量を低減する方法が提案された．MARS[43] は三次元の補間平面方程式を再構築する代わりに，図 2.2 のように格子断面の面積分平均値に関する一次元補間方程式を構築することで気液界面を捕獲する方法である．MARS では，一次元補間方程式は簡易な一次式で表す．

2.3 CICSAM 法

VOF/PLIC や MARS では格子体積ごとに再構築計算を行うが，そもそも有限体積法では格子界面でのフラックスを計算すればよいことに着目し，格子体積での再構築を行うことなくフラックス計算する方法が可能である．図 2.3 のように，格子界面 f での上流側格子を C ，下流側を D ， C のさらに上流側の格子を U とおき，この 3 セルにまたがる一次元の補間方程式を考える．格子界面 f での補間方程式の値 F_f がフラックスとして用いられる．

$D \rightarrow C \rightarrow U$ 方向を x 軸方向と取ったときに，Compressive Interface Capturing Scheme

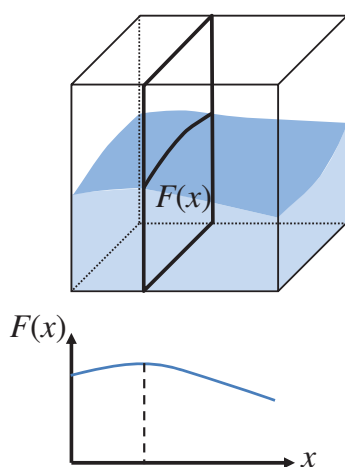


図 2.2 The definition of surface integrated average for MARS.

for Arbitrary Meshes (CICSAM) [61] では、この 3 格子にまたがる関数を以下のように正規化する．

$$f'(x) = \frac{F(x) - F_U}{F_D - F_U} \quad (2.3)$$

気液界面が x 軸と直交する場合、この条件でのフラックスは、VOF/SLIC でのドナーアクセプター法と同様に計算できる．ただし VOF/SLIC の場合は流体率 0 および 1 で境界制限されていたが、CICSAM の場合は上流側の値 F_U と下流側の値 F_D で境界制限されることが大きな違いとなる．

一方気液界面と x 軸が直交しない場合は、 $f'(-1/2) = 1$ 、 $f'(3/2) = 0$ および $f'(1/2) = f'_C$ の条件より、二次補間関数を作成する．

$F'_{f,1}$ および $F'_{f,2}$ から F'_f を計算する． x 軸方向と気液界面法線方向との角度 θ_f を求め、 $\gamma_f = (\cos 2\theta_f + 1)/2$ により $F'_{f,1}$ と $F'_{f,2}$ の重みを算出する．正規化フラックスは

$$F'_f = \gamma_f F'_{f,1} + (1 - \gamma_f) F'_{f,2} \quad (2.4)$$

により計算される．最終的に正規化フラックス F'_f はもとのフラックス F_f に逆写像される．

$$F_f = F'_f(F_D - F_U) + F_U \quad (2.5)$$

この手法では $F'_{f,1}$ が気液界面の圧縮のはたらきを行い、数値拡散を防止するとともに、高精度の $F'_{f,2}$ により気液界面の形状精度を保持する効果がある．また写像関数により上流格子の値と下流格子の値で境界制限することで、オーバーシュートやアンダーシュートを防止していることが分かる．

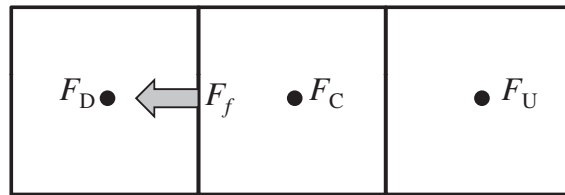


図 2.3 The definition of cells for CICSAM reconstruction.

2.4 MULES 法

MULES ソルバーは、OpenFOAM の多相流ソルバー “interFOAM” で用いられている気液界面捕獲計算手法である [64]。MULES は CICSAM と同様に界面格子におけるフラックスを直接計算する手法であり、各体積格子での気液界面再構築は行わない。図 2.3 において、 f における基本的なフラックスは TVD スキーム [47] により計算する。

$$F_{f,0} = F_C + \frac{1}{2}\psi(r_f)(F_D - F_C) \quad (2.6)$$

$$r_f = \frac{F_C - F_U}{F_D - F_C} \quad (2.7)$$

$\psi(r_f)$ は TVD スキームにおけるリミッタである。このままでは気液界面近傍での数値拡散が発生するため、フラックスの計算において気液界面圧縮項を追加する。

$$F_f = F_{f,0} + F_{f,a} \quad (2.8)$$

$$F_{f,a} \equiv \cos \theta_f F_f^a (1 - F_f^a) \quad (2.9)$$

$$F_f^a \equiv F_C + \frac{1}{2}\psi'(F_C, F_D)(F_D - F_C) \quad (2.10)$$

$$\psi'(a, b) \equiv \min(\max(1 - \max(q(a), q(b)), 0), 1) \quad (2.11)$$

$$q(a) \equiv (1 - 4a(1 - a))^2 \quad (2.12)$$

$\psi'(a, b)$ で定義されるリミッタを用いて、 F の不連続性を考慮した TVD スキームにより得られる f での補間値 F_f^a から、数値拡散補正フラックス $F_{f,a}$ を計算している。 $F_{f,a}$ の定義から、 F_f^a が 0 または 1 の場合は補正は行われず。また $F_{f,a}$ は θ_f が 90 度に近づくとき 0 に近づく。すなわち気液界面方向ベクトルが移流計算方向と直交する場合に補正されなくなり、妥当な定式化であることが分かる。

2.5 THINC 法

近似 Heaviside 関数 $H(\mathbf{x})$ を双曲線関数により表し、 H を直接移流計算する方法を THINC 法 [44] とよぶ。

$$H(\mathbf{x}) = \frac{1}{2} (1 + \tanh(\beta(G(\mathbf{x}) + d))) \quad (2.13)$$

β は界面の勾配を表し、 β を大きくしていくと流体率 $F(\mathbf{x})$ に近づいていく。 $G(\mathbf{x})$ はレベルセット関数である。 $H(\mathbf{x})$ は $F(\mathbf{x})$ と異なり連続関数であるため、格子解像度が十分であれば勾配値は $H(x)$ を中心差分することにより得られる。また各セルの再構築では PLIC のような平面方程式ではなく曲面を表す二次補間式を利用できるため、計算精度の向上が期待できる。



MTHINC 法 [65] では、三次元の各格子の流体率 $F_{i,j,k}$ に $H(\mathbf{x}_{i,j,k})$ が一致するように d を再構築する。MTHINC では

$$\frac{1}{V_{i,j,k}} \int_{V_{i,j,k}} H(\mathbf{x}) dx dy dz = F_{i,j,k} \quad (2.14)$$

の式を直接解く。tanh 関数を直接積分するのは非常に複雑になるため、ガウスルジャンドル数値積分を利用する。計算精度の観点より、界面法線ベクトルに最も近い軸のみ直接積分を利用する実装においては、三角関数を係数に含む 4 次方程式を計算する必要があり、極めて計算は複雑となる。また THINC/QQ 法 [66] では、全てガウスルジャンドル積分を用いるとともに、再構築計算に反復法を利用し、定式化の簡易化を図っている。ただし反復法の計算量が多くなる問題がある。

THINC/SW[45] および THINC/WLIC 法 [46] は、三次元の再構築を行うことなく、CIC-SAM や MULES のように直接フラックスを計算する手法である。すなわち一次元の双曲線関数

$$H(x) = \frac{1}{2} (1 + \tanh(\beta(x + d))) \quad (2.15)$$

により格子界面の法線方向の分布を表し、フラックスを計算できる。THINC/SW では、気液界面法線ベクトルと格子界面法線ベクトルの関係より β を変化させることでフラックスを計算できる。一方 THINC/WLIC ではフラックスは

$$F_f \equiv wH(x_f) + (1 - w)F_{f,upwind} \quad (2.16)$$

$$w = \frac{|n_j|}{|n_x| + |n_y| + |n_z|} \quad (2.17)$$

で簡易に計算する。重み w は j 軸方向の気液界面法線ベクトル成分の大きさで決まる。 j 軸方向は格子界面法線ベクトル方向を示す。THINC/SW および THINC/WLIC は定式化は異なるものの計算量および計算精度においてほぼ同等である。

2.6 CLSVOF 法

CLSVOF 法 [67][68][69] はレベルセット法と VOF 法を併用したアルゴリズムである。すなわち流体率 F の移流計算は VOF 法で行い、レベルセット関数 G の移流計算は一般的な移流計算手法を用い、 $F = 0.5$ および $G = 0$ の気液界面の適合を行う。VOF 法の計算時に必要な界面法線ベクトルの計算は G を用いることで、法線ベクトルの精度向上を図る。[67][68][69] 以外にも様々な定式化が考えられる。

2.7 各既存手法の比較

VOF/PLIC における Rider らの方法は様々な格子への適用性に優れるが, Brent 法を用いた繰り返し計算による再構築の計算量が多い. 一方 Zaleski らのアルゴリズムはより洗練され計算量が比較的少ないが, 直交格子以外の格子への適用性に課題がある.

MARS は VOF/PLIC と比較して計算量は大幅に低減できるが, 一次関数による補間方程式を利用するため, 計算精度は VOF/PLIC より低いことが課題である. CICSAM および MULES も計算量は VOF/PLIC より大幅に少ないものの, 計算精度は VOF/PLIC にはおおよばないとされている.

THINC 法では MTHINC や THINC/QQ などの多次元補間関数の再構築により, VOF/PLIC と同等の計算精度が得られるものの, 計算式は複雑であり, 計算量は VOF/PLIC を上回ると考えられる. THINC/SW や THINC/WLIC では計算量が少ないが計算精度は若干劣る.

CLSVOF 法は様々な VOF 法とレベルセット移流計算法とを組み合わせることが可能なため, 計算精度の評価は難しい. VOF 値とレベルセット値を両方移流計算し適合を行うため, 計算量が多くなると考えられる. レベルセット関数を用いて表面張力の計算が容易になる点が利点といえる.

2.8 提案手法の概要

GPGPU 化および並列化による高速化が適用可能な移流計算手法の開発を目標とする. 計算精度についてはマルチモーメント利用による高次精度補間関数を活用して, 既存の代数的 VOF 法よりも高精度な手法の開発を目標とする. VOF/PLIC は定式化が他の手法よりも大幅に複雑で計算量も多いため, VOF/PLIC よりも簡易な定式化かつ同等の計算精度を持つ代数的 VOF 法の構築を目指す.



第 3 章

直交格子の気液二相流体計算新手法

3.1 移流計算手法 UTI-VSIAM3

3.1.1 計算手法の詳細

以下のような二次元の移流方程式を考える．

$$\frac{\partial \phi}{\partial t} + u \frac{\partial \phi}{\partial x} + v \frac{\partial \phi}{\partial y} = 0. \quad (3.1)$$

式の中で x および y は二次元空間における X 軸および Y 軸の座標値を表し， u および v は流速ベクトル $\mathbf{u}$ の X 軸成分および Y 軸成分を表す． ϕ は，移流される物質質量を表す．移流とは，ある物質質量 ϕ が流速ベクトル $\mathbf{u}$ に乗って時刻 t に従い移動していく現象を表す．この式は，以下のように書き換えることができる．

$$\frac{\partial \phi}{\partial t} + \frac{\partial u \phi}{\partial x} + \frac{\partial v \phi}{\partial y} = \phi \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right). \quad (3.2)$$

流速が低マッハ数のおだやかな流れにおいては，水や空気のような一般的な流体は非圧縮性の傾向を表す．非圧縮性とは流れによって局所的に流体が縮められたり膨張したりすることがない理想状態をいう．二次元の非圧縮性流れにおいては，以下の式

$$\nabla \cdot \mathbf{u} = \frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.3)$$

が成立する．これは，流体のある微小検査体積 (=コントロール・ボリューム) において，体積 $V = dx dy$ における単位時間あたりの体積変化 $\Delta V = (u + du)dy - udy + (v + dv)dx - vdx$ が変化しない，すなわち $\Delta V/V = 0$ から導かれる．したがって式 (3.2) の右辺は 0 となり，次の輸送方程式と等価になる．

$$\frac{\partial \phi}{\partial t} + \frac{\partial u \phi}{\partial x} + \frac{\partial v \phi}{\partial y} = 0 \quad (3.4)$$

いま、図 3.1 に示す二次元の直交格子を用いて有限体積法により式 (3.4) の輸送方程式を解くことを考える。有限体積法では、式 (3.4) は以下のように離散化される。

$$\frac{\phi_{i,j}^{(n+1)} - \phi_{i,j}^{(n)}}{\Delta t} = - \frac{\mathcal{F}_{i+1/2,j}^x - \mathcal{F}_{i-1/2,j}^x}{\Delta x} - \frac{\mathcal{F}_{i,j+1/2}^y - \mathcal{F}_{i,j-1/2}^y}{\Delta y} \quad (3.5)$$

ここで n は現在の時刻 $t = T$ を表し、 $n + 1$ は次の時刻 $t = T + \Delta t$ を表す。 Δx および Δy は各々 X 軸方向および Y 軸方向のメッシュ間隔を表す。また $\mathcal{F}^x$ および $\mathcal{F}^y$ は各々 X 軸および Y 軸方向のフラックス $u\phi$ および $v\phi$ を表す。フラックスは流束ともよばれ、物理量 ϕ がある面から流速 u に乗って侵入する現象を表す。この定式化により、有限体積法ではスカラー値 ϕ の保存性が保証されることは明らかである。有限体積法では、この流束をどのように定式化するかで、各々の数値計算手法における精度や安定性に大きな影響をおよぼす。

Constrained Interpolation Profile-Conservative Semi-Lagrangian (CIP-CSL) 法は有限体積法に基づく、高精度かつ効率的な数値計算手法である [70][71][72][73][74]。CIP-CSL 法の他の既存のスキームにはない最も大きな特徴は、複数のモーメント値を用いていることであり、それらの値を数値計算変数として同時に更新していくところにある。すなわち、既存の有限体積法ではセル中心値=体積積分平均値を一つ、モーメントとして保持するシングルモーメント法が用いられるのに対し、CIP-CSL 法では体積積分平均値の他にセル界面の面積分平均値なども独立変数として保持され、それらは全て各時刻ステップにおいて移流方程式を満たすように更新されていくことが、他の手法とは大きく異なる点である。

多次元場において CIP-CSL 法を実装する場合は、有限体積法におけるフラックス計算の複雑性を排除するため、通常は次元分割手法が用いられる。多次元流れにおいては、セル界面からのフラックスは法線方向ではなく斜め方向から侵入してくるため、法線方向の一次元分布を仮定してフラックスを計算するだけでは不十分である。そこで次元分割手法を用いることで、斜め方向のフラックスを考慮する必要がなくなり、一次元計算を次元数回分繰り返すだけで多次元

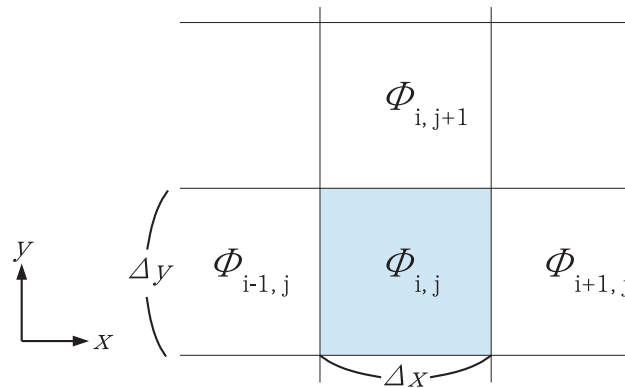


図 3.1 Structured grid



を実装できる．CIP-CSL 法では，オリジナルの手法として点値，線積分平均値，面積分平均値および体積積分平均値のすべてのモーメントを用いる次元分割手法 [72][73][74] が提案されたが，アルゴリズムをより簡易化した手法が研究されてきている．簡易化された手法では，三次元計算において体積積分平均値 (Volume Integrated Average = VIA) と面積分平均値 (Surface Integrated Average = SIA) の二種類のモーメントのみを使用する．その数値計算手法は Volume/Surface Integrated Average based Multi-Moment Method (VSIAM3)[75][76][77] とよばれる．二次元の場合，VIA および SIA は以下のように定義される．

$$\phi_{i,j}^v \equiv \frac{1}{\Delta x \Delta y} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \phi(x,y) dx dy \quad (3.6)$$

$$\phi_{i+\frac{1}{2},j}^x \equiv \frac{1}{\Delta y} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \phi(x_{i+\frac{1}{2}},y) dy \quad (3.7)$$

$$\phi_{i,j+\frac{1}{2}}^y \equiv \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \phi(x,y_{j+\frac{1}{2}}) dx \quad (3.8)$$

次元分割型 VSIAM3 では，まず X 軸方向の移流計算を行う．セル (i,j) において時刻 n における既知の $\phi_{i,j}^v$ ， $\phi_{i+1/2,j}^x$ および $\phi_{i-1/2,j}^x$ を用いて二次補間関数 $ax^2 + bx + c$ の係数 a ， b および c を決定する．この再構築された補間式を元に，時刻 $n+1$ における $\phi_{i,j}^v$ および $\phi_{i+1/2,j}^x$ を決定する．このプロセスは，一次元保存型 CIP 法とよばれる [70]．一方，時刻 $n+1$ の $\phi_{i,j+1/2}^y$ をこの手法から計算することはできないので，別途両側の $\phi_{i,j}^v$ および $\phi_{i,j+1}^v$ の変化量を空間補間することで $\phi_{i,j+1/2}^y$ を更新する．この方法を Time Evolution Converting (TEC) [77] とよぶ． Y 軸方向の移流計算も X 軸のときと同様に行う．

ここで我々は，既存の次元分割型 VSIAM3 ではなく，非次元分割手法により SIA の更新およびフラックスの計算を行う計算手法を提案する．この非次元分割型手法では，SIA は式 (3.1) に基づきセミラグランジェ手法で計算され，VIA は式 (3.2) の有限体積法の公式により更新される．

図 3.2 に示すフラックスの計算をガウス・ルジャンドル積分により近似計算する．例えばコントロールボリュームの右面のフラックスは，

$$\begin{aligned} \mathcal{F}_{i+\frac{1}{2},j}^x &= \frac{u\Delta t}{2} \left\{ \phi^{x(n)}(x_{i+\frac{1}{2}} - g_1 u\Delta t, y_j - g_1 v\Delta t) \right. \\ &\quad \left. + \phi^{x(n)}(x_{i+\frac{1}{2}} - g_2 u\Delta t, y_j - g_2 v\Delta t) \right\} \end{aligned} \quad (3.9)$$

により計算される．ここで g_1 ， g_2 はガウス・ルジャンドル積分の積分点位置を表し， $g_1 \equiv (3 - \sqrt{3})/6$ ， $g_2 \equiv (3 + \sqrt{3})/6$ である．同様に，コントロールボリュームの上面のフラックスは

$$\begin{aligned} \mathcal{F}_{i,j+\frac{1}{2}}^y &= \frac{v\Delta t}{2} \left\{ \phi^{y(n)}(x_i - g_1 u\Delta t, y_{j+\frac{1}{2}} - g_1 v\Delta t) \right. \\ &\quad \left. + \phi^{y(n)}(x_i - g_2 u\Delta t, y_{j+\frac{1}{2}} - g_2 v\Delta t) \right\} \end{aligned} \quad (3.10)$$

により計算される．なお，上記のガウス・ルジャンドル積分は，シンプソン則に変えても同等の計算精度が得られる．

$$\begin{aligned}\mathcal{F}_{i+\frac{1}{2},j}^x &= \frac{u\Delta t}{6} \left\{ \phi^{x(n)}(x_{i+\frac{1}{2}}, y_j) \right. \\ &\quad + 4\phi^{x(n)}(x_{i+\frac{1}{2}} - \frac{1}{2}u\Delta t, y_j - \frac{1}{2}v\Delta t) \\ &\quad \left. + \phi^{x(n)}(x_{i+\frac{1}{2}} - u\Delta t, y_j - v\Delta t) \right\}\end{aligned}\quad (3.11)$$

$$\begin{aligned}\mathcal{F}_{i,j+\frac{1}{2}}^y &= \frac{v\Delta t}{6} \left\{ \phi^{y(n)}(x_i, y_{j+\frac{1}{2}}) \right. \\ &\quad + 4\phi^{y(n)}(x_i - \frac{1}{2}u\Delta t, y_{j+\frac{1}{2}} - \frac{1}{2}v\Delta t) \\ &\quad \left. + \phi^{y(n)}(x_i - u\Delta t, y_{j+\frac{1}{2}} - v\Delta t) \right\}\end{aligned}\quad (3.12)$$

この場合， $\phi^{x(n)}(x_{i+\frac{1}{2}}, y_j)$ ， $\phi^{y(n)}(x_i, y_{j+\frac{1}{2}})$ は既知の値であるので，求めなければならない SIA の値は，ガウス・ルジャンドル積分のときと同じ 4 つである．ただしガウス・ルジャンドル積分のときの (g_1, g_2) 各点における重み $(1, 1)$ とは異なり，シンプソン則では $(0, 1/2, 1)$ の各点における重みは $(1, 4, 1)$ と不均一になっており， $1/2$ の点における重みが大きいに注意する．

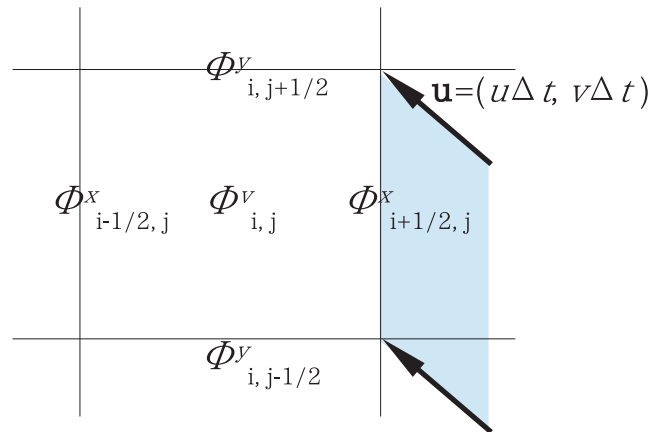


図 3.2 X axis direction flux



VIA は有限体積法の定式化に沿って以下のように更新される.

$$\begin{aligned} \phi_{i,j}^{v(n+1)} = & \phi_{i,j}^{v(n)} - \left(\frac{\mathcal{F}_{i+\frac{1}{2},j}^x - \mathcal{F}_{i-\frac{1}{2},j}^x}{\Delta x} + \frac{\mathcal{F}_{i,j+\frac{1}{2}}^y - \mathcal{F}_{i,j-\frac{1}{2}}^y}{\Delta y} \right) \\ & + \Delta t \phi_{i,j}^{v(n)} \left(\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} \right)_{i,j} \end{aligned} \quad (3.13)$$

一方, SIA はセミラグランジェ法に基づき, 以下のように更新される.

$$\phi_{i+\frac{1}{2},j}^{x(n+1)} = \phi^{x(n)}(x_{i+\frac{1}{2}} - u\Delta t, y_j - v\Delta t) \quad (3.14)$$

$$\phi_{i,j+\frac{1}{2}}^{y(n+1)} = \phi^{y(n)}(x_i - u\Delta t, y_{j+\frac{1}{2}} - v\Delta t) \quad (3.15)$$

ここで鍵となる解決されるべき問題は, いかに式 (3.9), (3.10), (3.14) および (3.15) を適切に計算するかということである. いま, ある一般化された流速 u' および v' に対し, $\phi^{x(n)}(x_{i+\frac{1}{2}} - u'\Delta t, y_j - v'\Delta t)$ および $\phi^{y(n)}(x_i - u'\Delta t, y_{j+\frac{1}{2}} - v'\Delta t)$ は, 図 3.3 に示すように以下の 2 ステップの計算プロセスにより計算される.

Step 1 x 軸方向について, SIA の中間値 $\phi^{x(*)}(x_{i+\frac{1}{2}} - u'\Delta t, y_j)$ およびフラックス $\mathcal{F}^{x(*)}(x_{i+\frac{1}{2}} - u'\Delta t, y_j)$ を一次元 CIP-CSL により計算する. 同様に, y 軸方向について, $\phi^{y(*)}(x_i, y_{j+\frac{1}{2}} - v'\Delta t)$ および $\mathcal{F}^{y(*)}(x_i, y_{j+\frac{1}{2}} - v'\Delta t)$ を計算する.

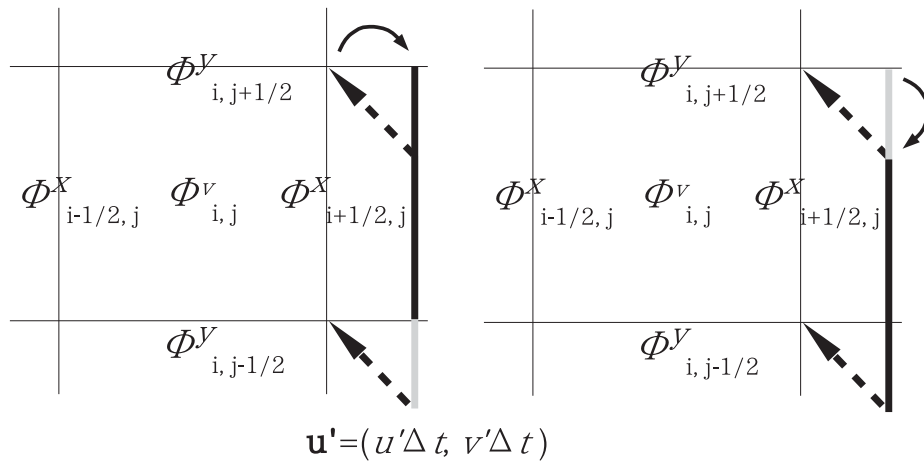


図 3.3 Two step SIA calculation method, STEP1(left) and STEP2(right).

Step 2 Step 1 で得られた中間値より, SIA を以下のように更新する.

$$\begin{aligned}
 \phi^{x(n)}(x_{i+\frac{1}{2}} - u'\Delta t, y_j - v'\Delta t) \\
 &= \phi^{x(*)}(x_{i+\frac{1}{2}} - u'\Delta t, y_j) - \Delta t \left(v' \frac{\partial \phi}{\partial y} \right) \\
 &= \phi^{x(*)}(x_{i+\frac{1}{2}} - u'\Delta t, y_j) - \Delta t \left\{ \frac{\partial v' \phi}{\partial y} - \phi \left(\frac{\partial v'}{\partial y} \right) \right\} \quad (3.16)
 \end{aligned}$$

式 (3.16) の右辺における y 軸方向のフラックスの勾配は, Step 1 の中間フラックスを用いて以下のように線形補間により近似できる.

$$\frac{\partial v' \phi}{\partial y} = \frac{F^+ - F^-}{\Delta y} \quad (3.17)$$

$$\begin{aligned}
 F^+ &\equiv \left(\frac{1}{2} + \xi \right) \mathcal{F}^{y(*)}(x_i, y_{j+\frac{1}{2}} - v'\Delta t) \\
 &\quad + \left(\frac{1}{2} - \xi \right) \mathcal{F}^{y(*)}(x_{i+1}, y_{j+\frac{1}{2}} - v'\Delta t) \quad (3.18)
 \end{aligned}$$

$$\begin{aligned}
 F^- &\equiv \left(\frac{1}{2} + \xi \right) \mathcal{F}^{y(*)}(x_i, y_{j-\frac{1}{2}} - v'\Delta t) \\
 &\quad + \left(\frac{1}{2} - \xi \right) \mathcal{F}^{y(*)}(x_{i+1}, y_{j-\frac{1}{2}} - v'\Delta t) \quad (3.19)
 \end{aligned}$$

ここで $\xi = u'\Delta t / \Delta x$ である. 式 (3.16) の右辺の y 軸方向流速勾配は, 中心差分により近似できる. 同様の方法により, SIA $\phi^{y(n)}(x_i - u'\Delta t, y_{j+\frac{1}{2}} - v'\Delta t)$ を更新することができる.

以上のプロセスを繰り返し用いることにより, フラックスの計算および SIA の更新ができる. フラックス (式 (3.9), (3.10)) は $u' = g_1 u$ および $g_2 u$, $v' = g_1 v$ および $g_2 v$ とすることにより求められる. シンプソン則 (式 (3.11), (3.12)) を用いる場合は, $u' = 1/2 u$ および u , $v' = 1/2 v$ および v とすることにより求められる. また SIA (式 (3.14), (3.15)) は $u' = u$, $v' = v$ とすることにより更新できる.

この計算手法は容易に三次元計算に拡張できる. 本提案手法を既存の VSIAM3 アルゴリズムと区別するため, 我々はこの手法を Unsplit Time Integration VSIAM3 (UTI-VSIAM3) とよぶことにする. 非次元分割手法であるため, 既存の VSIAM3 よりも次元対称性は大幅に改善される. 我々は, UTI-VSIAM3 を気液界面の移流計算および流体ソルバーにおける物理量の輸送方程式の双方に用いることとする.



3.1.2 UTI-VSIAM3 の移流計算ベンチマークテスト

二次元移流計算問題により、提案手法の妥当性を確認する。各セルの移流速度 u および v は、計算領域の中心点のまわりに剛体回転速度を与えるように決定される。メッシュ数は 75×75 とする。初期値分布として、TOTO の”T”の字の形を 0 から 1 の不連続ジャンプ値分布で計算領域の中心に配置した。

CFL=0.3 (1 回転で 775 サイクル) での UTI-VSIAM3 の計算結果を図 3.4 に示す。比較のために、有限体積法の市販ソフトウェアでは事実上の標準として広く用いられている MUSCL 法 [78][79][80] の計算結果を併せて示す。この両者の比較の前提として、非圧縮性流れ計算において同じ格子で計算する場合に、圧力ポアソン方程式の求解が全体計算時間に占める割合は支配的であり、SIA モーメントを追加することによる移流計算量の増加が、全体の計算量増加へおよぼす影響は軽微であることを想定している。また精度の高い既存の移流計算手法として ENO 法 [52][53] や WENO 法 [33][34] などの高次精度計算手法があるが、これらの高次精度計算手法は $i+2$ など隣接格子の外側の格子の値を必要とするため、領域分割並列計算における通信量の増大が見込まれること、また非構造格子への拡張が本質的に容易でないことから、今回は比較対象としない。一方 DG 法など一つの格子で複数の自由度を持つ計算手法の場合、同じ格子数の比較では圧力計算等他の計算量が飛躍的に増大するため、こちらも比較対象から外す。提案の UTI-VSIAM3 法は既存の MUSCL 法と同じ格子数で比較した場合、SIA はほぼ移流計算のみで用いられるため、移流計算以外の計算量の増加は無視できる。UTI-VSIAM3 の非構造格子への拡張は容易であり、詳細は第 6 章で述べる。

図 3.4 の計算結果では、まず 0~1 の界面が MUSCL 法では拡散している箇所があるが、UTI-VSIAM3 ではほぼどの界面も均一にコンパクトな幅を保っている。また T の字形の角の部分が MUSCL 法では丸くなっているようすが顕著であるが、UTI-VSIAM3 の方はより丸みが抑えられていることが分かる。したがって UTI-VSIAM3 において SIA を追加したマルチモーメントの精度向上効果が明らかとなった。

また提案手法はオリジナルの VSIAM3 とほぼ同等の結果であることも分かる。VSIAM3 の場合、次元分割手法であるため、今回の $X \rightarrow Y$ 軸の順で計算した場合と、 $Y \rightarrow X$ の順で計算した場合とでは数値計算結果が異なるという問題があるが、UTI-VSIAM3 では X 軸および Y 軸同時に計算する非次元分割手法であるため、そのような問題が排除できることは一つの利点である。以上より、提案手法は妥当であることが示された。

ただし UTI-VSIAM3 は本質的には他のオイラータイプの有限体積法と同様、界面付近の不連続性を完全には保持することはできず、数値拡散を排除することはできない。したがってこの計算手法を流体率 F の移流計算に用いる場合、気液移動界面の遷移領域のぼやけを抑制するための数値的制御が別途必要となる。次節で効果的な逆数値拡散手法を提案する。

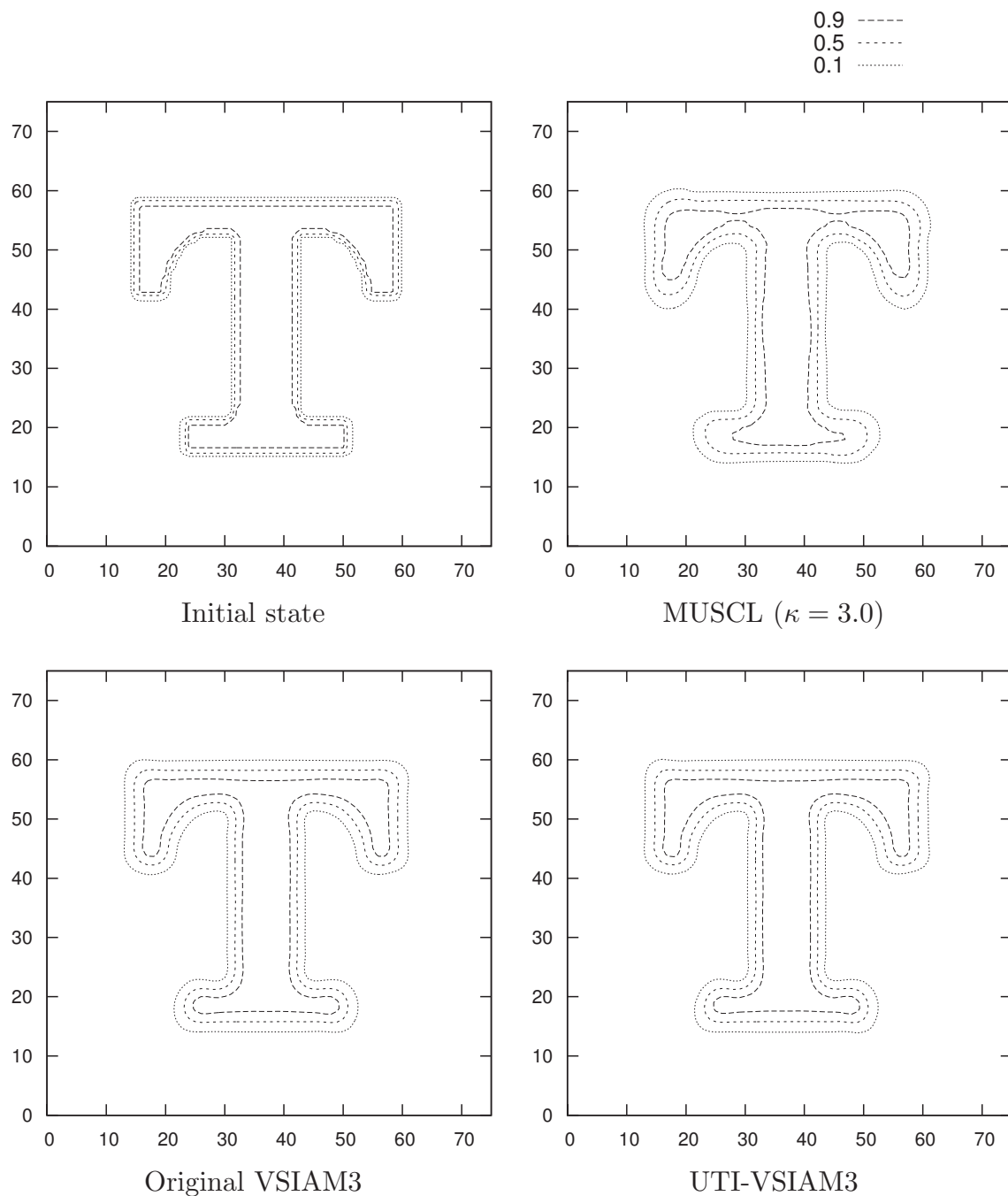


図 3.4 Results of 2D advection test with various schemes (after 1 rotation).

3.2 気液界面捕獲手法 STAA

3.2.1 計算手法の詳細

前節で述べたとおり，UTI-VSIAM3 を用いて移流方程式 $\partial F / \partial t + (\mathbf{u} \cdot \nabla) F = 0$ を解く場合，流体 ($F = 1$) および気体 ($F = 0$) の間の気液界面における数値拡散を低減するための処置が必要となる．

我々が提案する手法は次の 2 ステップにより構成される．ステップ 1 では，レベルセット関数 [15][16][17]，すなわち気液界面からの符号つき距離 G が VOF 関数 F に基づいて生成される． G は液相を正，気相を負で表す． $G = 0$ は気液界面を表し，一般に $F = 0.5$ の位置が $G = 0$ となるようにレベルセット関数は構成される．ステップ 2 では， $|G_{i,j}| > \epsilon$ を満たすセル (i, j) において，VOF 関数が $F = 0$ もしくは $F = 1$ となるように逆数値拡散フラックスを用いて補正される． ϵ は平均メッシュ間隔の数倍の値をとる定数である．逆数値拡散フラックスを通してセルの数値拡散が移動するため， F の保存性は厳密に保証される．

まずステップ 1 のレベルセット関数の計算手法を示す．いま任意の位置 $\mathbf{x}$ において，各タイムステップにおける移流計算の計算結果を $F(\mathbf{x})$ とおく． $F(\mathbf{x})$ の気液界面からの垂直距離を液相において正，気相において負により表した距離関数すなわちレベルセット関数 [16][17] を $G(\mathbf{x})$ とおく．気液界面方向はレベルセット関数 $G(\mathbf{x})$ より

$$\mathbf{n} = -\text{SGN}(G(\mathbf{x})) \frac{\nabla G(\mathbf{x})}{|\nabla G(\mathbf{x})|} \quad (3.20)$$

で求められ， $G(\mathbf{x})$ はレベルセット法の再初期化を行う移流方程式 [17]

$$\frac{\partial G}{\partial t'} - \mathbf{n} \cdot \nabla G = \text{SGN}(G) \quad (3.21)$$

を展開した

$$G^{(n+1)}(\mathbf{x}) = G^{(n)}(\mathbf{x} + \Delta t' \mathbf{n}) + \text{SGN}(G(\mathbf{x})) \Delta t' \quad (3.22)$$

で繰り返し計算により収束解として得られる． $\text{SGN}(a)$ は符号関数であり， a が正のとき 1，負のとき -1 をとる． $\Delta t'$ はメッシュ間隔を超えない適当な正の値である．通常はメッシュ間隔最小値の 0.5 倍程度とする．式 (3.22) の右辺第一項は，気液界面側の隣接メッシュ 4 点を用いて一次線形補間式により簡易に計算できる．すなわち，

$$\begin{aligned} G^{(n)}(\mathbf{x} + \Delta t' \mathbf{n}) = & (1 - t_x)(1 - t_y)G^{(n)}(x_i, y_j) \\ & + t_x(1 - t_y)G^{(n)}(x_{i'}, y_j) \\ & + (1 - t_x)t_yG^{(n)}(x_i, y_{j'}) \\ & + t_x t_y G^{(n)}(x_{i'}, y_{j'}) \end{aligned} \quad (3.23)$$

$$t_x \equiv \frac{\Delta t' |\mathbf{n} \cdot \mathbf{i}|}{\Delta x} \quad (3.24)$$

$$t_y \equiv \frac{\Delta t' |\mathbf{n} \cdot \mathbf{j}|}{\Delta y} \quad (3.25)$$

とする. i', j' は図 3.5 に示すように各々 X 軸方向および Y 軸方向の気液界面側の隣接メッシュを表す. $\mathbf{i}$ および $\mathbf{j}$ は各々 X および Y 軸正方向の単位ベクトルを表す. $G^{(n)}(\mathbf{x})$ の初期値 $G^{(0)}(\mathbf{x})$ は, $F(\mathbf{x}) = 0.5$ において $G^{(0)}(\mathbf{x}) = 0$ となるように,

$$G^{(0)}(\mathbf{x}) \equiv \alpha(F(\mathbf{x}) - 0.5) \quad (3.26)$$

とおく. α は通常はメッシュ間隔最小値の 1.0~3.0 倍とする.

次に Step2 における数値拡散の補正方法を以下に示す. 各セルにおける数値拡散の補正量を $\Delta F_{i,j}$ とおくと, 隣接メッシュの値の変化は以下の式で表される.

$$\delta F_{i',j} = -\Delta F_{i,j} r_x (1 - r_y) \quad (3.27)$$

$$\delta F_{i,j'} = -\Delta F_{i,j} r_y (1 - r_x) \quad (3.28)$$

$$\delta F_{i',j'} = -\Delta F_{i,j} r_x r_y \quad (3.29)$$

$$r_x \equiv \frac{|\mathbf{n} \cdot \mathbf{i}|}{r_{xy} \Delta x} \quad (3.30)$$

$$r_y \equiv \frac{|\mathbf{n} \cdot \mathbf{j}|}{r_{xy} \Delta y} \quad (3.31)$$

$$r_{xy} \equiv \max \left(\frac{|\mathbf{n} \cdot \mathbf{i}|}{\Delta x}, \frac{|\mathbf{n} \cdot \mathbf{j}|}{\Delta y} \right) \quad (3.32)$$

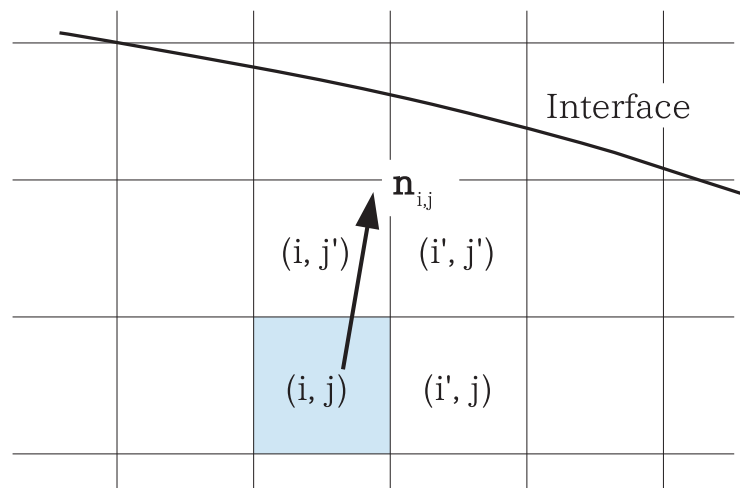


図 3.5 The definition of neighboring cells.



補正量 $\Delta F_{i,j}$ は、液相のメッシュでは $1 - F_{i,j}$ 、気相のメッシュでは $0 - F_{i,j}$ である。ただし気液界面がずれないようにするため、 $|G_{i,j}| \leq \epsilon$ を満たすセルは $\Delta F_{i,j} = 0$ とする。 ϵ は平均メッシュ間隔の 0.5~2.0 倍の値をとる。

このアルゴリズムにより、気液界面における流体率 F の数値拡散を簡易に修正することができる。本手法は三次元へも容易に拡張できる。

以上の一連の数値拡散補正手法を Surface Tracking by Artificial Anti-diffusion (STAA) 法とよぶ。STAA 法では複雑な条件分岐式は必要なく、高性能計算への適用は極めて容易である。

3.2.2 STAA 法の移流計算ベンチマークテスト

3.1.2 節の移流方程式ベンチマークテストの UTI-VSIAM3 と STAA 法を組み合わせた結果を図 3.6 に示す。図 3.4 の UTI-VSIAM3 のみの結果では、 $F = 0$ と $F = 1$ との間の気液界面のぼやけが生じていたが、STAA 法の補正により、シャープに保たれるようになった。一方”T”の字の形は、STAA 法を組み合わせても大きく変形することなく、元の形状を精度良く再現できていることが分かる。

STAA 法は、Lax-Wendroff 法 [81] や MUSCL 法 [78][79][80] など様々な移流計算手法と組み合わせることが可能である。図 3.7 に Rudman[82] の移流計算ベンチマークの実施結果を示す。切り欠きの入った円の内側に位置するメッシュを $F = 1.0$ 、外側を $F = 0.0$ と設定し、剛体回転流速場により時計回りに 1 回転させたものである。200 × 200 メッシュを用いた。三本の線は $F = 0.05, 0.5, 0.95$ を示す。いずれの手法も、STAA 法の数値拡散補正効果により、

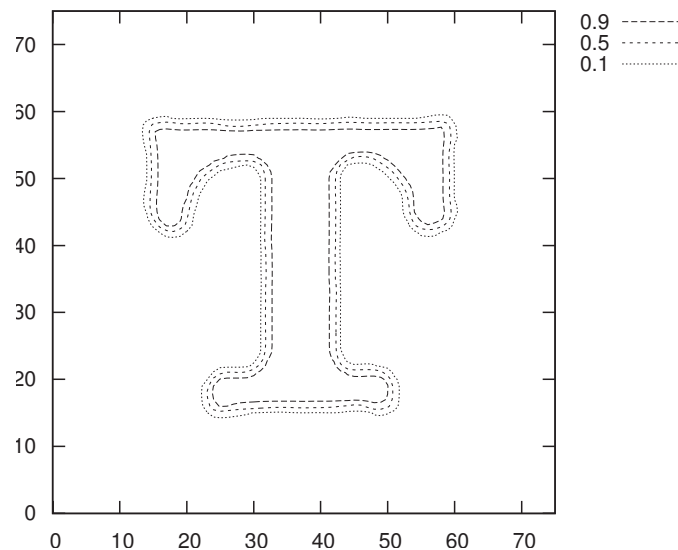


図 3.6 Result of 2D advection test with UTI-VSIAM3 + STAA (after 1 rotation)

界面はシャープに保持されている。しかし計算結果の精度は UTI-VSIAM3 との組み合わせが最も良いことが分かる。また図 3.8 に THINC/SW および THINC/WLIC の計算結果を示

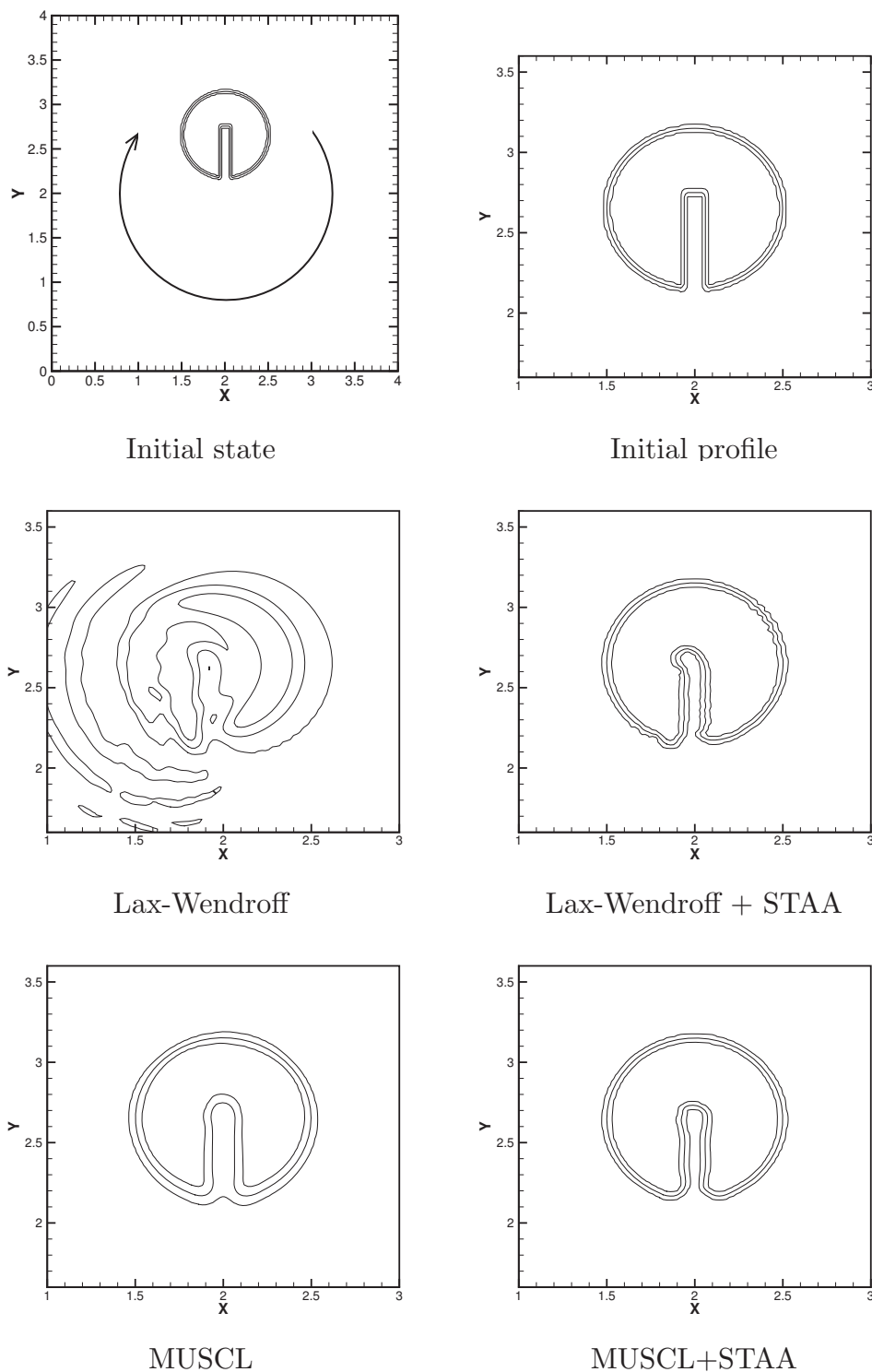


図 3.7 Rudman's problem of notched circle rigid rotation advection (1/2).

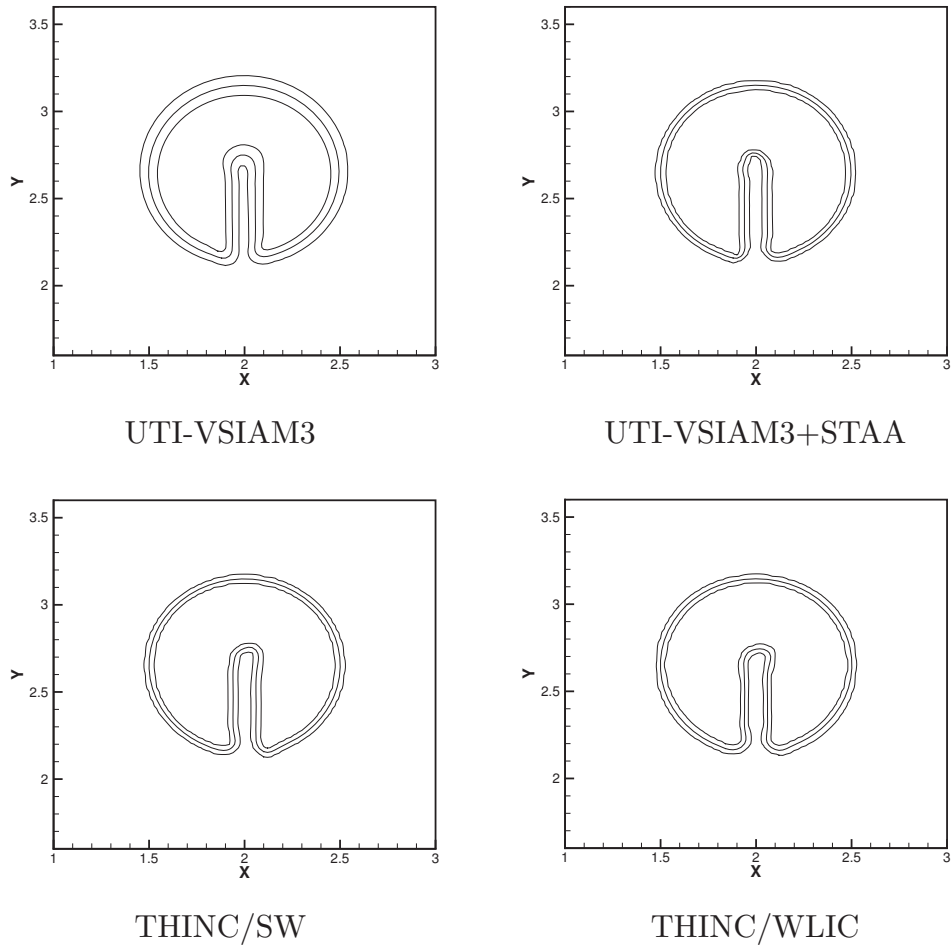


図 3.8 Rudman's problem of notched circle rigid rotation advection (2/2).

表 3.1 Comparison of L1 error with other advection calculation methods.

Calculation method	L1 error
Lax-Wendroff + STAA	2.54×10^{-2}
MUSCL + STAA	1.96×10^{-2}
UTI-VSIAM3 + STAA	1.12×10^{-2}
THINC/SW	1.66×10^{-2}
THINC/WLIC	1.71×10^{-2}

す。いずれの計算手法よりも，UTI-VSIAM3+STAA の方が，気液界面の変形が小さいことが分かる。

表 3.1 に，誤差評価結果を示す。気液界面形状の精度を比較するため，レベルセット関数 $G_{i,j}$ を近似重み関数 $H_{i,j}$ [17] に変換し，初期値 $H_{i,j}^{init}$ との差を計算領域全体で積算した L1

誤差

$$Err = \frac{\sum_{i,j} |H_{i,j} - H_{i,j}^{init}|}{\sum_{i,j} |H_{i,j}^{init}|} \quad (3.33)$$

$$H_{i,j} = \begin{cases} \frac{1}{2} + \frac{G_{i,j}}{2\epsilon} + \frac{1}{2\pi} \sin\left(\pi \frac{G_{i,j}}{\epsilon}\right) & |G_{i,j}| \leq \epsilon \\ 1 & G_{i,j} > \epsilon \\ 0 & G_{i,j} < -\epsilon \end{cases} \quad (3.34)$$

により評価した．なお ϵ は，メッシュ間隔の 0.75 倍とする．表 3.1 から UTI-VSIAM3+STAA が最も計算精度が高いことが分かる．



3.3 表面張力の定式化

表面張力は、STAA 法で生成するレベルセット関数 G を用いて、以下のように定式化される。

$$\mathbf{F}_{V,\text{surf}} \equiv \sigma \kappa \delta(G) \mathbf{g} \quad (3.35)$$

$$\mathbf{g} \equiv \frac{\nabla G}{|\nabla G|} \quad (3.36)$$

σ は表面張力係数、 κ は曲率である。また $\delta(G)$ はデルタ関数であり、 $G \neq 0$ で 0、 $G = 0$ で無限大の値をもつ。実際には緩和されたデルタ関数 $\delta_\epsilon(G)$ を用いる。

$$\delta_\epsilon(G) = \begin{cases} \frac{1}{2\epsilon} \left(1 + \cos \frac{\pi G}{\epsilon}\right) & |G| < \epsilon \\ 0 & |G| \geq \epsilon \end{cases} \quad (3.37)$$

ϵ は $1.0 \sim 3.0 \Delta x$ 程度の値としている。 $\mathbf{g}$ は界面法線ベクトルを表す。したがって式 (3.35) は、界面法線ベクトル方向に、曲率に比例した張力を界面幅 $2.0 \sim 6.0 \Delta x$ にわたってスムーズに与えているという意味になる。このような表面張力係数モデルを Continuum Surface Force (CSF) モデル [83] とよぶ。いま界面近傍でスムーズな、もしくはスムーズ化された VOF 関数 F が与えられている場合、デルタ関数の代わりに、

$$\delta(G) \mathbf{g} \simeq \nabla F \quad (3.38)$$

で与えることもできる。

さらに界面曲率 κ は

$$\kappa = -\nabla \cdot \mathbf{g} \quad (3.39)$$

により求められる。

壁面での固気液三相界面における表面張力、すなわち濡れ性は、接触角モデルを使用して計算する。

3.4 フラクショナルステップ法による気液二相流非圧縮性流体解析手法

3.4.1 手法の詳細

二次元の Navier-Stokes 方程式

$$\frac{\partial \rho u}{\partial t} + \frac{\partial \rho u u}{\partial x} + \frac{\partial \rho u v}{\partial y} = -\frac{\partial P}{\partial x} + \left(\frac{\partial}{\partial x} \mu \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \mu \frac{\partial}{\partial y} \right) u + F_{V,x} \quad (3.40)$$

$$\frac{\partial \rho v}{\partial t} + \frac{\partial \rho v u}{\partial x} + \frac{\partial \rho v v}{\partial y} = -\frac{\partial P}{\partial y} + \left(\frac{\partial}{\partial x} \mu \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \mu \frac{\partial}{\partial y} \right) v + F_{V,y} \quad (3.41)$$

を非保存形に変形すると次の式が得られる.

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \frac{1}{\rho} \left(\frac{\partial}{\partial x} \mu \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \mu \frac{\partial}{\partial y} \right) u + \frac{1}{\rho} F_{V,x} \quad (3.42)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial y} + \frac{1}{\rho} \left(\frac{\partial}{\partial x} \mu \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \mu \frac{\partial}{\partial y} \right) v + \frac{1}{\rho} F_{V,y} \quad (3.43)$$

この非保存形 Navier-Stokes 方程式の移流項は流速 u , v に対する式になっており, 元の ρu , ρv に対する式とは異なり直接流速を計算できるため, 既存の多相流シミュレーションではほとんどこの非保存形の式が用いられる. フラクショナルステップ法 [85][86][87] では, 式 (3.42), (3.43) を各項に分解して解く.

$$\text{移流項: } \frac{\partial u}{\partial t} = -u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y} \quad (3.44)$$

$$\frac{\partial v}{\partial t} = -u \frac{\partial v}{\partial x} - v \frac{\partial v}{\partial y} \quad (3.45)$$

$$\text{非移流項: } \frac{\partial u}{\partial t} = \frac{1}{\rho} \left(\frac{\partial}{\partial x} \mu \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \mu \frac{\partial}{\partial y} \right) u + \frac{1}{\rho} F_{V,x} \quad (3.46)$$

$$\frac{\partial v}{\partial t} = \frac{1}{\rho} \left(\frac{\partial}{\partial x} \mu \frac{\partial}{\partial x} + \frac{\partial}{\partial y} \mu \frac{\partial}{\partial y} \right) v + \frac{1}{\rho} F_{V,y} \quad (3.47)$$

$$\text{圧力項: } \frac{\partial u}{\partial t} = -\frac{1}{\rho} \frac{\partial P}{\partial x} \quad (3.48)$$

$$\frac{\partial v}{\partial t} = -\frac{1}{\rho} \frac{\partial P}{\partial y} \quad (3.49)$$

本計算手法では, 全ての物理量の VIA は格子中心で定義されるコロケート配置の有限体積法を用いる. 移流項 (式 (3.44), (3.45)) は, 3.1.1 節の UTI-VSIAM3 により計算する. また流体率の移流方程式 (式 (1.2)) を解いて各格子の流体率を更新し, 非移流項の計算で用いる密度および粘性係数を計算する. 以下の式により物性値が得られる. なお "liq" は液体, また "gas"



は気体の物性値を表す．

$$\rho = (1 - F)\rho_{gas} + F\rho_{liq} \quad (3.50)$$

$$\mu = (1 - F)\mu_{gas} + F\mu_{liq} \quad (3.51)$$

次に非移流項 (式 (3.46), (3.47)) を計算する．粘性項の空間微分は二次精度中心差分を用い，時間項は一次前進差分を用いて u , v を更新する．

最後に圧力項の計算では，式 (3.48) および (3.49) の両辺に発散をとり，非圧縮性流れの条件 $\nabla \cdot \mathbf{u}^{n+1} = 0$ により，以下の圧力に関するポアソン方程式が得られる．

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla P^{(n+1)} \right) = \nabla \cdot \mathbf{u}^{(*)} \quad (3.52)$$

流速の発散項 $\nabla \cdot \mathbf{u}^{(*)}$ は，以下の式により計算される．

$$\left(\nabla \cdot \mathbf{u}^{(*)} \right)_{i,j} = \frac{u_{i+1/2,j}^{(*)} - u_{i-1/2,j}^{(*)}}{\Delta x} + \frac{v_{i,j+1/2}^{(*)} - v_{i,j-1/2}^{(*)}}{\Delta y} \quad (3.53)$$

ここでセル界面での流速 $u_{i-1/2,j}^{(*)}$ および $v_{i,j-1/2}^{(*)}$ はマルチモーメントの特徴を生かし，SIA の値を用いる．

$$u_{i-1/2,j}^{(*)} \equiv u_{i-1/2,j}^{x(*)} \quad (3.54)$$

$$v_{i,j-1/2}^{(*)} \equiv v_{i,j-1/2}^{y(*)} \quad (3.55)$$

この SIA の中間値 $u_{i-1/2,j}^{x(*)}$ および $v_{i,j-1/2}^{y(*)}$ は，Time Evolution Converting (TEC) [77] の定式化に沿い，以下の VIA の時間変化補間により計算する．

$$u_{i-1/2,j}^{x(*)} = u_{i-1/2,j}^{x(n)} + \frac{1}{2} \left((u_{i-1,j}^{v(*)} - u_{i-1,j}^{v(n)}) + (u_{i,j}^{v(*)} - u_{i,j}^{v(n)}) \right) \quad (3.56)$$

$$v_{i,j-1/2}^{y(*)} = v_{i,j-1/2}^{y(n)} + \frac{1}{2} \left((v_{i,j-1}^{v(*)} - v_{i,j-1}^{v(n)}) + (v_{i,j}^{v(*)} - v_{i,j}^{v(n)}) \right) \quad (3.57)$$

また圧力のラプラシアン項は，以下のように離散化する．

$$\begin{aligned} \left(\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla P^{(n+1)} \right) \right)_{i,j} &= a_w(P_{i-1,j}^{(n+1)} - P_{i,j}^{(n+1)}) \\ &\quad + a_e(P_{i+1,j}^{(n+1)} - P_{i,j}^{(n+1)}) \\ &\quad + a_s(P_{i,j-1}^{(n+1)} - P_{i,j}^{(n+1)}) \\ &\quad + a_n(P_{i,j+1}^{(n+1)} - P_{i,j}^{(n+1)}) \end{aligned} \quad (3.58)$$

$$a_w \equiv \left(\frac{2}{\rho_{i-1,j} + \rho_{i,j}} \right) \frac{\Delta t}{\Delta x^2} \quad (3.59)$$

$$a_e \equiv \left(\frac{2}{\rho_{i+1,j} + \rho_{i,j}} \right) \frac{\Delta t}{\Delta x^2} \quad (3.60)$$

$$a_w \equiv \left(\frac{2}{\rho_{i,j-1} + \rho_{i,j}} \right) \frac{\Delta t}{\Delta y^2} \quad (3.61)$$

$$a_e \equiv \left(\frac{2}{\rho_{i,j+1} + \rho_{i,j}} \right) \frac{\Delta t}{\Delta y^2} \quad (3.62)$$

圧力 $P^{(n+1)}$ に関するポアソン方程式を解いた後は、式 (3.48), (3.49) により流速を修正する。式 (3.48), (3.49) の圧力勾配は、 $P^{(n+1)}$ の中心差分により計算する。

なお Navier-Stokes 方程式の解法としては、市販ソフトウェア等では Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) 系の解法が広く用いられている。これらの解法では、圧力を予測値として流速について Navier-Stokes 方程式を陰解法で解き、圧力はポアソン方程式により修正する。この過程を繰り返し計算することにより、1 タイムステップにおける収束解に到達する。これらの解法は繰り返し計算におけるアルゴリズムの違いにより SIMPLE[88][89], SIMPLER[90][89], SIMPLEC[91], PISO[92] 等がある。フラクショナルステップとは異なり、1 タイムステップにおいて反復計算が必要なため、通常は計算量がフラクショナルステップよりも多くなる。SIMPLE 系はフラクショナルステップよりも定常解に収束するのが早いため定常流れの計算に好適であり、非定常流れの気液二相流ではフラクショナルステップが優れると考えられる。

3.4.2 二次元ダムブレイク問題による計算精度検証

二次元ダム崩壊流れのテストシミュレーションの結果を図 3.9 に示す。メッシュ数は 100×100 であり、 $\Delta x = \Delta y = 3.0\text{mm}$ とした。重力加速度を Y 軸方向下方に 9.8m/s^2 と設定した。時刻 $t = 0.0$ において、流体率 F は $0.0 \leq x \leq 150.0\text{mm}$ かつ $0.0 \leq y \leq 150.0\text{mm}$ の計算領域において 1 とし、その領域を水と設定した。それ以外の領域は $F = 0$ とし、空気領域とした。密度および粘性係数は F に応じて水もしくは空気の物性値を設定した。

シミュレーション結果からは、水と空気の界面幅はコンパクトに保持されていることが分かる。また微小飛沫や気泡の形状や動きもよく再現されている。さらにこのシミュレーションは終始極めて安定していたことも強調しておきたい。以上の結果より、本章で提案した多相流計算手法の妥当性が示されたといえる。



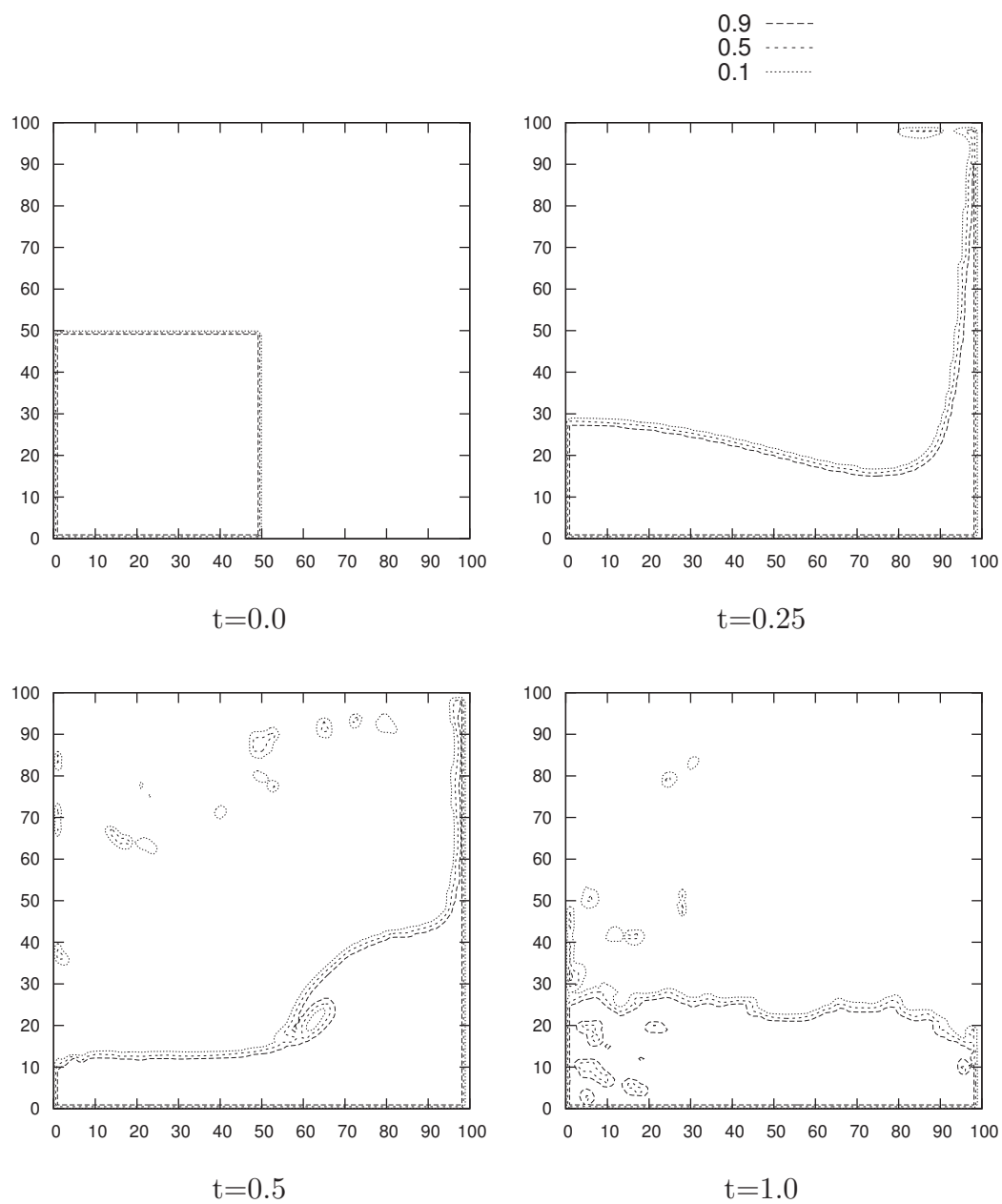


図 3.9 Result of dam break test

3.4.3 三次元気泡上昇問題による検証

三次元の精度検証として, Hnat-Buckmaster[93] の気泡上昇問題を実施した. 空気と水の密度は $\rho_{gas} = 1.0 \text{ kg/m}^3$ および $\rho_{liq} = 875.5 \text{ kg/m}^3$ とし, 粘性係数は各々 $\mu_{gas} = 0.001 \text{ Pa}\cdot\text{s}$ および $\mu_{liq} = 0.118 \text{ Pa}\cdot\text{s}$ とした. 表面張力係数は $\sigma = 3.22 \times 10^{-2} \text{ N/m}$ とした. 計算格子は $128 \times 128 \times 256$ 均一格子を用いた. 壁面の影響を除外するために, 壁境界に接しているセルにおいて far-field boundary 条件 [94] を設定した.

図 3.10 に示すように, Bubble-A の実験結果の気泡変形をよく再現している. さらに図 3.11 に示すように, 最終的な気泡上昇速度は 0.214 m/s となり, この値は実験での測定値 0.215 m/s に極めてよく一致していることが分かった.

3.4.4 住宅設備機器におけるシミュレーション例

図 3.12 および図 3.13 に, 衛生陶器の流れおよびキッチンの流れにおけるシミュレーション実施例を示す. 衛生陶器のシミュレーションでは, 水の流れに沿って複数剛体が上流から下流へ流れており, 妥当な結果といえる. またキッチンではバケツに見立てた剛体を回転させることで, 水がキッチンのシンクに流れ出るようすや, シンク表面およびトラップ内の流れも自然な結果が再現されている. したがって UTI-VSIAM3 および STAA を用いたフラクショナルステップ法による構造格子固気液三相流計算手法は妥当であるといえる.



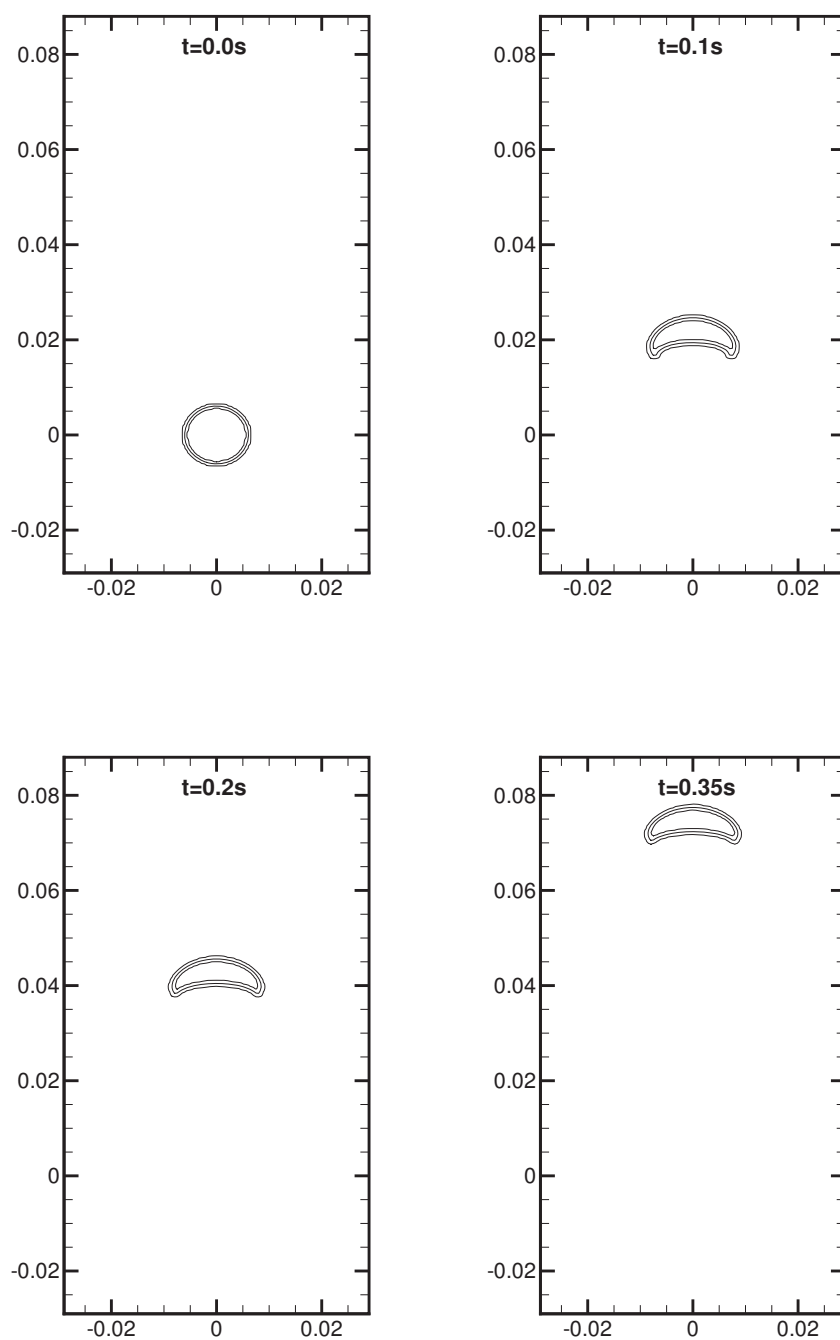


図 3.10 The VOF function of the rising bubble. Displayed are the contours of 0.05, 0.5 and 0.95.

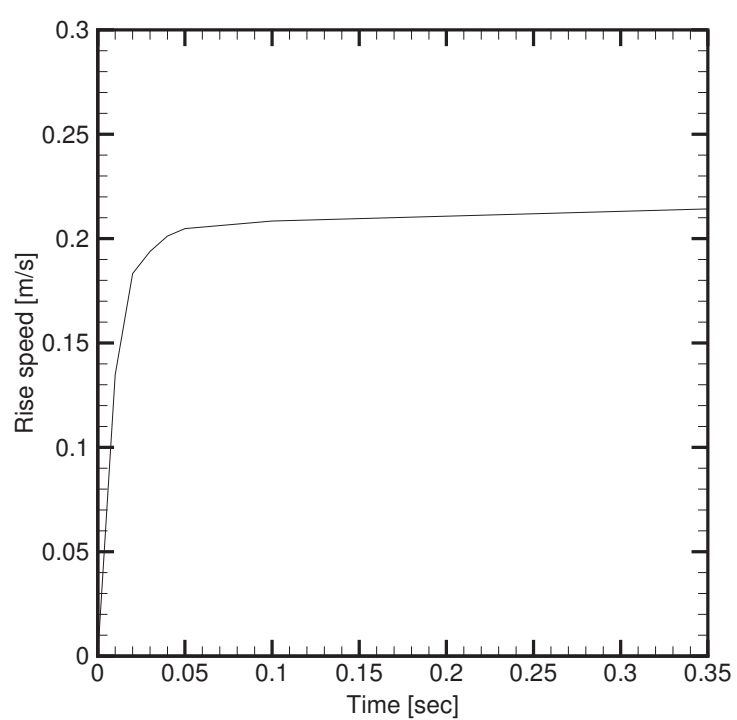


図 3.11 The velocity profile at the top of the bubble.



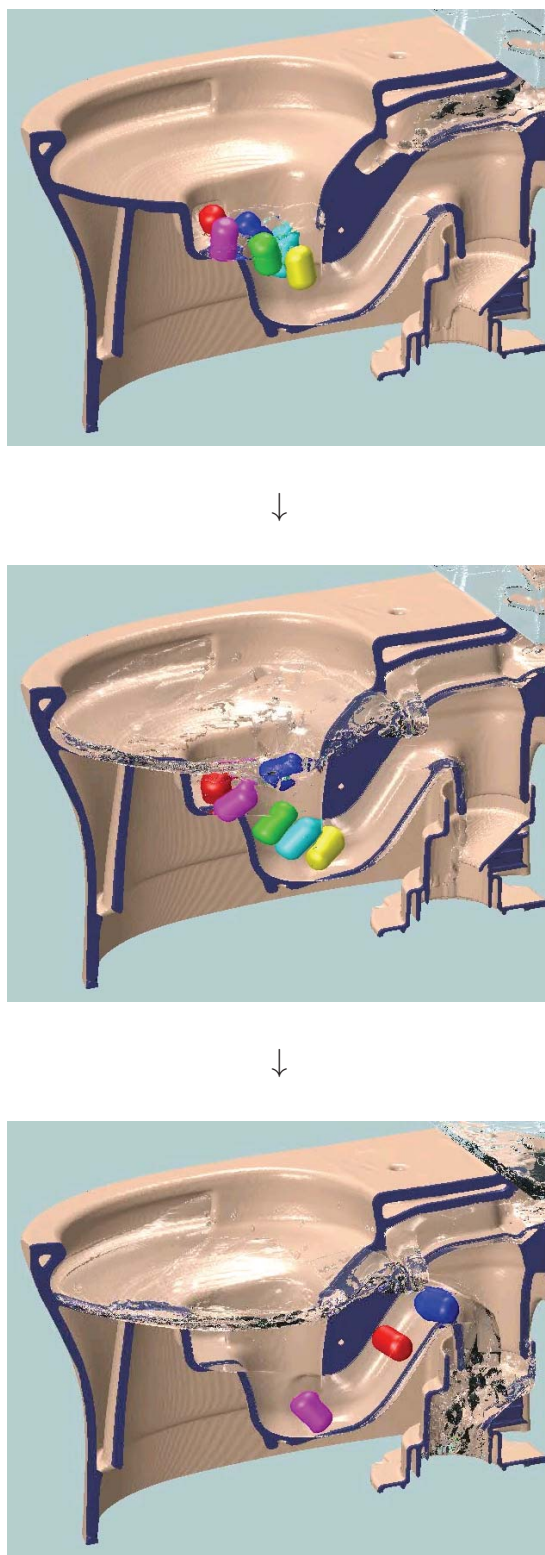


図 3.12 Result of simulation of a sanitary ware (mesh number = $181 \times 135 \times 267$, CFL=0.48)

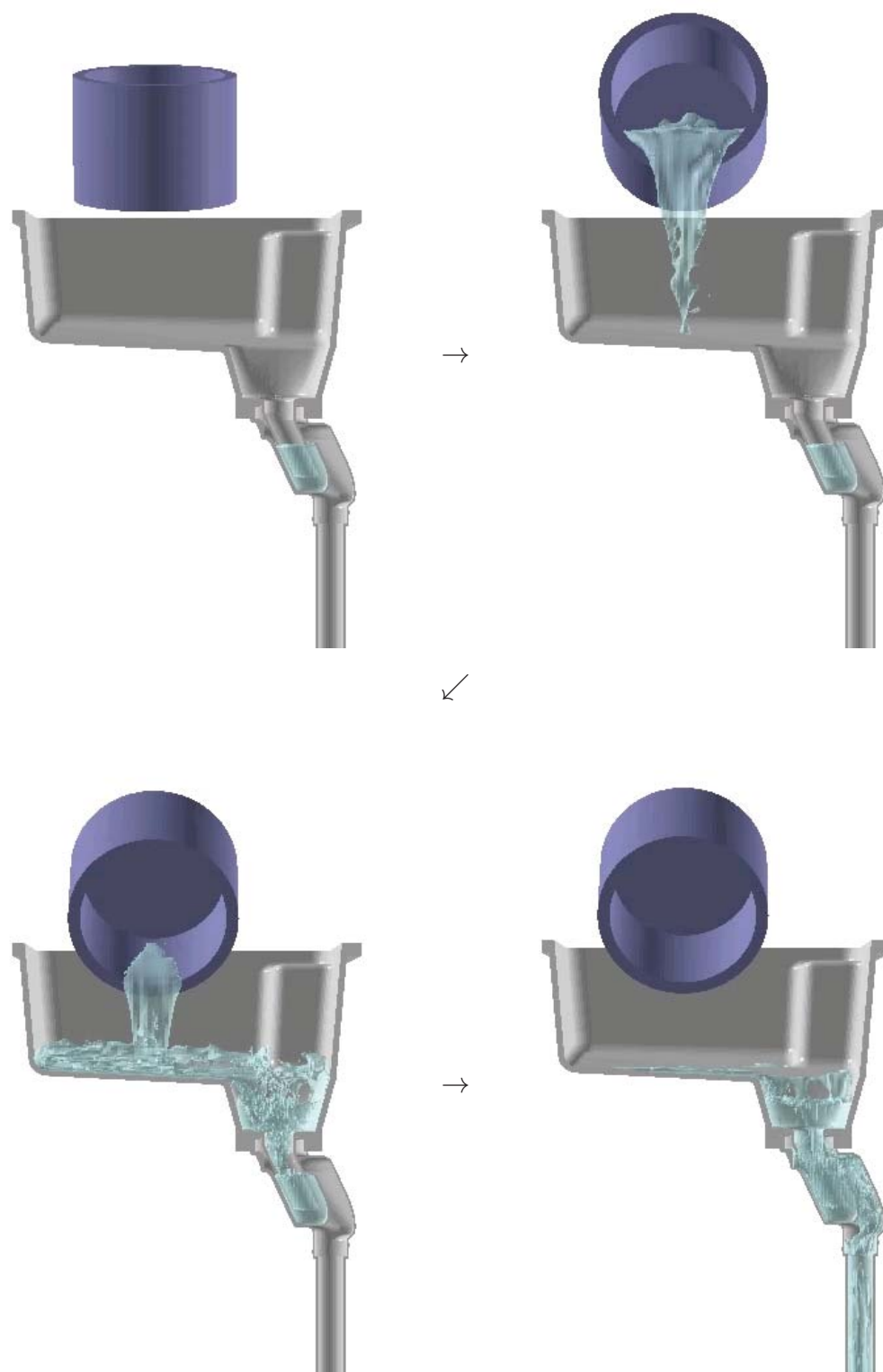


図 3.13 Result of simulation of a kitchen (mesh number = $181 \times 223 \times 160$, CFL=0.48)

3.5 第 3 章のまとめ

直交格子における多相流シミュレーション手法を提案した。UTI-VSIAM3 とよぶマルチモーメント移流計算手法により、既存手法と比較して高精度な移流計算結果が得られることを示した。また移流計算において数値拡散を補正する STAA 法を提案し、UTI-VSIAM3 との組み合わせで高精度な気液界面捕獲スキームを構築できることをベンチマークテストを通じて示した。さらにフラクショナルステップ法をベースとした有限体積法にこれらの移流計算手法を組み込み、衛生陶器やキッチンの水の流れのシミュレーションに適用可能なフレームワークを提案した。

第 4 章

大規模計算対応のための GPGPU 化および並列化

4.1 多相流計算における計算効率化の方策

4.1.1 直交格子有限体積法におけるデータ構造の効率化

図 4.1 に衛生陶器の流れのシミュレーション例を示す．衛生陶器では画面中央部のボウル部と右側のトラップ部とで流路が構成されており，その流路に非常に多くの微小気泡，飛沫および薄膜流れが発生する．したがって流路のある局所的部分の計算精度に着目するのではなく，



図 4.1 A example of multiphase simulation of a sanitary ware.

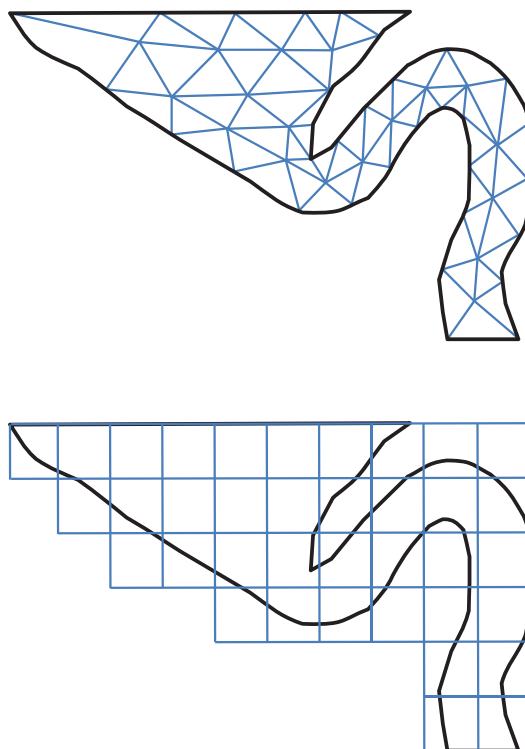


図 4.2 The difference between tetrahedral (upper) and Cartesian (lower) grids.

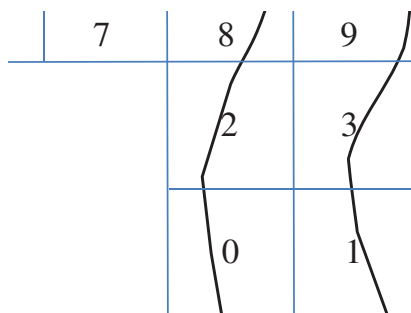


図 4.3 Mesh numbers of unstructured grids.

流路全体にわたって精度の高い計算が必要である。一方流路の外側の大気空間では空気の流れは緩やかであり、流路の水の流れにおよぼす影響は無視できる。したがって計算領域としては、ボウル部とトラップ部の流路に計算対象領域を限定して計算格子を配置することで、衛生陶器を囲む矩形領域全体を計算するよりもおよそ 1/10 程度の計算格子数で済む。

このボウルとトラップの複雑流路領域を解析する場合、一般的にはテトラ格子などの非構造格子が用いられる (図 4.2 上)。テトラ格子を用いる場合は、任意の流路形状に対して自動格子分割が可能であり、多くの流体解析ソフトウェアで採用されている。しかしテトラ格子などの非構造格子は一般的に直交格子よりも計算精度が劣り、また計算量が多いという問題がある。

本手法では、図 4.2 下のように、流路が含まれる直交格子のみを残し、他の格子は計算対象外として削除する方法を採用した。この場合、残された各格子は昇順に番号を付され、各格子番号に隣接する格子番号をインデックス配列として保持する。図 4.3 の例では、メッシュ番号 2 の格子は、右、下、上で各々メッシュ番号 3, 0, 8 の格子と隣接している。このメッシュ接続情報は、

$$\text{im}[i], \text{ip}[i], \text{jm}[i], \text{jp}[i], i = 0, \dots, N-1$$

(N はメッシュ数)

というインデックス配列 im , ip , jm , jp により保持され、

$$\text{im}[2] = 2, \text{ip}[2] = 3, \text{jm}[2] = 0, \text{jp}[2] = 8$$

と定義される。隣接領域が存在しない場合は自分自身のメッシュ番号とする。三次元の場合はインデックス配列として $\text{km}[i]$, $\text{kp}[i]$ が追加される。このように隣接する格子のアドレスを常にインデックス配列を介して間接的に参照するために余分なメモリアクセスが必要となるが、衛生陶器のように流路領域が全領域の 1/10 程度しかない場合は、全矩形領域を保持する通常の直交格子計算手法と比較して計算量、記憶容量を大幅に低減できる。

4.1.2 CPU コードの GPU 化による加速化

多相流体シミュレーションコードは、データの初期設定、GPU メモリへの転送、GPU カーネルの発行および計算結果出力にのみ C 言語で記述され、流体計算自体は全て NVidia CUDA C による GPU カーネルで実装された。UTI-VSIAM3, STAA 法, FAVOR 法およびフラクショナルステップ法の定式化は簡素であるため、GPU カーネルの実装は容易である。東工大スパコン「TSUBAME2」[97] をはじめとする GPU 並列計算サーバーで広く用いられている NVidia CUDA デバイスは CPU プロセッサと比較して高いメモリバンド幅を持っているため、流体計算の大部分を占める配列のメモリアクセス速度が速く、全体の計算時間短縮が期待できる。

4.1.3 境界不整合領域分割法を用いた計算並列化

大規模並列計算の実装においては、図 4.2 下の直交格子計算領域を均等に領域分割して各プロセッサに分配するとともに、計算中のメッセージパッシングデータ量を最小化して、実用的なスケーラビリティを確保することが大きな課題である。既存の一般的な手法であるブロック分割法 [98][99] を適用し、例として図 4.4 のように全体矩形領域のメッシュ分割数 $N = N_x \times N_y \times N_z$ に対して Z 軸方向に M 個の領域に等間隔分割し、各々 M 個のプロセッサに割り当てて並列計算を実行することを考える。流路領域の形状が複雑な場合、各分割領域



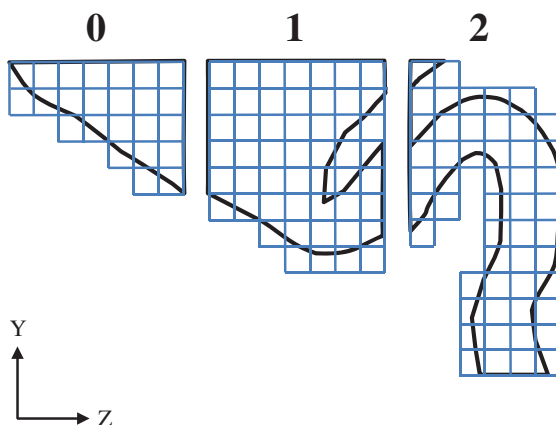


図 4.4 Block decomposition method.

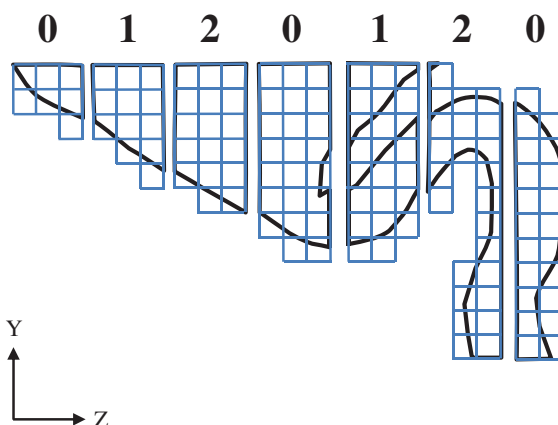


図 4.5 Block cyclic decomposition method.

の流路メッシュ数は N/M に対して著しい不均等が生じ，並列化効率は大幅に低下する．不均等を緩和するため図 4.5 のようにブロック・サイクリック法 [98][99] を適用する方法は有力であるが，各プロセスにおける流路メッシュ数を N/M に近づけるためにブロックのサイズを小さくしていくと，1 ブロックの計算におけるベクトル長が小さくなり GPU 計算での効率が低下する可能性や，分散メモリ並列計算においてプロセス間通信量が増大する可能性が考えられる．

そこでブロック分割法において，各分割領域の流路メッシュ数をなるべく N/M に近づけるように， Z 軸方向の分割位置を調整する方法が有力と考えられる．しかし M が数十以上の比較的大きな並列計算の場合，各分割領域は Z 軸方向に対して極めて薄い形状となり，計算量に対する通信量の割合が増大する．そこで Y 軸方向および X 軸方向に対しても順次領域分割を行うアルゴリズムを提案する． Z ， Y および X 軸方向の領域分割数を M_z ， M_y および M_x とおくと，図 4.6 に示すように，

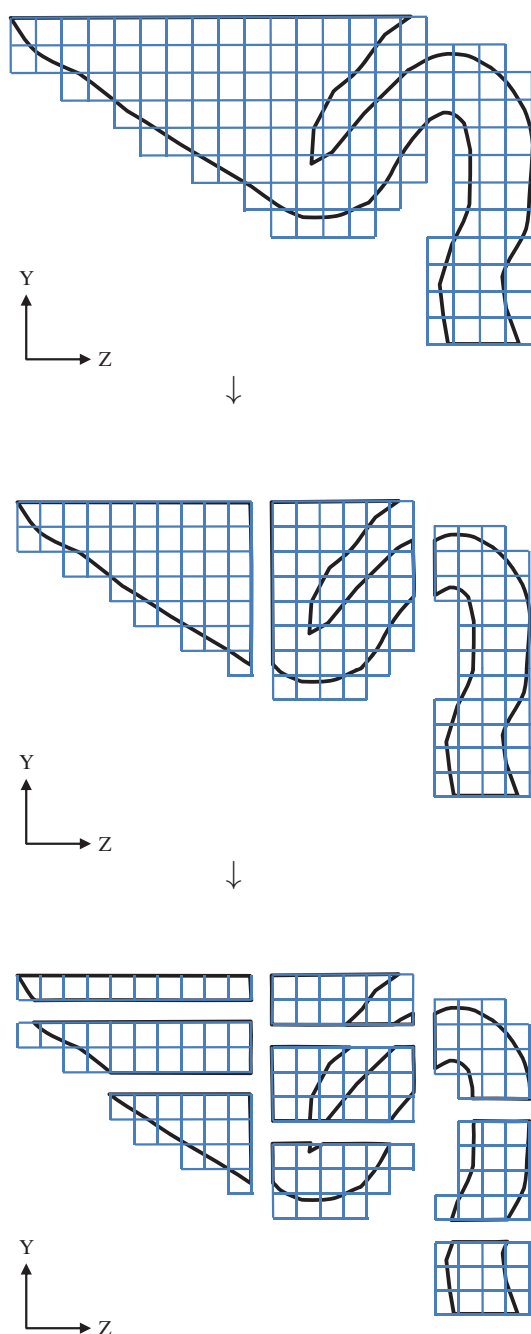


図 4.6 Non-conformal boundary domain decomposition method.

1. 各分割領域の流路メッシュ数が最も均等になるように Z 軸方向に領域分割する．すなわち， Z 軸方向の m 番目の領域の最後尾のメッシュ番号 $k_{max,m}$ は以下のように決定される．

```

Ns = 0;
m = 0;
for (k = 1; k <= Nz, k++) {
    for (j = 1; j <= Ny, j++) {
        for (i = 1; i <= Nx, i++) {
            if (メッシュ (i,j,k) が計算領域ならば) Ns++;
        }
    }
    if (Ns >= N*(m+1)/Mz) {
        kmax,m = k;
        m++;
    }
}

```

上記の， $Ns \geq N*(m+1)/Mz$ の判定により，各分割領域の計算領域メッシュ数は N/M_z に近い値となる．

2. 分割された領域ごとに，流路のメッシュ数が最も均等になるように (1) と同様の方法で Y 軸方向に領域分割する．したがって Y 軸方向の分割位置は Z 軸方向分割領域に対して不整合となる．
3. 分割された領域ごとに，流路のメッシュ数が最も均等になるように (1) と同様の方法で X 軸方向に領域分割する．したがって X 軸方向の分割位置は Y ， Z 軸方向分割領域に対して不整合となる．

の 3 ステップで領域分割することにより，最終的な分割領域の流路メッシュ数はほぼ均等となる．この領域分割法を境界不整合領域分割法 (Non-Conformal boundary Domain Decomposition Method, NCDDM) とよぶことにする．このアルゴリズムにより，任意の複雑流路形状においてメッシュ数がほぼ均一の領域分割を実現できる．ブロック分割法よりもブロックサイズの均一性が高く，またブロック・サイクリック分割法よりもブロックサイズを大きくできる．

また Z ， Y および X 軸方向の領域分割数 M_z ， M_y および M_x は解析実施者が自由に設定できるようにし，解析実施者が事前のテスト計算などにより，計算時間が最短となるように M_z ， M_y および M_x を決定する．

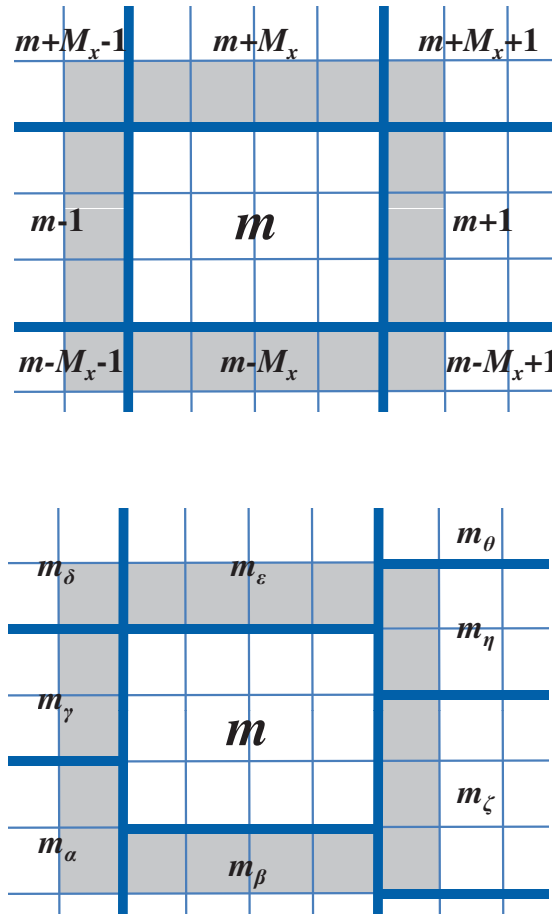


図 4.7 Difference from general domain decomposition method. (upper : general DDM, lower : presented method)

STAA 法の計算において、格子 (i, j, k) は隣接格子

$$\begin{aligned}
 (i_{nb}, j_{nb}, k_{nb}), \quad & i_{nb} = i - 1, i, i + 1, \\
 & j_{nb} = j - 1, j, j + 1, \\
 & k_{nb} = k - 1, k, k + 1, \\
 & (i_{nb}, j_{nb}, k_{nb}) \neq (i, j, k)
 \end{aligned}$$

の計 26 点の情報を必要とする。したがって分割領域の端部の格子では、計算が開始される前に他の分割領域のデータをメッセージパッシングにより受け取っておく必要がある。この部分の領域は一般に袖領域とよばれる。

通常の矩形領域直交格子の多次元領域分割法では、図 4.7 上のように分割領域は境界整合しているため、袖領域（灰色部）の通信相手の領域番号およびデータ通信量は一意に決定される。いま二次元の場合に $M = M_x \times M_y$ に領域分割すると仮定すると、領域番号 m は領域番号



```

for (i = 0; i < M, i++) {
    for (j = 0; j < M, j++) {
        if (i == j) continue;
        if (i == my_Rank && nsbuf[j] > 0) {
            MPI_Isend(&sbuf[nsobuf[j]], nsbuf[j],
                     MPI_DOUBLE, j, ...);
        }
        if (j == my_Rank && nrbuf[i] > 0) {
            MPI_Irecv(&rbuf[nrobuf[i]], nrbuf[i],
                     MPI_DOUBLE, i, ...);
        }
    }
}
MPI_Waitall(...);

```

図 4.8 Pseudo-code for message passing of the sleeve meshes.

$m-1$, $m+1$, $m-M_x-1$, $m-M_x$, $m-M_x+1$, $m+M_x-1$, $m+M_x$, $m+M_x+1$ の計 8 領域と通信すればよい。

一方，提案の境界不整合領域分割法では図 4.7 下のように境界が揃っていないため袖領域の通信の手順は極めて複雑となる．そこで MPI の非ブロッキング通信機能を用いて簡易に全ての袖領域通信を行う実装方法を提案する．図 4.8 の擬似コードに示すように，全ての隣接領域に対して非ブロッキング通信 `MPI_Isend()` および `MPI_Irecv()` を一斉に発行し，`MPI_Waitall()` により全ての通信完了を待機する方法とする．したがって送り側のプロセスと受け手側のプロセスはユーザープログラムからは陽的には同期させておらず，送り側と受け手側の接続を確立するタイミングは MPI を処理する OS 側に委ねる。

図 4.9 に示すように，送信バッファ `sbuf` は送信する領域番号ごとに昇順にデータを先頭から詰めていく．`nsbuf[j]` は，送信する領域番号に対する送信メッシュ数であり，0 の場合（すなわち領域どうしが隣接していない場合）は `if` 文により `MPI_Isend` の発行をしないようにして OS の MPI 処理を軽減している．`nsobuf[j]` が，データの送信を開始する `sbuf` の先頭アドレスからのオフセット値である．`nsbuf` および `nsobuf` は不変なので初期設定時に設定して保持しておくことで，通信実行時の処理を軽減している．同様に `MPI_Irecv` により受信バッファ `rbuf` に各領域からの受信データを詰め込み，全ての通信が終わった後に袖領域に再配置する（図 4.10）。

4.1.4 オーバーラップ通信による通信時間の隠蔽手法

大規模並列計算において実用的なスケーラビリティを達成するためには，演算時間に対して通信時間の割合を低減することが必要であり，演算時間が CPU 計算と比較して短くなる GPU 計算の場合，並列化効率が低下する可能性が高くなる．そこで袖領域の通信時間を GPU

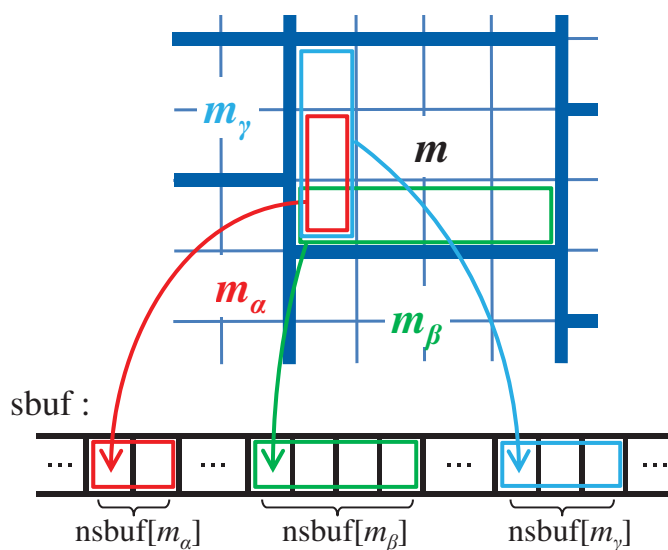


図 4.9 Packing of send buffer.

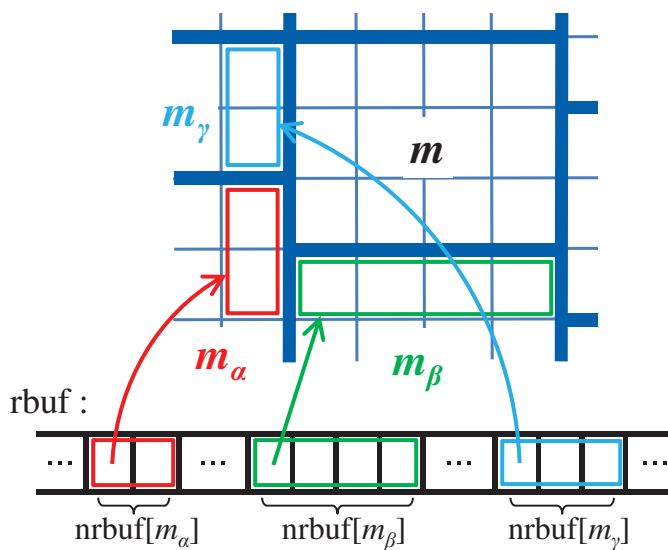


図 4.10 Exploring to sleeve meshes from receive buffer.

計算時間で隠蔽するためのオーバーラップ通信を採用する．図 4.11 にアルゴリズムを示す．図中の赤太線部が GPU によるカーネル計算，青二重線部が袖領域通信を示す．図 4.9 の sbuf へのデータパッキングの後，GPU カーネルのメッシュ番号 N_b から $N - 1$ ，すなわち袖領域に接している境界メッシュ以外のメッシュの GPU 計算を開始する． N_b は袖領域に接している境界メッシュの数を表し，0 から $N_b - 1$ が境界メッシュ， N_b から $N - 1$ がそれ以外のメッ



シュとなるようにメッシュ番号を付けておく．この GPU 計算と並行して，

1. sbuf の GPU→CPU 転送
2. sbuf→rbuf の MPI 非ブロッキング通信
3. rbuf の CPU→GPU 上の袖領域への転送

を行う．この処理が N_b から $N - 1$ の GPU 計算よりも先に終了すれば，通信処理時間の完全な隠蔽が達成される．GPU 上の sbuf と rbuf の直接通信は一般的なサーバーでは不可能なため，CPU–GPU 間通信が余分に必要なことに注意する．しかし境界不整合領域分割では，適切な三次元領域分割数を指定することにより，通信データ量を最小化することができるため，通信処理時間の隠蔽の割合が，通常の一次元領域分割よりも高まると考えられる．

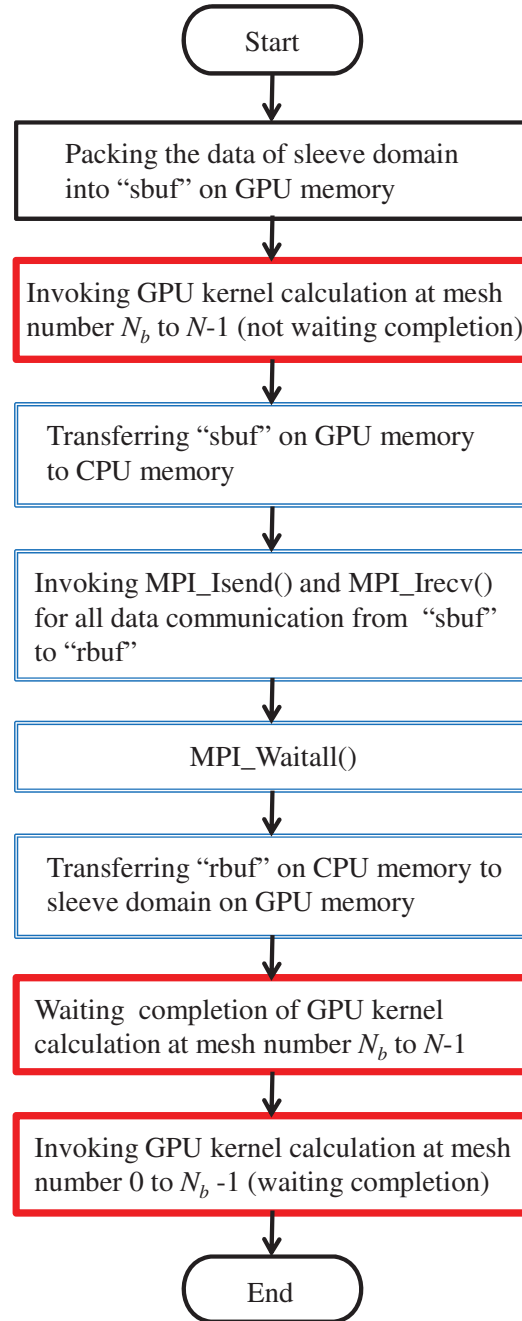


図 4.11 The algorithm of overlapped communication in GPU code.



4.2 加速化手法による計算効率向上の検証

4.2.1 GPU 化による加速効果

GPU を搭載した 1 台のサーバーを用いて、GPU 化による計算速度の高速化割合を測定した。衛生陶器シミュレーションの圧力方程式の求解において、PCG 法 (Preconditioned Conjugate Gradient method[100]) を利用したときの計算時間を CPU と GPU とで比較した結果を表 4.1 に示す。前処理は単純な点ヤコビ法 [100] を用いた。衛生陶器の流れのシミュレーションにおいては、一般的な非圧縮性流れと同様に圧力方程式の求解が全体の計算時間の大半を占めるため、本テストでは圧力方程式の計算のみとし、気液界面計算や移流項、粘性項の計算は含んでいない。全体矩形領域のメッシュ分割数は $293 \times 387 \times 475 = 53,860,725$ 、計算領域のメッシュ数は 4,770,858 とした。PCG 法の反復回数は 1000 回とした。計算結果から、1GPU の計算性能は 1CPU の約 6 倍程度と判明した。したがって GPU アクセラレータの衛生陶器シミュレーションへの適用は極めて有効であることが分かった。

4.2.2 オーバーラップ通信による効果

境界不整合領域分割を用いたときの各プロセスの計算負荷および通信時間について、前節の計算条件における $2 \times 2 \times 2$ の領域分割の結果を図 4.12 に示す。本テストでは、圧力方程式の求解の中で大半の計算時間を占める行列とベクトルの積の計算について測定した。横軸に 8GPU の各プロセス番号、縦軸に行列・ベクトル積 1 回における実行時間を表す。三角が GPU 計算実行時間、四角が他の GPU との通信時間を表す。この通信時間の中には、GPU、CPU 間のデータ転送時間も含む。

8GPU の中で最も計算時間が長いものは GPU 番号 0 の 0.665ms であり、8GPU の平均の 0.598ms に対して約 11% 大きかった。これは複雑な流路を持つ実際製品の計算において、インデックス配列 $im[i]$, $ip[i]$, $jm[i]$, $jp[i]$ で示される隣接メッシュのメモリアクセスパターンが、各分割領域間で差があるためと推測される。しかしながら GPU 間の計算時間のばらつきは、実製品への適用において十分実用的な範囲といえる。また通信時間のばらつきも比較的大きいが、いずれも GPU 計算時間の範囲内に収まっており、オーバーラップ通信による通信時間隠蔽は達成されていることが分かった。

4.2.3 GPU 化および並列化によるスケーラビリティの検証

東京工業大学の GPU スーパーコンピュータ「TSUBAME2」[97] を用いて提案計算手法の大規模並列計算における並列化効率の測定を実施した。メッシュ間隔 0.5mm、全体矩形領域

表 4.1 Comparison of the performance between CPU and GPU.

	1CPU	1GPU
	Xeon 3.6GHz	Geforce 690
Elapse time	64.3 sec	10.6 sec
Speedup ratio (1CPU = 1)	1.0	6.07

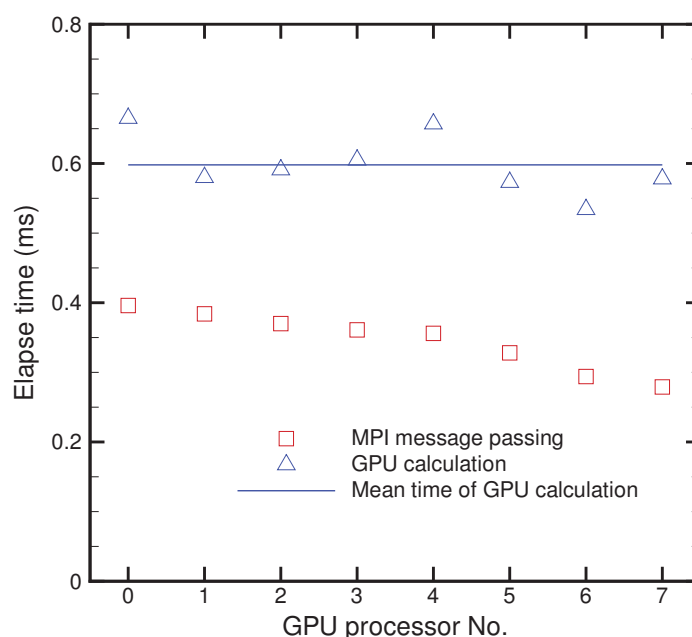


図 4.12 The result of measuring load balance of each GPU.

のメッシュ分割数 $N_x \times N_y \times N_z = 880 \times 1320 \times 1426 = 1,656,441,600$, 計算領域メッシュ数=152,347,494 の実製品衛生陶器シミュレーションにおいて, 計算時間の大半を占める圧力方程式求解時間を, PCG 法と MGCG 法 (Multi-Grid Conjugate Gradient method[100]) とで比較した. MGCG 法は, 前処理にマルチグリッド法 [101] を用いた手法である.

図 4.13 に, 計算領域の境界不整合領域分割の結果を示す. 衛生陶器の流路モデルは製品形状に忠実に作成されており, 極めて複雑な構造であることが分かる. 図 4.13 の $2 \times 2 \times 3$ 領域分割の場合, 最もメッシュ数の少ない領域で 12,472,585 メッシュ, 最も多い領域で 13,041,028 メッシュであり, その差は約 5% であった. また $4 \times 5 \times 6$ 領域分割の場合, 最もメッシュ数の少ない領域で 1,210,259 メッシュ, 最も多い領域で 1,345,800 メッシュであり, その差は約 11% であった.



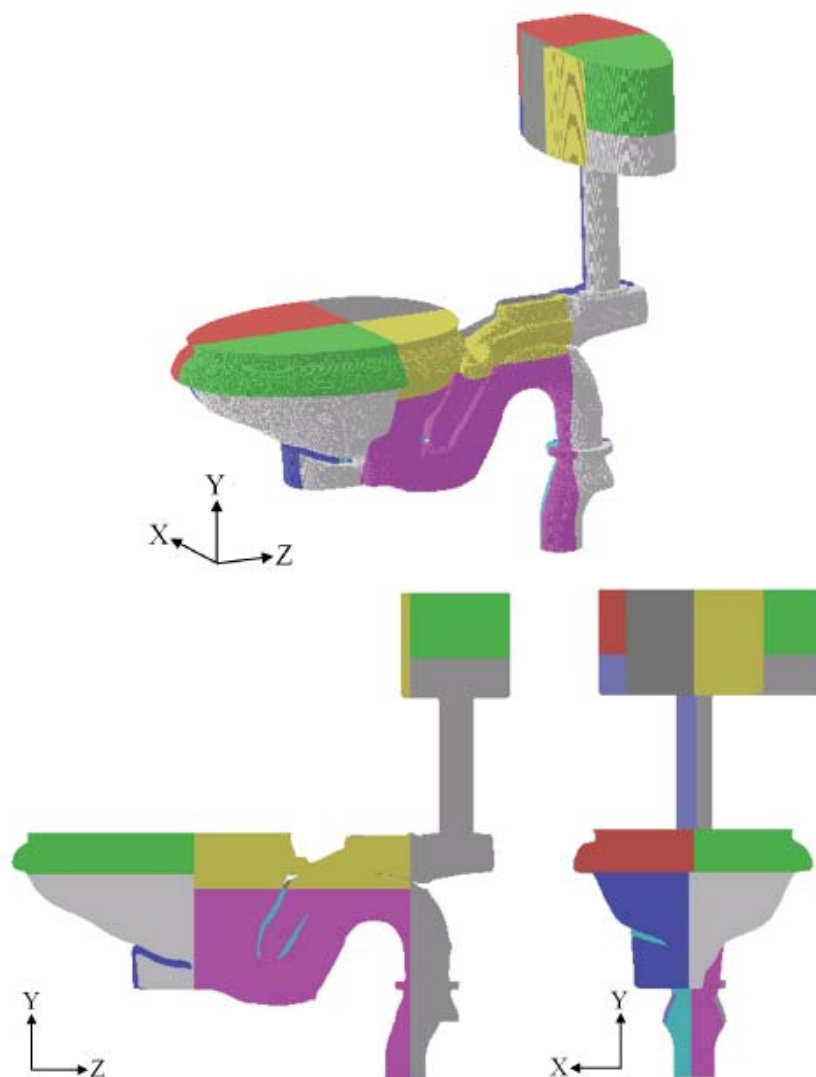


図 4.13 The test model of the sanitary ware (Ex. $2 \times 2 \times 2$ domain decomposition).

1 タイムステップにおける測定結果を図 4.14 に示す．PCG 法の GPU 数が 1 の場合に対し，GPU 数を増加させたときの計算速度向上率を縦軸に取っている．三次元領域分割における軸ごとの領域分割数は，最も計算時間が短かったもののみ，図 4.14 に掲載している．反復回数は収束判定値 1×10^{-4} の場合，PCG 法で 5,286 回，MGCG 法で 37 回であった．

PCG 法の場合は，60GPU 程度まで理想的なスケーラビリティが得られており，また 150GPU 利用の場合でも約 100 倍の計算速度向上が得られることが分かった．MGCG 法の場合は PCG 法よりも求解の収束が速いため，計算速度向上率が高いが，理想的なスケーラビリティが得られているのは 30GPU 程度までと判明した．しかしながら 120GPU 利用時に PCG 法 1GPU の 300 倍の計算速度向上が得られている．このときの Tsubame における計算時間は PCG 法 14.8 秒に対し，MGCG 法は 4.5 秒である．計算時間刻み Δt はクーラン

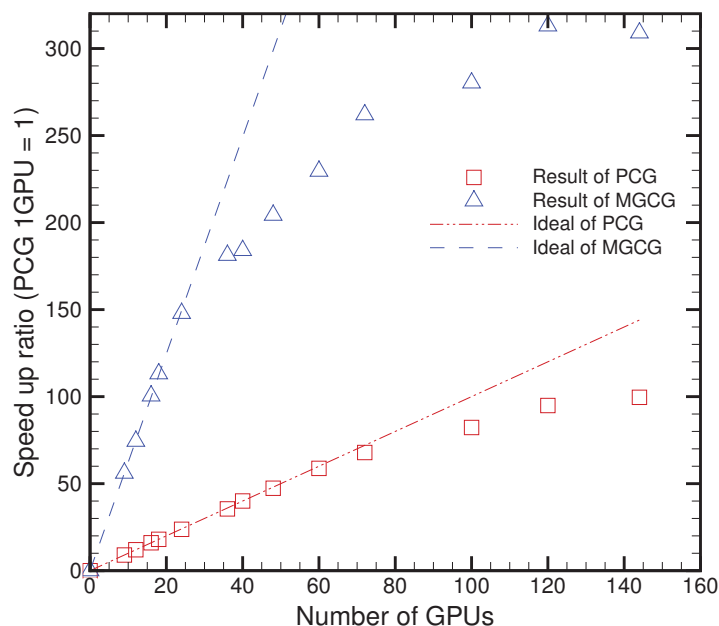


図 4.14 The result of performance measurement using TSUBAME supercomputer.

条件により毎ステップ変化するが、0.5mm メッシュにおいて流速の最大値は高々 10 m/s であり、おおよそ Δt の平均値は 3×10^{-5} 秒と見積もられる．計算開始から終了までの $t = 3$ 秒間の流体計算の場合、計算ステップ数は 10^5 と見積もられ、実際の流体計算でのステップ数とほぼ同等であることを確認した．MGCG 法の圧力方程式の総計算時間は 4.5×10^5 秒 = 5.2 日であることから、流体計算全体では約 1 週間程度で計算可能と分かった．したがって、メッシュ間隔の最大目標値である 0.5mm 高精細メッシュによる気液二相流体シミュレーションが十分実用的な計算時間で実施可能となった．

本テストの条件においては、PCG 法の場合は 60 並列、また MGCG 法の場合は 30 並列程度で、オーバーラップ通信における通信時間と GPU 計算時間が釣り合うものと推定される．MGCG 法の方が並列化効率が悪いのは、MGCG 法における前処理部の計算は、通常の直交格子メッシュ分割数の各軸方向半分のメッシュ数で行われるため、通信量の割合が GPU の計算量に対して大きくなるためと推測される．



4.3 大規模計算による衛生陶器シミュレーション結果

図 4.15 に前節の 0.5 ミリメッシュ計算モデルによる TSUBAME でのシミュレーション結果を示す。本シミュレーションでは、前章で提案した Navier-Stokes 流体方程式計算手法を用いた。UTI-VSIAM3 による移流計算，UTI-VSIAM3+STAA による気液界面捕獲，二次精度中心差分法による粘性項計算および MGCG 法による圧力ポアソン方程式解法のいずれの計算コードも完全に GPU 化している。流体計算では，初期状態として上部のタンクには規定の水位にて水を充填した。その状態でタンクから水を自由落下させ，水の流れを解析した。計算結果の可視化は，流体率から気液界面のポリゴンを生成して，そのポリゴンを POV-Ray の mesh2 オブジェクトにより表示した。衛生陶器の形状は，水と同様に mesh2 オブジェクトで透明体により表した。本シミュレーション結果により，陶器表面の薄膜流れやトラップ内の微小気泡の動きなどが精密に捉えられていることが分かった。トラップ内部の流れは極めて複雑であるが，計算は破綻することなく終始安定していた。図 4.16 に実機との定量評価として，0.5 ミリメッシュと従来の 1.5 ミリメッシュにおける水位高さの時系列変化を示す。解像度の高い格子を用いることでより実機流れに近い水位が得られていることが分かる。

次の例として，複数剛体を含む衛生陶器のシミュレーション結果を図 4.17 に示す。固体の動きは前章で示した定式化を用いた。流体計算では，初期状態として衛生陶器のトラップ部に水を充填し，その表面に比重 0.95 の円柱状の固体 6 個を横倒しで浮かべた。また上部のタンクには規定の水位にて水を充填した。その状態でタンクから水を自由落下させ，水の流れおよび固体の動きを解析した。メッシュ間隔は本モデルでは 0.8mm メッシュを採用し，全体矩形領域のメッシュ分割数は $515 \times 877 \times 939 = 424,104,045$ ，計算領域のメッシュ数は 43,605,975 であった。領域分割 $2 \times 2 \times 8 = 32$ において 1 サイクルあたりの計算時間は 4.1 秒であり，150,000 サイクルの計算において総計算時間は $4.1 \times 150,000$ 秒 = 7.1 日であった。

0.8mm 間隔の高精細メッシュによる大規模並列計算により，無数の微小気泡の形状および動きを捕捉することに成功した。微小気泡群の動きを追跡することで，水の主流はボウル部鉛直軸に沿って竜巻状のらせん渦を構成しており，この渦にのって固体が下流に流れていくことが明らかとなった。衛生陶器の製品開発においては，このらせん渦の強さをいかに高めていくかが大きな設計ポイントであることが分かった。

このように衛生陶器の固体・気体・液体の三相流高精度シミュレーションでは，汚物を固体と見立てて，コンピュータ上で仮想的な汚物洗浄実験が可能となるだけでなく，その洗浄力を向上させるための設計変更・最適化のための知見出しにも大きな利用価値があるといえる。



図 4.15 A simulation result of a sanitary ware using TSUBAME supercomputer.

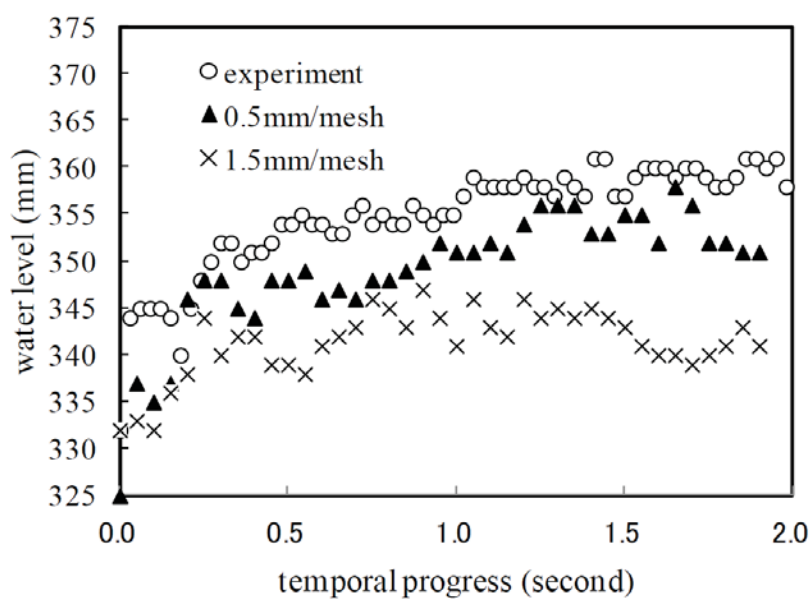


図 4.16 The water level at a probe point. Displayed are the numerical results computed with 1.5mm and 0.5mm grid resolutions respectively in comparison with the experiment measurement..

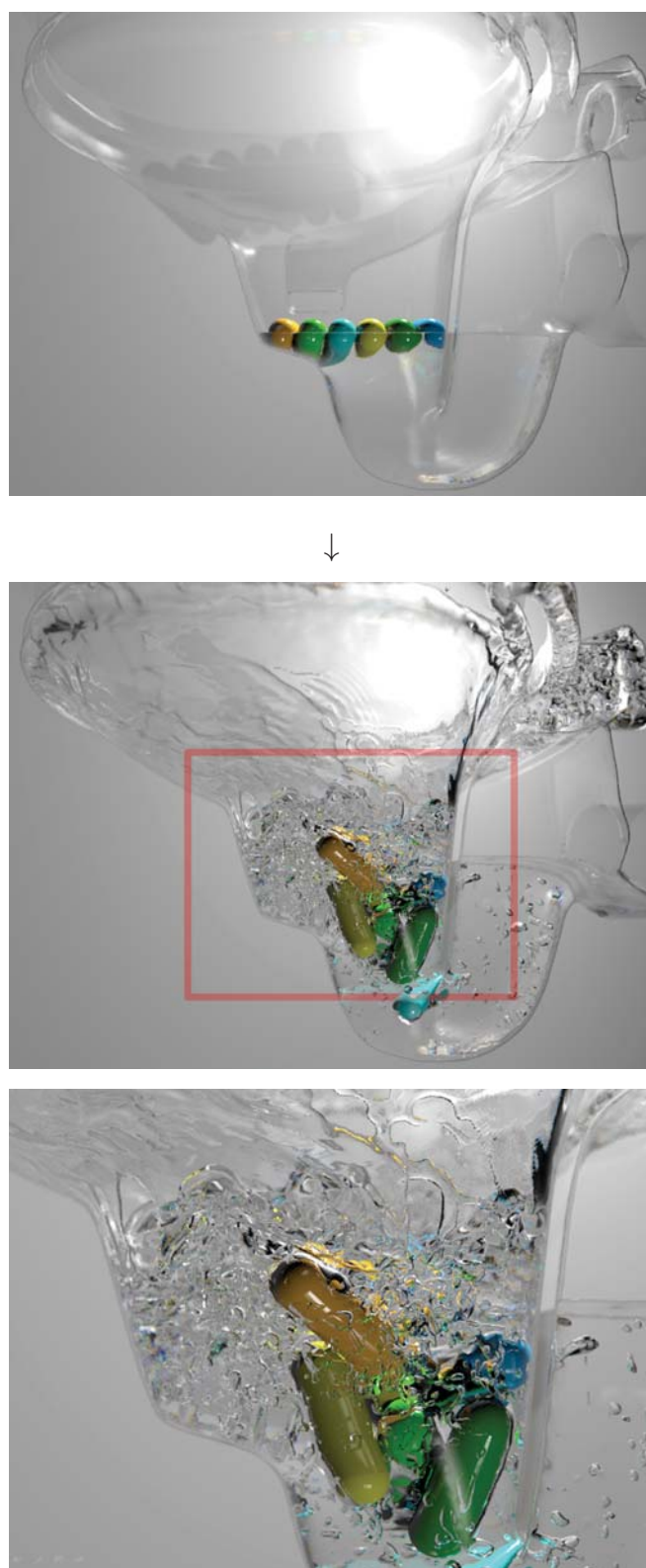


図 4.17 Another simulation result of a sanitary ware using TSUBAME supercomputer.

4.4 第 4 章のまとめ

第 3 章で提案した構造格子多相流シミュレーション手法に対してコードの GPGPU 化および並列計算手法を適用することにより大規模計算における飛躍的な高速化を実現した。UTI-VSIAM3 および STAA に基づく有限体積法では計算手法およびアルゴリズムの簡潔さにより、プログラム全域にわたる GPGPU 化および領域分割並列計算が容易に適用できる。GPGPU 化により CPU 計算の 6 倍以上の高速化を達成した。1 プロセスにおける計算速度が大幅に速くなると、マルチプロセス並列計算におけるスケーラビリティは一般に悪化する傾向にあるが、境界不整合領域分割やオーバーラップ通信の適用により、実用的な並列化効率を実現した。GPGPU クラスター型スーパーコンピュータの利用により、1 億メッシュレベルの大規模な衛生陶器二相流シミュレーションを実用的な計算時間内で計算できるようになった。したがって提案の計算手法は実用製品のシミュレーションにおいて極めて有効であることを実証した。

第 5 章

運動量保存性の改良

5.1 多相流における運動量保存性の問題

多次元の Navier-Stokes 流体方程式は以下の式で表される.

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = \text{圧力項} + \text{粘性項} + \text{外力項} \quad (5.1)$$

この式と質量保存則

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (5.2)$$

を用い数学的に変形すると, 下式 of 非保存型の Navier-Stokes 方程式が得られる. この式の導出過程では, 非圧縮性などの仮定を一切おいていないことに注意する.

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = \frac{1}{\rho} (\text{圧力項} + \text{粘性項} + \text{外力項}) \quad (5.3)$$

この式の左辺の移流項は, 二相流における気液界面の移流方程式

$$\frac{\partial F}{\partial t} + \mathbf{u} \cdot \nabla F = 0 \quad (5.4)$$

と全く同じ形をしている. F は VOF (Volume Of Fluid) 関数であり, 一つのセルの中の流体占有体積率を表す. 式 (5.4) の流体率 F は $0 \sim 1$ の値に制限され, しかも気液界面での不連続性を保持するために, VOF 法 [41] や THINC 法 [44] などの特別な方法で計算されるのに対し, 式 (5.3) の流速 $\mathbf{u}$ の移流については, $\mathbf{u}$ は気液界面においても連続的な値を持ち, かつ値の範囲は $-\infty \sim \infty$ で制限がないことから, MUSCL 法 [78] などの一般的な移流方程式解法で計算される. ゆえに F と $\mathbf{u}$ の移流項は計算手法が異なるため, それぞれの計算手法が持つ計算誤差が結果に影響を及ぼすことを考慮する必要がある [102]. 例えば図 5.1 に示すような静止した空気中に運動している液滴がある場合に, F と $\mathbf{u}$ の移流項の解き方が異なるため, 計算を進めていくと液滴の位置と液滴の流速にずれが生じることが考えられる. このような状況が生じると, 液滴の動きを正しく捉えることはできない.



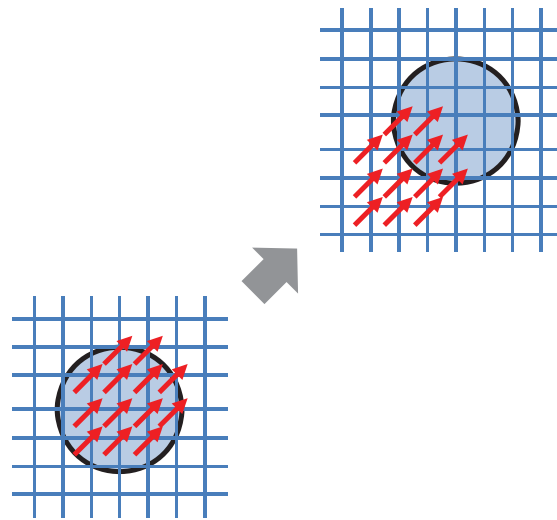


図 5.1 The error of VOF and velocity advection equation. The arrows indicate the velocity field.

この問題に対処するためには，保存型の Navier-Stokes 方程式 (5.1) を解く必要がある．この方程式は，単位体積あたりの運動量 $\rho \mathbf{u}$ が保存されるという意义があり，図 5.1 のような運動量保存性の破綻は生じないと考えられる．しかしこの方法では，左辺第一項の時間項の計算が $\mathbf{u}$ ではなく $\rho \mathbf{u}$ に関する式になっていることから，陽解法においては $\mathbf{u}$ ではなく $\rho \mathbf{u}$ が計算されることになる．したがって式 (5.4) で計算された流体率 F から

$$\rho = (1 - F)\rho_{gas} + F\rho_{liq} \quad (5.5)$$

により計算される ρ を用いて，

$$\mathbf{u} = \frac{\rho \mathbf{u}}{\rho} \quad (5.6)$$

により流速を求める必要がある．式 (5.6) の問題点は，水と空気のような密度比が 1,000 倍にも達する場合，式 (5.4) の移流計算によって式 (5.6) の分母の密度が大きく変動するため，移流計算誤差によっては， $\mathbf{u}$ が非常に大きな値となり，流体計算が破綻する可能性があることである．この不安定性を避けるために，従来の気液二相流シミュレーションでは式 (5.3) の非保存型運動方程式解法が広く用いられてきた．

この保存型解法における計算不安定性の問題を解決するために，運動量と密度の移流計算をカップリングして実行するアルゴリズム [103][104][105][106][107][108] が提案されている．気液界面追跡にレベルセット法を使用した方法 [104][105][107][108] では，移流項の計算にお

いて

$$\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = 0 \quad (5.7)$$

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (5.8)$$

を解く．式 (5.8) は式 (5.4) とは別に計算される．式 (5.7), (5.8) の左辺第二項の $\rho \mathbf{u}$ は前のサイクルの ρ および $\mathbf{u}$ を代入して同じ移流計算手法により左辺の $\rho \mathbf{u}$, ρ を計算する．この結果を式 (5.6) に代入して流速 $\mathbf{u}$ を求める．ここで式 (5.8) で計算された ρ は破棄され，次のサイクルの計算には使用しないことに注意する．Navier-Stokes の式全体で用いられる ρ は，通常の計算手法と同じく式 (5.5) から計算されたものを使用する．この式 (5.8) と式 (5.5) により計算される ρ のずれは移流計算手法の相違により生じるものであり，その結果運動量の保存性は厳密には満足されない．しかし式 (5.7) を採用することで，従来手法の不具合を解消できると考えられる．

5.2 提案計算手法の詳細

5.2.1 運動量保存式の定式化

衛生陶器の流れを安定的かつ高精度に計算するために，保存型運動方程式において VSIAM3 を適用する．式 (5.6) の分母の ρ を管理することで計算を安定させる新しい手法を以下に提案する．

まず移流計算を行う前に，現在の時刻の流体率 F の VIA, SIA より密度 ρ の VIA, SIA を計算する．

$$\rho^{v(n)} = \left(1 - F^{v(n)}\right) \rho_{gas} + F^{v(n)} \rho_{liq} \quad (5.9)$$

$$\rho^{x(n)} = \left(1 - F^{x(n)}\right) \rho_{gas} + F^{x(n)} \rho_{liq} \quad (5.10)$$

$$\rho^{y(n)} = \left(1 - F^{y(n)}\right) \rho_{gas} + F^{y(n)} \rho_{liq} \quad (5.11)$$

$$\rho^{z(n)} = \left(1 - F^{z(n)}\right) \rho_{gas} + F^{z(n)} \rho_{liq} \quad (5.12)$$

次に X 方向の移流計算では，以下の手順を行う．

1. 現在の時刻の密度の SIA $\rho^{x(n)}$ を補正

$$\rho^{x(n)} = \begin{cases} \rho_{up}^v & \left(\rho_{up}^v < \rho_{lim}\right) \\ \rho^x & \left(\rho_{up}^v \geq \rho_{lim}\right) \end{cases} \quad (5.13)$$

2. 密度の一次元移流計算により，密度の VIA $\rho^{v(*)}$ および SIA $\rho^{x(*)}$ を更新

$$\frac{\partial \rho^{(n) \rightarrow (*)}}{\partial t} + u^{(n)} \frac{\partial \rho^{(n)}}{\partial x} = 0 \quad (5.14)$$



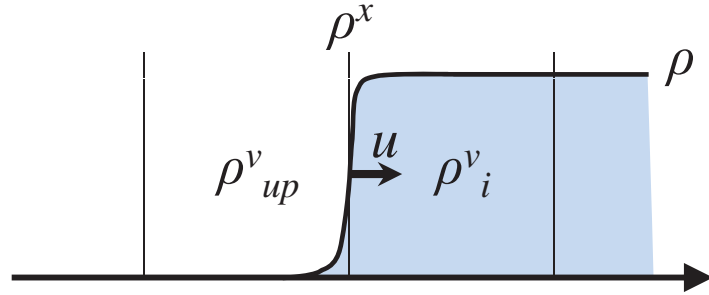


図 5.2 The correction of density.

3. 密度の残りの SIA $\rho^{y(*)}$, $\rho^{z(*)}$ を TEC[77] により更新
4. $\varphi \equiv (u, v, w)^T$ について以下の一次元移流計算を行う
 - (a) 現在の時刻の ρ および φ より, 現在の時刻の運動量 $(\rho\varphi)^n$ の VIA および SIA を計算する

$$\text{VIA} : (\rho\varphi)^{v(n)} \equiv \rho^{v(n)} \cdot \varphi^{v(n)} \quad (5.15)$$

$$\text{SIA} : (\rho\varphi)^{x(n)} \equiv \rho^{x(n)} \cdot \varphi^{x(n)} \quad (5.16)$$

- (b) 運動量の一次元移流計算により, 運動量の VIA $(\rho\varphi)^{v(*)}$ および SIA $(\rho\varphi)^{x(*)}$ を更新

$$\frac{\partial(\rho\varphi)^{(n) \rightarrow (*)}}{\partial t} + u^{(n)} \frac{\partial(\rho\varphi)^{(n)}}{\partial x} = 0 \quad (5.17)$$

- (c) 流速を計算する

$$\varphi^{v(*)} = \frac{(\rho\varphi)^{v(*)}}{\rho^{v(*)}} \quad (5.18)$$

$$\varphi^{x(*)} = \frac{(\rho\varphi)^{x(*)}}{\rho^{x(*)}} \quad (5.19)$$

- (d) 流速の残りの SIA $\varphi^{y(*)}$, $\varphi^{z(*)}$ を TEC により更新

以上の手順を, Y 方向および Z 方向の移流計算についても同様に行う.

VSIAM3 では面積積分平均値 SIA は主に移流計算精度を向上させる働きとして使用され、方程式全体で利用される物理量としては体積積分平均値 VIA が多くの場合利用される。したがって提案手法では、この面積積分平均値を修正することで、全体の計算安定性を確保するとともに、計算の妥当性を担保する。すなわち面積積分平均値の式 (5.19) の不安定性が生じるのは分母の ρ^x が液体から気体に変わり急激に小さくなる場合であることに着目し、そのような状況を防止するために、式 (5.13) により補関数を変え、計算の不安定性回避する。 ρ^x が急激に低下するのは、図 5.2 のように上流側の流体率が 1 から 0 に急激に変わる場合であるので、上流側のメッシュの体積積分平均密度 ρ_{up}^v がしきい値 ρ_{lim} よりも小さい場合、面積積分平均密度 ρ^x を ρ_{up}^v に修正する。本修正において ρ_{lim} を様々な計算条件において都度適正化することで、衛生陶器シミュレーションにおいて計算の安定性を大幅に向上させることができた。このような値の修正は面積積分平均値のみであり体積積分平均値は修正しないため、移流計算における精度の問題は軽微と考えられる。

本アルゴリズムを VSIAM3-CM(Conservative Momentum) とよぶことにする。従来の VSIAM3 もしくは UTI-VSIAM3 を用いた非保存型運動量解法と比較してアルゴリズムが複雑となり計算量が増加しているが、流体計算全体では圧力方程式の求解が大半の計算時間を占めるため、全体の計算時間に占める増加割合は計算モデルや条件により数 % から数十 % 程度である。



5.3 ベンチマークテストによる計算精度検証

5.3.1 移動液滴のベンチマークテスト

運動量保存性の基本的な検証問題として、無重力下の静止空気中において直線移動する二次元の真円液滴のテスト計算 [109] を実施した。本テストでは、気体の密度は $\rho_g = 1$ とし、液体と気体の密度比は $\rho_l/\rho_g = 10^6$ とした。粘性係数は気体、液体ともに 0 とした。計算領域は $[0, 1] \times [0, 1]$ である。計算メッシュは均一な 128×128 メッシュを用いた。液滴は初期に $(0.4, 0.5)$ の位置に置かれ、半径は $r = 0.1$ とした。また初期流速は $F > 0$ のメッシュを $u = 1$, $F = 0$ のメッシュを $u = 0$ とした。

$T = 0.2$ における計算結果を図 5.3 に示す。上が従来の UTI-VSIAM3 を非保存型運動方程式の移流項に用いた結果であり、下が VSIAM3-CM の結果である。本テストでは表面張力はなしとしているため、液滴形状が崩れた場合に円形に復元する作用はない。したがって従来手法では、初期の円形状は大きく変形しており、大きな計算誤差が発生していることがわかる。一方提案手法ではほぼ円形状を保ったまま右に並行移動しており、精度の高い結果が得られた。

5.3.2 強制対流中の静止液滴のベンチマークテスト

無重力下において三次元の静止球体液滴を置き、その周りの空気に強制流れを与える新たなベンチマークテストを提案する。本テストでは、気体の密度は $\rho_g = 1$ とし、液体と気体の密度比は $\rho_l/\rho_g = 10^3$ とする。粘性係数は気体、液体ともに 0 とする。表面張力はなしとする。計算領域は $[0, 0.1] \times [0, 0.1] \times [0, 0.1]$ である。計算メッシュは均一な $100 \times 100 \times 100$ メッシュを用いる。液滴は初期に $(0.03, 0.03, 0.03)$ の位置に置かれ、半径は $r = 0.015$ とする。また初期流速は全領域を 0 とし、流入境界条件として上流側の境界面に $\mathbf{u} = (1, 1, 1)$ を与える。以上の条件において、液体は気体よりも 10^3 倍重いため、液滴は気体の流れにおいてもほとんど動くことはないはずである。数値計算ではこの状況を再現するかどうか、運動量保存性の検証となる。

$T = 5 \times 10^{-2}$ における計算結果を図 5.4 に示す。上が従来の UTI-VSIAM3 を非保存型運動方程式の移流項計算に用いた結果であり、下が VSIAM3-CM の結果である。明らかに提案手法の方が球形を保持しており、運動量保存性が高いことが分かる。一方従来手法では球体は原型を止めないほど崩れており、実現象では起こりえない、運動量保存則の破綻が明確に現れる結果となった。

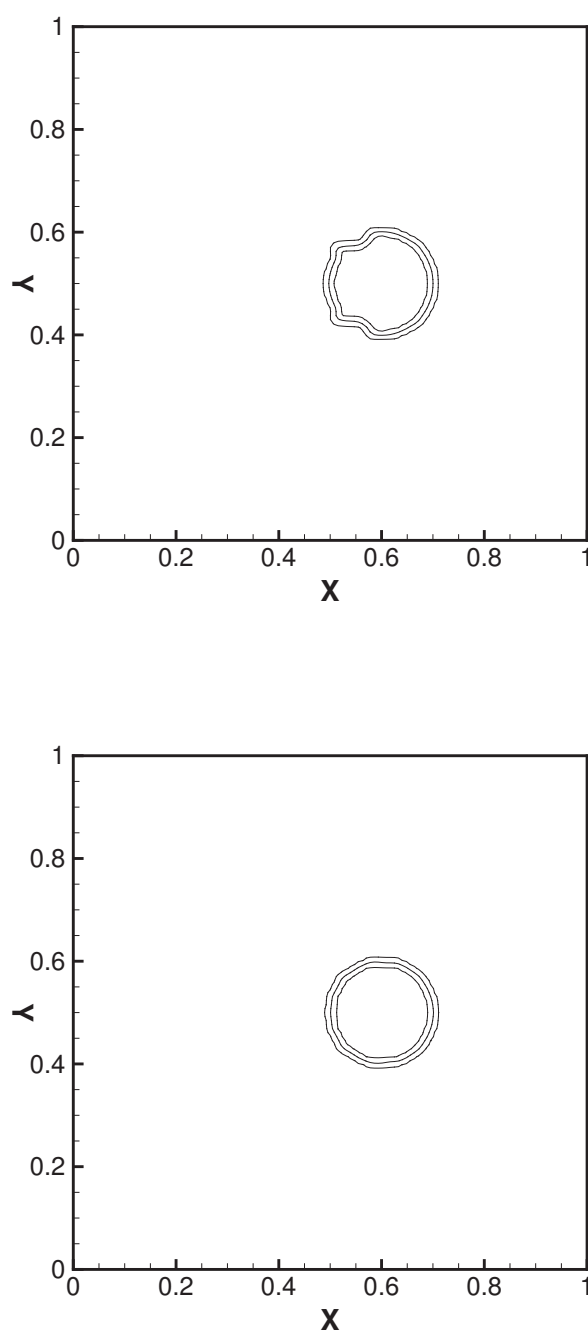


図 5.3 The result of benchmark test of moving particle in the static atmosphere without surface tension. Upper panel is the result of the calculation method where momentum equations are solved by UTI-VSIAM3, lower is that of VSIAM3-CM.



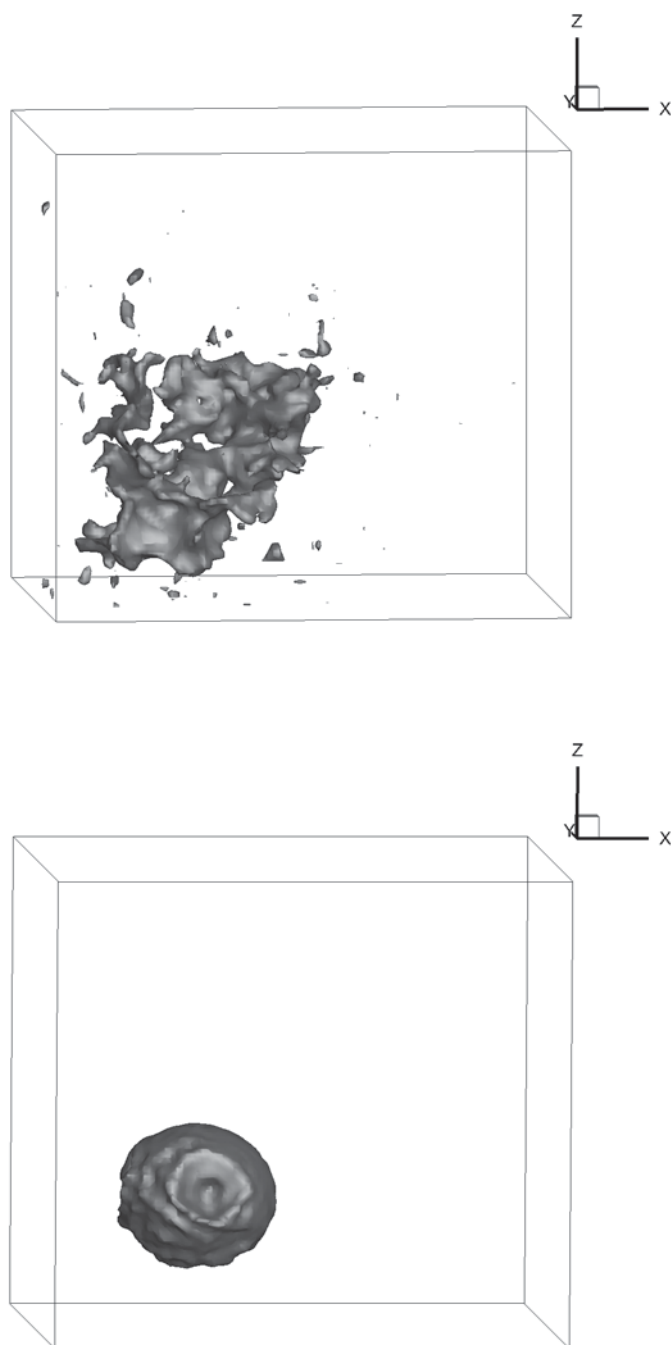


図 5.4 The result of benchmark test of the static particle in the air flow without surface tension. Upper is the result of the calculation method where momentum equations are solved by UTI-VSIAM3, lower is that of VSIAM3-CM.

5.3.3 三次元のダム崩壊ベンチマークテスト

三次元のダム崩壊テストシミュレーション結果を示す (図 5.5). このテストでは $\rho_{gas} = 1.226 \text{ kg/m}^3$, $\rho_{liq} = 1000 \text{ kg/m}^3$ とした. 表面張力係数を $\sigma = 0.0728 \text{ N/m}$, 粘性係数を $\mu_{gas} = 1.78 \times 10^{-5} \text{ Pa}\cdot\text{s}$, $\mu_{liq} = 1.137 \times 10^{-3} \text{ Pa}\cdot\text{s}$ とした. 計算領域は $[0, 0.4] \times [0, 0.1] \times [0, 0.1]$ とし, Y 軸方向に重力加速度 -9.81 m/s^2 とした. 初期に $[0, 0.05715] \times [0, 0.05715] \times [0, 0.1]$ の領域内を $F = 1$ とし, それ以外の領域を $F = 0$ とした. 計算メッシュは均一な $512 \times 128 \times 128$ メッシュを用いた. 壁面境界条件は, 全ての壁面において NO SLIP 条件とした.

図 5.5 のシミュレーション結果では従来手法と提案手法とでほぼ同等の結果が得られており, VSIAM3-CM の計算の不安定性も生じなかった. したがって提案アルゴリズムにおける密度修正による計算安定化効果が得られており, また移流計算精度への影響も極めて小さいことが分かった.



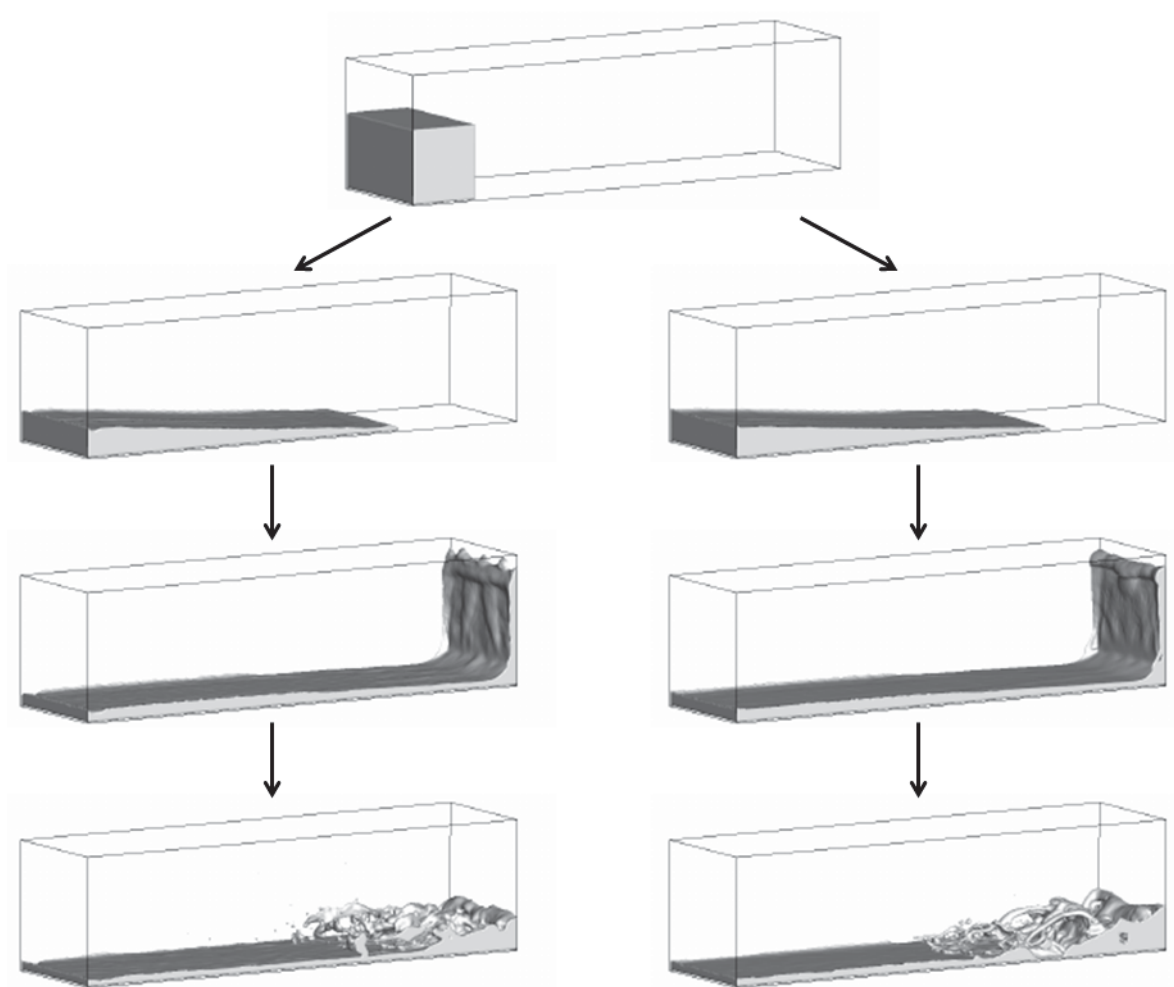


図 5.5 The result of dam break benchmark test. Left is the result of UTI-VSIAM3. Right is that of VSIAM3-CM.

5.4 衛生陶器シミュレーションでの検証

運動方程式解法に VSIAM3-CM を用いて衛生陶器シミュレーションを実施した．計算領域全体はメッシュ数を $489 \times 681 \times 889$ としており，0.8mm 間隔の格子を用いた．また乱流モデルとして，LES のスマゴリンスキーモデルを採用した．

メッシュ数が比較的多いため，東京工業大学のスーパーコンピュータ「TSUBAME2.5」を利用した．TSUBAME 2.5 は最新の NVidia CUDA GPU を搭載したサーバーを高速の InfiniBand により相互結合した高効率な GPU スパコンであり，本研究に適合する．

図 5.6 は吐水口付近の水の流れのようすを緑色で表示したものである．運動方程式解法に UTI-VSIAM3 を用いた従来手法では，実機では発生しない鉛直方向上方への飛沫の跳ねが多く観察された．これは，吐水口手前の流路から吐水口に吐水される際に，出口手前で圧力開放が発生し，その際の運動量保存性の誤差により，三次元静止液滴ベンチマークテストと同様の数値計算上の水跳ねが生じたものと推測される．一方提案手法では実機と同様に，飛沫の跳ねは吐水方向の左方水平方向に限定されており，実現象に沿った妥当な結果を得ることができた．計算時間は，従来の計算手法と比較して 30% 程度の増加となった．ただし総計算サイクル数はほぼ同等であり，計算不安定によるクーラン条件に基づく Δt の微小化は発生しておらず，計算安定性に問題はないことが分かった．

なお吐水の主流速度は UTI-VSIAM3 と VSIAM3-CM とでほぼ同等であり，ダム崩壊テストシミュレーション同様，全体的なシミュレーション結果の傾向は変わらないことも分かった．したがって飛沫の動きなどの細部のシミュレーション結果が必要な場合は提案手法は有効であるといえるものの，従来の非保存型解法は相対的に計算量が少なく，多くの適用問題に対して十分利用可能であると考えられる．





図 5.6 The result of numerical simulation of a sanitary ware. Upper is the result of UTI-VSIAM3, lower is that of VSIAM3-CM.

5.5 第 5 章のまとめ

保存型の Navier-Stokes 方程式の移流項を直接計算する新しい移流計算手法 (VSIAM3-CM) を提案し, ベンチマークテスト計算や衛生陶器シミュレーションを通して有効性を示した. VSIAM3-CM では, 従来の VSIAM3 や UTI-VSIAM3 での非保存型 Navier-Stokes 方程式解法とは異なり, 流速 u に関する移流方程式ではなく運動量 ρu に関する輸送方程式に変わるため, 運動量保存性が大幅に向上し, 気液界面での非物理的な変形や微小液滴の不自然な動きを抑制できることを実証した. 衛生陶器の洗浄時水はねのシミュレーションでは非保存型解法で生じていた過剰な水はねの問題を解消した. したがって提案手法は妥当であると結論付けられる.



第 6 章

任意多面体非構造格子の気液二相流体計算手法

本章では，第 3 章の直交格子多相流シミュレーションモデルを任意多面体格子に拡張する新しい計算手法およびフレームワークを提案し，数値計算テストによる検証結果および衛生陶器シミュレーションへの適用結果について述べる．

6.1 VSIAM3 による移流計算手法の多面体格子への拡張

6.1.1 計算手法の詳細

任意多面体非構造格子において，輸送方程式

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}) = 0 \quad (6.1)$$

を高精度な有限体積法により計算することを考える．

いま任意多面体格子において，直交格子と同様に， ϕ_C のほかに格子界面値 ϕ_f も保持する．まず直交格子の一次元 CIP-CSL 法を任意多面体に適用するために，仮想上流点 u の値 ϕ_u を以下のように補間計算する．点 C ，点 f および点 u は一直線上にあるとする．

$$\phi_u = \phi_f + 2 \left(\nabla \phi_C \cdot \vec{fC} \right) \quad (6.2)$$

ここで， $\nabla \phi_C$ を格子 C に接する全ての面 f の ϕ_f を用い，グリーン・ガウスの定理に基づいて以下のように計算する方法を提案する．

$$\nabla \phi_C = \frac{1}{V_C} \sum_{f \in \partial V_C} s_f \phi_f \mathbf{n}_f \quad (6.3)$$

体積積分平均値 ϕ_C ，面積積分平均値 ϕ_f および計算された ϕ_u を用いることで，CIP-CSL 法に基づき一次元の二次補間関数を構築できる．いま $f \rightarrow C \rightarrow u$ 方向を N 軸方向とし，格子

C における ϕ に関する近似的な一次元補間関数を $\Phi_C(N)$ とおく． $\Phi_C(N)$ は $N = f$ において ϕ_f ， $N = u$ において ϕ_u をとり，さらに f から u までの間の $\Phi_C(N)$ の定積分を f から u までの距離で除した積分平均値として ϕ_C をとる．この 3 つの条件を用いて $\Phi_C(N)$ の二次補間関数 $aN^2 + bN + c$ の未知数 a ， b および c を決定することができる．ゆえに次の時刻の ϕ_f は， $\Phi_C(N)$ に対するセミラグランジェ法を用いて直接計算できる．

$$\phi_f^{(n+1)} = \Phi_C(f + u_f^N \Delta t) \quad (6.4)$$

$$u_f^N \equiv \mathbf{u}_f \cdot \mathbf{n}_f \quad (6.5)$$

本提案計算手法を，一次元の移流計算に戻って考察する． N 方向を X 軸方向に置き換えて u を f の上流側の既存点とし， f と u の間隔を Δx とおくと，式 (6.3) は

$$\nabla \phi_C = \frac{\partial \phi}{\partial x} = \frac{1}{\Delta x} (\phi_u - \phi_f) \quad (6.6)$$

すなわち格子中心点 C における格子点値 ϕ_f および ϕ_u を用いた一次元の中心差分に帰着することがわかる．これを式 (6.2) に代入すると，

$$\begin{aligned} & \phi_f + 2 \left(\nabla \phi_C \cdot \vec{fC} \right) \\ &= \phi_f + 2 \left(\frac{1}{\Delta x} (\phi_u - \phi_f) \times \frac{1}{2} \Delta x \right) \end{aligned} \quad (6.7)$$

すなわち上流側の格子点値 ϕ_u そのものである．したがって式 (6.4) は，二点 ϕ_f ， ϕ_u および挟まれた ϕ_C による一次元 CIP-CSL 法に帰着する．以上のことより本提案手法は，一次元 CIP-CSL 法の任意多面体非構造格子への自然な拡張となっている．

ϕ_f の更新のために，式 (6.4) をセミラグランジェ法で計算する方法を以下に提案する．面 f における移流速度 $\mathbf{u}$ を法線方向成分および接線方向成分に分解する．すなわち f において，移流方程式

$$\frac{\partial \phi_f}{\partial t} + \mathbf{u}_f \cdot \nabla \phi_f = 0 \quad (6.8)$$

を図 6.1 に示すように法線方向 N および接線方向 T に関する移流の二つの項に分解する．

$$\frac{\partial \phi_f}{\partial t} = -\mathbf{u}_f^N \cdot \nabla \phi_f - \mathbf{u}_f^T \cdot \nabla \phi_f \quad (6.9)$$

$$\mathbf{u}_f^N \equiv (\mathbf{u}_f \cdot \mathbf{n}_f) \mathbf{n}_f \quad (6.10)$$

$$\mathbf{u}_f^T \equiv \mathbf{u}_f - \mathbf{u}_f^N \quad (6.11)$$

式 (6.9) の右辺第一項は，先述した一次元 CIP-CSL 移流計算手法により計算できる．一方，



右辺第二項はグリーン・ガウスの定理を用いて以下のように簡易に計算することができる。

$$\mathbf{u}_f^T \cdot \nabla \phi_f \simeq \mathbf{u}_f^T \cdot \left(\frac{1}{V_C} \sum_{f' \in \partial V_C} s_{f'} \phi_{f'} \mathbf{n}_{f'} \right)_{C \rightarrow f} \quad (6.12)$$

また簡易な計算式 (6.12) のほかに、UTI-VSIAM3 の定式化を参考にして、以下の $\mathbf{u}_f^T \cdot \nabla \phi_f$ の計算手法も併せて提案しておく。

$$\mathbf{u}_f^T \cdot \nabla \phi_f \simeq (\nabla \cdot (\phi_f \mathbf{u}_f^T) - \phi_C \nabla \cdot \mathbf{u}_f^T)_{C \rightarrow f} \quad (6.13)$$

$$\simeq \left(\frac{1}{V_C} \sum_{f' \in \partial V_C} s_{f'} (\phi_{f'} - \phi_C) \mathbf{u}_{f'} \cdot \mathbf{n}'_{f'} \right)_{C \rightarrow f} \quad (6.14)$$

$$\mathbf{n}'_{f'} \equiv \mathbf{n}_{f'} - (\mathbf{n}_{f'} \cdot \mathbf{n}_f) \mathbf{n}_f \quad (6.15)$$

提案手法は、補間関数 $\Phi_C(N)$ に二次関数を用いた場合、面の法線方向を一次元方向とみなしたときに三次精度を有する。本手法を、非構造格子における VSIAM3 とよぶことにする。計算手法としては、直交格子における次元分割手法である VSIAM3 とは異なり、非次元分割手法の UTI(=Unsplit Time Integration) -VSIAM3 と同じ考え方、すなわち法線方向と接線方向に分解する定式化ではあるが、そもそも非構造格子において次元分割手法は成立しないため、単に VSIAM3 とよぶことにする。次節で、移流計算精度を TVD スキームと比較検証する。

6.1.2 移流計算精度のベンチマークテスト検証

二次元の均等な三角形格子を用い、3.2.2 節の Rudman ベンチマークテスト [82] を実施した。テスト条件を表 6.1 に示す。移流計算において速度 $\mathbf{u}$ は時間に対して一定とし、正方形の計算領域の中心点を回転軸として、反時計回りに剛体回転する速度場を設定した。格子数は、

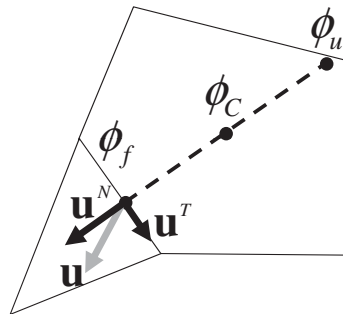


図 6.1 Normal and tangential velocity.

直交格子における Rudman のテストの 200×200 格子と同等の三角形格子数とした。Rudman ベンチマークテストでは、円に矩形の切り欠きが入った図形の内側を 1，外側を 0 と初期設定とし (図 6.2)，一回転後に元の形状がどの程度保持されているかを検証する。なお図形の初期設定は、各非構造格子について、切り欠き円を区分的直線に近似し、その直線が格子を切り取ったときの体積から流体率を厳密に計算し、VOF 値の初期値として与えている。しかしながら図 6.2 の下の図を見ると明らかとなおり、コンターの線はがたつきがあることが分かる。これは、計算結果の描画処理において、各格子単位でコンター線は格子中心の VOF 値から格子頂点への簡易な線形補間により描かれているためと推測される。したがってコンター図の格子単位のがたつきは、計算結果を正しく表していない。非構造格子の描画処理にはこのような描画誤差の問題があることに留意する必要がある。

TVD スキームと VSIAM3 との比較結果を図 6.3 に示す。TVD スキームでは広く用いられている Van Leer リミッタを用い、時間積分は TVD-RK3 を用いた。VSIAM3 では二次補間関数の CSL2 の他に有理関数の CSLR を用いたケースも示す。図 6.3 から明らかとなおり、CSL2 および CSLR とともに TVD スキームよりも優れた結果が得られた。CSL2 は TVD と比較して数値拡散は大幅に小さく、また切り欠き部の形状もより初期形状を忠実に再現している。CSLR は CSL2 よりも TVD に近い結果ではあるが、数値拡散は TVD よりも少ないことは明らかである。また形状保存性も、切り欠きの頂点の部分の丸みの度合いが TVD よりも小さいことが分かる。また表 6.2 に各々の結果における L_1 誤差を示す。 L_1 誤差は以下の式で計算される。

$$L_1 = \frac{\sum_C |F_C - F_C^{init}|}{\sum_C |F_C^{init}|} \quad (6.16)$$

L_1 誤差は、 $CSL2 < CSLR < TVD$ の順となっており、図 6.3 の結果と符合する。したがって ϕ_f を加えたマルチモーメント法を採用することにより、比較的簡易な計算手法および TVD スキーム同様のコンパクトな格子ステンシルを用いても、より高い精度を得られることが示された。

表 6.1 Setting of Rudman benchmark test.

Cells (N_C)	39,762
Faces (N_f)	139,449
Vertices (N_v)	40,328
Courant number	0.1
Calculation steps for one revolution	4,430



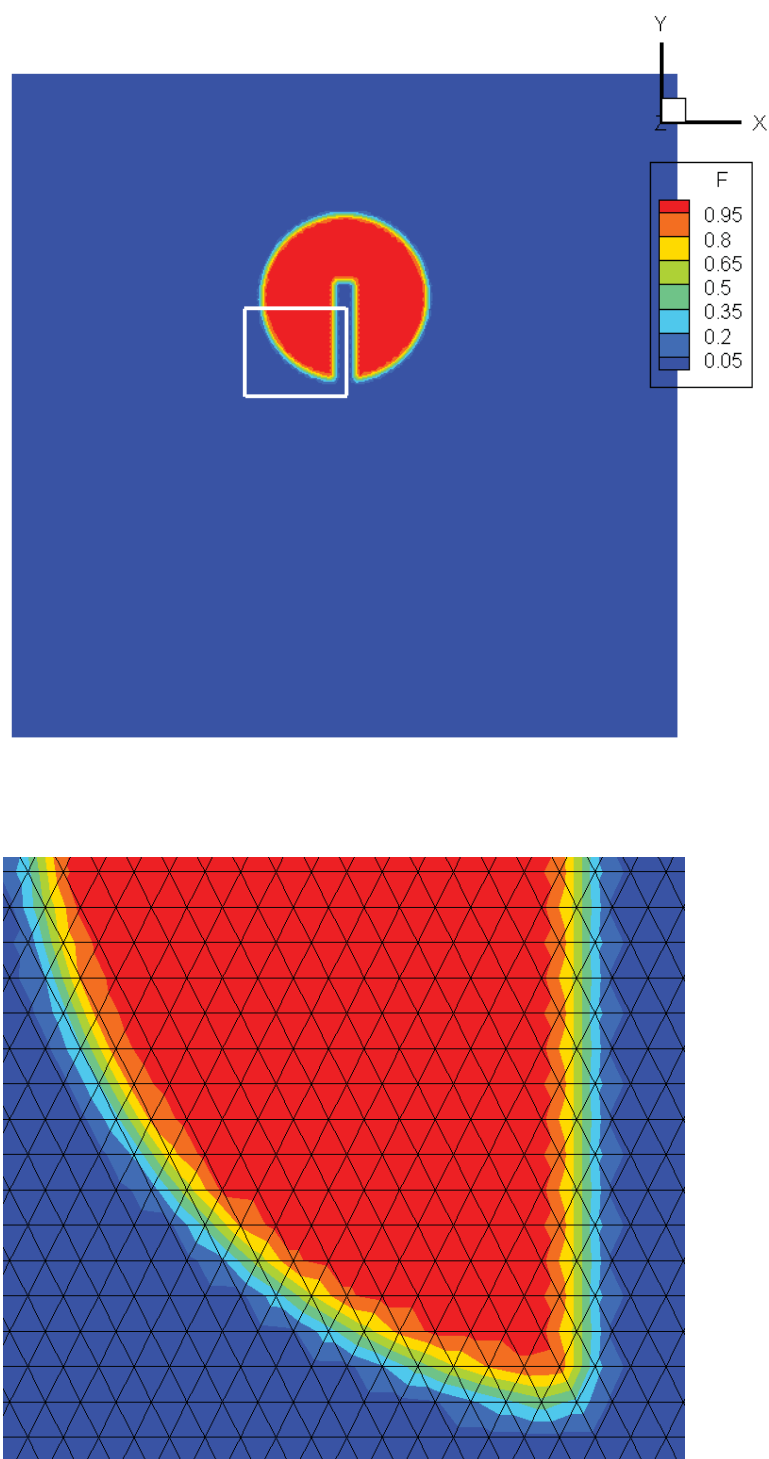
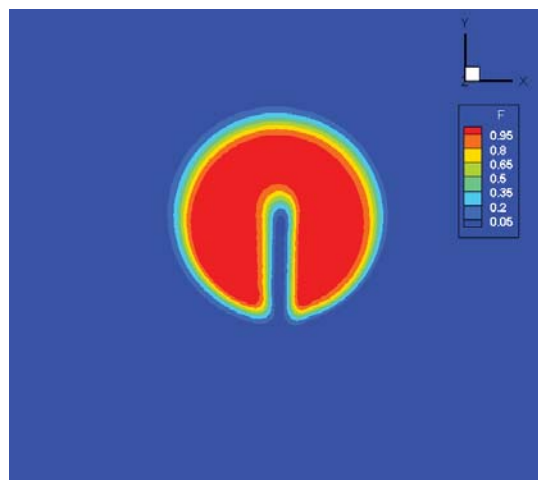


図 6.2 Initial profile of Rudman benchmark test.



TVD (Van Leer limiter)



VSIAM3 (CSL2, semi-Lagrangian)



VSIAM3 (CSLR, semi-Lagrangian)

図 6.3 The results of TVD and VSIAM3s after one revolution.

表 6.2 Comparison of L_1 error between TVD and VSIAM3.

TVD (Van Leer)	VSIAM3 (CSL2)	VSIAM3 (CSLR)
0.190	0.144	0.167

6.1.3 計算量の評価

前節までの評価により，非構造格子 VSIAM3 では TVD スキームを上回る移流計算精度が得られることを示したが，一方 VSIAM3 はマルチモーメント法であり，TVD スキームでは持たないセル境界値を持つため，自由度の上では二次元で約 3 倍，また三次元で約 4 倍となる．もし VSIAM3 では計算量が TVD スキームの 4 倍となるならば，TVD スキームで 4 倍のメッシュ数のモデルで計算した方が，VSIAM3 よりも計算精度が高くなると考えられる．本節では，TVD スキームと VSIAM3 の計算量について比較する．計算量はコードの実装方法で変わるため，ここではあくまでも概略の傾向把握を目的とする．したがって実装においてはキャッシュブロッキングなどの複雑もしくは特殊なチューニングは一切行わない．また計算モデルは前節の Rudman ベンチマークモデルをそのまま使用する．VSIAM3 ではセミラグランジェ手法で二次補間関数を用いる．

表 6.3 および表 6.4 に TVD スキームおよび VSIAM3 における総演算数および計算時間を示す．計算時間は gcc において最適化なし (-O0) および一般的な最適化 (-O3) の二種類のオプションでコンパイルした実行バイナリにて計測した．使用システムは Intel Xeon E5-1620 3.6GHz であり，マルチスレッドなどの並列計算処理はしていない．表 6.3 および表 6.4 から，-O3 最適化により 3~4 倍速度が向上していることが分かる．TVD スキームと VSIAM3 との比較では，算術演算量で 1.14 倍，移動演算量で 1.51~1.75 倍となっており，VSIAM3 の方が多いものの 2 倍未満にとどまっている．計算時間では -O0 では VSIAM3 の方がわずかに短い結果となっているが，演算量の比較と併せて考慮するとこの結果は妥当とはいえない．一方 -O3 では TVD スキームに対して VSIAM3 は 1.24 倍となっており，算術演算比 1.14 と移動演算比 1.51~1.75 の中間に位置するため妥当と考えられる．以上の結果から，VSIAM3 は TVD スキームと比較して計算量が多いものの，数十 % の増加にとどまると推測される．

今回の評価は 1 ループでのものであり，実際の計算では，例えば TVD スキームで時間積分を TVD-RK2 にした場合 2 回のループ，TVD-RK3 にした場合 3 回のループが必要となるため，計算時間は各々 2 倍，3 倍となる．一方 VSIAM3 ではセミラグランジェであり 1 回のループの計算であるため，総合的に判断すると VSIAM3 は TVD スキームよりも計算量が多いとはいえないことが分かる．自由度数で見ると 4 倍となるため意外な結果であるが，非構造格子の VSIAM3 では最小二乗法の計算をすることなく SIA の値を活用してグリーン・ガウスの定理により少ない計算量で勾配値を計算できること，また一般的に有限体積法ではフラックス計算としてセル界面値を計算するため，その値をそのままモーメントとして保持することは計算量の大きな増加にはつながらないことが VSIAM3 の大きな利点と考えられる．

表 6.3 Total number of double precision floating operations on TVD code

No.	Operations			Elapse time [ms]	
	Arithmetic	Move (all)	Move (cached)	-O0	-O3
1.	7,236,684	1,073,574	874,764	5.45	1.83
2.	4,322,919	1,952,286	1,394,490	5.72	1.36
3.	914,526	675,954	675,954	1.51	0.45
Total	12,474,129	3,701,814	2,945,208	12.7	3.64

表 6.4 Total number of double precision floating operations on VSIAM3 code

No.	Operations			Elapse time [ms]	
	Arithmetic	Move (all)	Move (cached)	-O0	-O3
1.	1,510,956	1,153,098	1,153,098	2.43	0.64
2.	11,016,471	3,067,878	2,649,531	7.42	3.05
3.	1,709,766	1,073,574	1,073,574	2.07	0.75
4.	0	278,898	278,898	0.31	0.08
Total	14,237,193	5,573,448	5,155,101	12.2	4.52
vs. TVD	1.14	1.51	1.75	0.96	1.24



6.2 VSIAM3 を用いた界面捕獲手法の多面体格子への拡張

前節の VSIAM3 は、任意多面体非構造格子における Navier-Stokes 方程式の移流項の計算に用いられるが、直交格子のときと同様に気液二相流計算における気液界面捕獲計算にも適用する。

6.2.1 STAA 法のためのレベルセット関数構築

STAA 法において、数値拡散補正されるべき格子の判定および人工的逆拡散フラックスの方向ベクトルを計算するために、レベルセット関数 [16] を計算する必要がある。レベルセット関数は気液界面からの符号付き距離関数であり、気相で負、液相で正の値をとり、気液界面上では 0 である。以下に任意多面体非構造格子におけるレベルセット関数計算方法を示す。

1. 各格子にレベルセット値の初期値を設定する。
2. レベルセット値が収束するまで、以下のステップを繰り返す。
 - (a) レベルセット関数の勾配を計算する。
 - (b) 気液界面方向に向かう法線ベクトルを計算する。
 - (c) 計算した法線ベクトルに沿った出発点の座標値を計算する。
 - (d) 最小二乗法により、出発点におけるレベルセット値を補間計算する。
 - (e) 出発点の値より、セミラグランジェ法を用いてレベルセット値を更新する。

ある格子 C におけるレベルセット値の初期値は以下の式で計算される

$$G_C^{(0)} \equiv 2.0\alpha (F_C - 0.5). \quad (6.17)$$

パラメータ α の値として $2.0 \times \Delta$ が用いられる。 Δ はメッシュ間隔の基準値である。

いま、あるレベルセット反復計算におけるサイクルを l とおく。格子 C のレベルセットの勾配値 ∇G_C は、その格子の頂点を一つ以上共有する全ての隣接格子のレベルセット値を用いて最小二乗法により計算する。 ∇G_C を用いて、セミラグランジェにおける出発点を決定するための法線ベクトルは以下のように計算される。

$$\mathbf{n}_C = -\text{SGN}(G_C^{(l)}) \frac{\nabla G_C^{(l)}}{|\nabla G_C^{(l)}|}, \quad (6.18)$$

$\text{SGN}(a)$ は a の符号関数を表し、 a が負のときに -1 、正のときに 1 をとる。したがって $\mathbf{n}_C$ は常に気液界面方向を示している。そして気液界面方向の出発点は以下のように計算される。

$$\mathbf{x}_d = \mathbf{x}_C + d_C \mathbf{n}_C \quad (6.19)$$

$$d_C \equiv \text{CFL} \times \Delta \quad (6.20)$$

CFL は 0.5 を用いる.

出発点におけるレベルセットの補間計算は, レベルセット計算を精度よく安定的に行うために最も重要なことである. ここでは, 以下のような最小二乗法を提案する. まず補間関数は以下の線形補間関数を用いる.

$$G_C^{(*)}(\mathbf{x}) = a(x - x_C) + b(y - y_C) + c(z - z_C) + G_C^{(l)} \quad (6.21)$$

未知係数 a , b および c の計算方法として, 上流側すなわち気液界面に近い側の隣接点のみを利用して最小二乗法計算を行うことで, 上流化により計算の安定化を図る. すなわち,

$$(\mathbf{x}_{nb} - \mathbf{x}_C) \cdot (\mathbf{x}_d - \mathbf{x}_C) \geq 0 \quad (6.22)$$

を満たす隣接格子のみを選択して, 最小二乗法を計算する. ここで $\mathbf{x}_{nb}$ は最小二乗法計算で選択される隣接格子の座標値を表す.

レベルセット値は以下のように更新される.

$$G_C^{(l+1)} = G_C^{(*)}(\mathbf{x}_d) + \text{SGN}(G_C^{(l)})d_C. \quad (6.23)$$

ただし出発点が気液界面を超えて参照される場合, すなわち

$$G_C^{(l)} G_C^{(*)}(\mathbf{x}_d) \leq 0, \quad (6.24)$$

の場合, CFL は局所的に以下のように修正される.

$$d_C \leftarrow d_C \frac{G_C^{(l)}}{G_C^{(l)} - G_C^{(*)}(\mathbf{x}_d)} \quad (6.25)$$

d_C が修正された後, 式 (6.19), (6.23) は再計算される. 式 (6.25) から明らかなように, $G_C \rightarrow 0$ すなわち C が気液界面上にある場合 $d_C \rightarrow 0$ となり, $\mathbf{x}_d \rightarrow \mathbf{x}_C$ となる. したがってこの場合式 (6.23) より $G_C^{(l+1)} \rightarrow G_C^{(l)}$ が得られ, 気液界面が動かないことを陰的に示している. 以上のステップを式 (6.23) の G_C が収束するまで繰り返し実行する. 実際の計算では, 気液界面から 5~10 格子分程度の距離以内の格子の G_C が収束するまでで実用上十分である.

6.2.2 多面体格子における STAA 法の実装

任意多面体非構造格子における STAA 法のアルゴリズムを以下に提案する.

1. VSIAM3 を用いて, 流体率 F の移流計算を行う.
2. F よりレベルセット関数 G を計算する.
3. G より計算された逆数値拡散フラックスを用いて F の数値拡散を修正する.



STAA 法によって修正される数値拡散量 ΔF は次式で定義される。

$$\Delta F_C = \begin{cases} \omega F_C & \text{if } G_C < 0.0 \\ \omega (F_C - 1.0) & \text{if } G_C > 0.0 \end{cases}, \quad (6.26)$$

ここで

$$|G_C| > \varepsilon \Delta. \quad (6.27)$$

である。 ω は安定化のための不足緩和係数であり、 $0 < \omega \leq 1$ である。 本論文では $\omega = 0.5$ とする。 また ε は補正対象とする格子の気液界面からの距離のしきい値であり、 通常は $1.0 \leq \varepsilon \leq 2.0$ である。 今回は $\varepsilon = 1.0$ を用いる。

各隣接格子に対する単位方向ベクトルは次式で定義される。

$$\mathbf{v}_{nb} \equiv \frac{\mathbf{x}_{nb} - \mathbf{x}_C}{|\mathbf{x}_{nb} - \mathbf{x}_C|}, \quad (6.28)$$

ここで $\mathbf{n}_C$ から $\mathbf{v}_{nb}$ への角度を θ_{nb} とする。

$$\theta_{nb} \equiv \angle(\mathbf{n}_C, \mathbf{v}_{nb}). \quad (6.29)$$

二次元計算では、 図 6.4 に示すように隣接格子 $nb1$ および $nb2$ の二格子が選ばれる。 $nb1$ は $\theta_{nb1} = \max_{nb}(\theta_{nb})$ であり、 一方 $nb2$ は $\theta_{nb2} = \min_{nb}(\theta_{nb})$ である。

$nb1$ および $nb2$ の方向ベクトルを各々 $\mathbf{v}_{nb1}$ および $\mathbf{v}_{nb2}$ とする。 $\mathbf{n}_C$ と $\mathbf{v}_{nb1}$, $\mathbf{v}_{nb2}$ との関係は、 重み係数 w_1 , w_2 を用いて以下のように表せる。

$$w_1 \mathbf{v}_{nb1} + w_2 \mathbf{v}_{nb2} = \mathbf{n}_C. \quad (6.30)$$

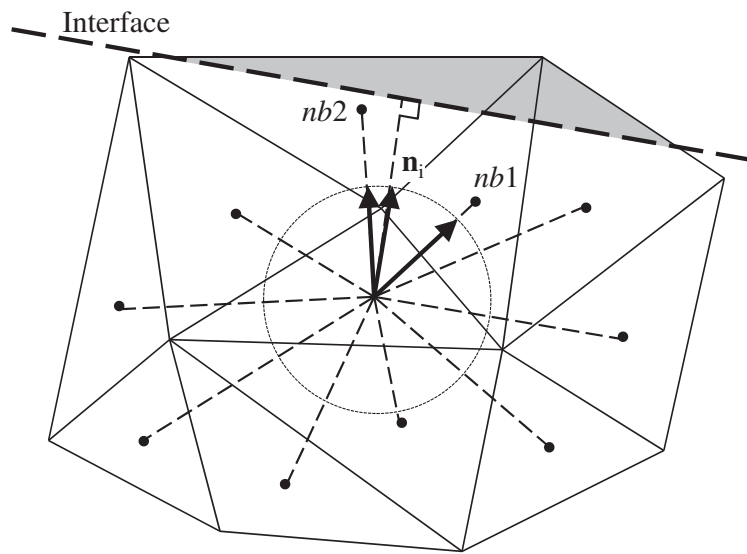


図 6.4 Anti-diffusion vector for STAA in 2D.

w_1 および w_2 は以下の連立方程式を解いて求められる。

$$\begin{bmatrix} v_{x,nb1} & v_{x,nb2} \\ v_{y,nb1} & v_{y,nb2} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} n_{x,C} \\ n_{y,C} \end{bmatrix}. \quad (6.31)$$

その結果、数値拡散 ΔF_C は隣接格子 $nb1$, $nb2$ に次式に基づいて分配される。これを人工的逆数値拡散フラックス (Artificial Anti-diffusion flux) とよぶ。

$$\Delta F_{C \rightarrow nb1} = \frac{w_1}{w_1 + w_2} \Delta F_C \quad (6.32)$$

$$\Delta F_{C \rightarrow nb2} = \frac{w_2}{w_1 + w_2} \Delta F_C. \quad (6.33)$$

三次元計算では、図 6.5 に示すように 3 つの隣接格子の単位方向ベクトル $\mathbf{v}_{nb1}$, $\mathbf{v}_{nb2}$ および $\mathbf{v}_{nb3}$ が選択される。 $\mathbf{n}_i$, $\mathbf{v}_{nb1}$, $\mathbf{v}_{nb2}$ および $\mathbf{v}_{nb3}$ の関係は、二次元同様、重み係数 w_1 , w_2 および w_3 を用い以下のように表せる。

$$w_1 \mathbf{v}_{nb1} + w_2 \mathbf{v}_{nb2} + w_3 \mathbf{v}_{nb3} = \mathbf{n}_C. \quad (6.34)$$

w_1 , w_2 および w_3 は以下の連立方程式を解くことにより求めることができる。

$$\begin{bmatrix} v_{x,nb1} & v_{x,nb2} & v_{x,nb3} \\ v_{y,nb1} & v_{y,nb2} & v_{y,nb3} \\ v_{z,nb1} & v_{z,nb2} & v_{z,nb3} \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix} = \begin{bmatrix} n_{x,C} \\ n_{y,C} \\ n_{z,C} \end{bmatrix}. \quad (6.35)$$

最終的に、数値拡散 ΔF_i は次式に基づいて隣接格子 $nb1$, $nb2$ および $nb3$ に人工的逆数値拡散フラックスとして分配される。

$$\Delta F_{C \rightarrow nb1} = \frac{w_1}{w_1 + w_2 + w_3} \Delta F_C \quad (6.36)$$

$$\Delta F_{C \rightarrow nb2} = \frac{w_2}{w_1 + w_2 + w_3} \Delta F_C \quad (6.37)$$

$$\Delta F_{C \rightarrow nb3} = \frac{w_3}{w_1 + w_2 + w_3} \Delta F_C. \quad (6.38)$$

三次元計算における隣接格子 $nb1$, $nb2$ および $nb3$ の選択方法は種々考えられるが、ここでは以下に示すアルゴリズムを提案しておく。

1. 図 6.6 に示すように法線ベクトル $\mathbf{n}_C$ の終端点を p とおく。
2. $\mathbf{n}_C$ を法線とし、端点 p を含む平面を平面 l とおく。
3. 全ての隣接格子点の中から、 $\mathbf{n}_C$ と $\mathbf{v}_{nb}$ のなす角度 $\theta_{nb} = \cos^{-1}(\mathbf{n}_C \cdot \mathbf{v}_{nb})$ が最も小さい格子点を $nb1$ とする。
4. 平面 l と $\mathbf{v}_{nb1}$ の交点を p_1 とおく。
5. 以下の条件を満たす $nb2$ および $nb3$ を選択する。



- 平面 l と $\mathbf{v}_{nb2}$ および $\mathbf{v}_{nb3}$ との交点 p_2, p_3 について以下を満たす.

$$\mathbf{s}_{12} \cdot \mathbf{n}_C > 0 \quad (6.39)$$

$$\mathbf{s}_{23} \cdot \mathbf{n}_C > 0 \quad (6.40)$$

$$\mathbf{s}_{31} \cdot \mathbf{n}_C > 0 \quad (6.41)$$

$$\mathbf{s}_{mn} \equiv \overrightarrow{p \cdot p_m} \times \overrightarrow{p \cdot p_n} \quad (6.42)$$

- 下式で計算される三角形 $p_1 p_2 p_3$ の面積が, 全ての三角形の中で最も小さい.

$$s_{p_1 p_2 p_3} = \frac{|\mathbf{s}_{12}| + |\mathbf{s}_{23}| + |\mathbf{s}_{31}|}{2} \quad (6.43)$$

次節に, 気液界面捕獲精度の検証を行う.

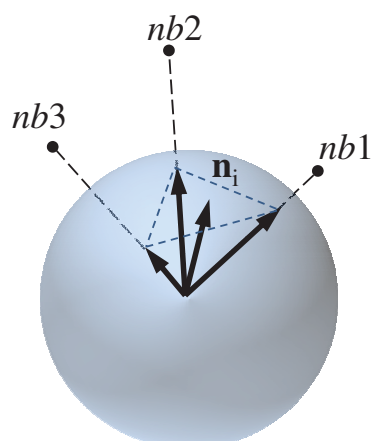


図 6.5 Anti-diffusion vector for STAA in 3D.

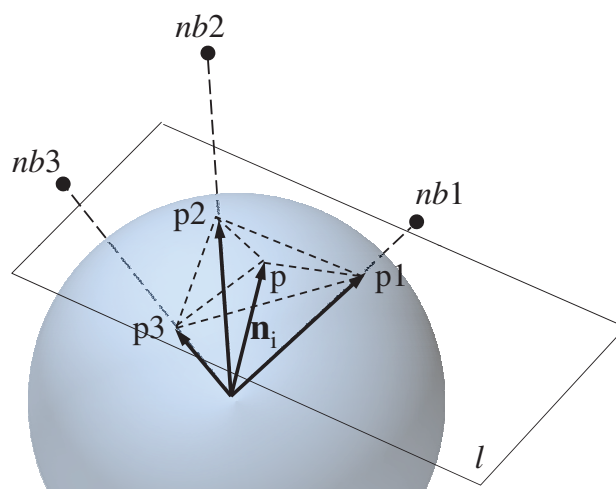


図 6.6 Searching three neighboring points for 3D-STAA.



6.2.3 気液界面捕獲精度のベンチマーク検証

VSIAM3 の移流計算精度検証のときと同様の, Rudman ベンチマークテストを実施した. 移流速度分布, 格子数および初期設定は表 6.1 と同じである. 精度検証の比較対象として, 第 2 章で述べたように今日広く用いられている代表的な幾何的 VOF 法として PLIC 法, また代数的 VOF 法として CICSAM および MULES を選定した.

PLIC の気液界面再構築において, 任意多面体非構造格子では Rider らによる Brent 法繰り返し計算手法を非構造格子に適用した.

図 6.8 に, 計算結果の他の手法との比較を示す. 他の手法の計算結果はどれも信頼性が高く大きな差はないが, 切り欠き部の変形具合でみると PLIC 法が若干良い結果と考えられる. いずれの手法も切り欠き部の角が丸くなっている結果となっており, ここではそのようすを「数値表面張力」とよぶことにする. しかし VSIAM3+STAA 法は, 切り欠き部の直線度や幅, さらに角部の丸みの小ささで PLIC 法より良い結果であり, 数値表面張力が他の手法よりも小さい結果といえる.

表 6.5 に L_1 誤差の比較を示す. まず Eval.1 では一回転後の F をそのまま用いて誤差評価したところ, VSIAM3 +STAA 法は PLIC よりは誤差値が大きいものの, 他の代数的 VOF

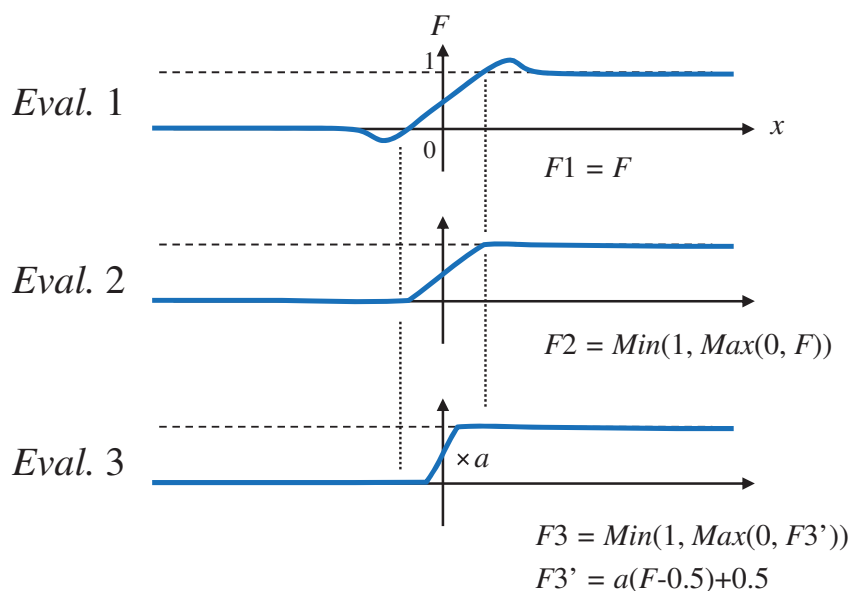


図 6.7 Evaluation method of L1 error.

法より優れる結果となった．そこで Eval.2 では図 6.7 に示すように，

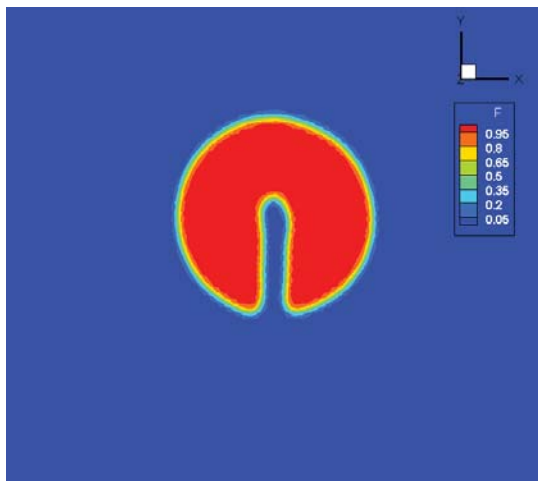
$$F_2 \equiv \min(\max(F, 0.0), 1.0) \quad (6.44)$$

すなわち計算結果の F に対しオーバーシュートおよびアンダーシュートを排除して L_1 誤差を評価したところ，MULES の結果がわずかに VSIAM3+STAA を上回ることが分かった．したがって MULES ではオーバーシュートもしくはアンダーシュートが多く発生し， L_1 誤差に大きな影響をおよぼしていることが分かる．次に Eval.3 では図 6.7 に示すように，

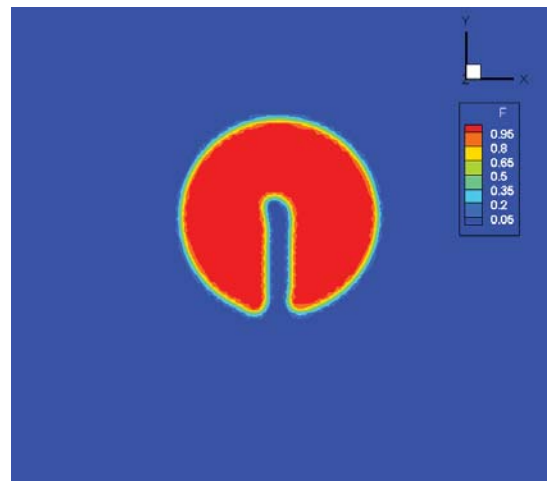
$$F_3 \equiv \min(\max(a(F - 0.5) + 0.5, 0.0), 1.0) \quad (6.45)$$

すなわち気液界面位置である $F = 0.5$ を動かさないように， F の勾配を大きくして気液界面幅を圧縮していくと L_1 誤差がどのように変化するかを確認した．この評価は，各計算手法により異なる気液界面幅が L_1 誤差におよぼす影響を排除することを意図している．その結果 VSIAM3+STAA 法では $a = 2.0$ とすることで大幅に L_1 誤差が低減し，他のスキームより大きく優れることが分かった．また $a = 1.5$ の場合に MULES において最も L_1 誤差が低減しているが，この場合においても VSIAM3+STAA 法は MULES および他の手法とも大幅に小さい結果となった．この結果は図 6.8 において VSIAM3+STAA 法の方が $F = 0.5$ の気液界面位置の精度が高いことと符合する．表中の下線の値は，各々のスキームにおいてもっとも L_1 誤差が小さいものを示している．この比較においても，STAA は代数的 VOF 法だけでなく幾何的 VOF 法の PLIC と比較しても良い結果が得られている．VSIAM3+STAA の Eval.1 および 2 の結果が良くない原因としては，STAA 法ではある程度の気液界面幅を設定してその外側のみの数値拡散を補正するため，気液界面近傍の F が他の手法よりも拡散的になっているためと推測される．STAA 法において，気液界面幅を拡大することなく保持できれば流体計算上は問題ないと考えられる．

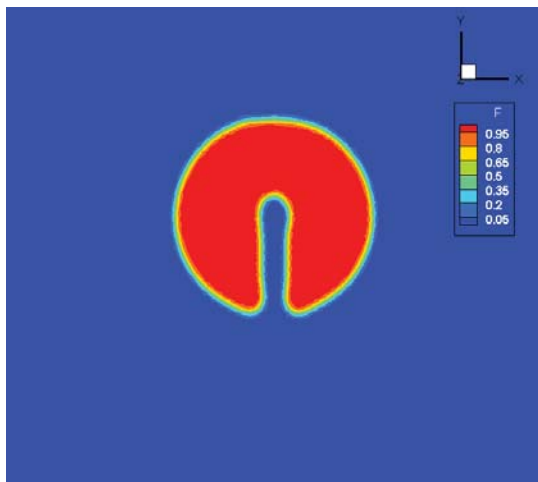




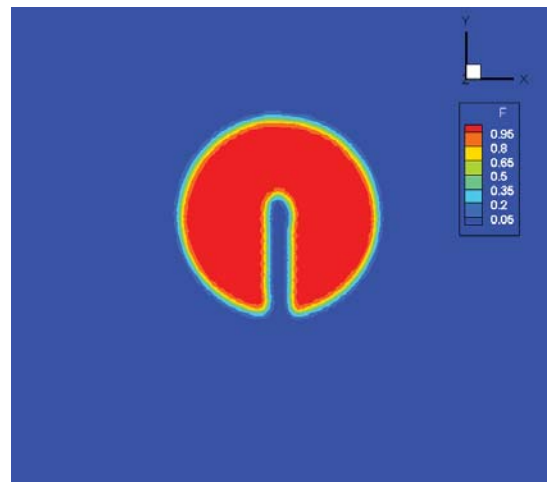
CICSAM



PLIC



MULES



STAA (VSIAM3, CSL2, semi-Lagrangian)

図 6.8 The results of STAA and other existing methods after one revolution.

表 6.5 Comparison of L_1 error among STAA and various existing methods.

		CICSAM	PLIC	MULES	STAA (VSIAM3, CSL2, semi-Lag.)
Eval.1		4.05×10^{-2}	1.93×10^{-2}	1.02×10^{-1}	3.74×10^{-2}
Eval.2		4.05×10^{-2}	1.60×10^{-2}	3.26×10^{-2}	3.69×10^{-2}
Eval.3	a=1.2	3.43×10^{-2}	1.69×10^{-2}	2.77×10^{-2}	2.40×10^{-2}
	a=1.5	3.27×10^{-2}	2.05×10^{-2}	2.67×10^{-2}	1.47×10^{-2}
	a=2.0	3.24×10^{-2}	2.42×10^{-2}	2.72×10^{-2}	1.25×10^{-2}
	a=2.5	3.28×10^{-2}	2.62×10^{-2}	2.79×10^{-2}	1.48×10^{-2}

6.3 フラクショナルステップ法による気液二相流非圧縮性流体解析手法

非保存型表示の Navier-Stokes 方程式

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho} \nabla P + \frac{1}{\rho} \nabla \cdot (\mu \nabla \mathbf{u}) + \frac{1}{\rho} \mathbf{F}_V \quad (6.46)$$

を非構造格子気液二相流において解く方法について考察する． $\mathbf{u}$, ρ および P は各々流速，密度および圧力を表し， μ は粘性係数，また $\mathbf{F}_V$ は表面張力および重力などの体積力を表す．この式を以下のようにフラクショナルステップ法 [87][129] により計算する．

$$\frac{\partial \mathbf{u}}{\partial t} = -\mathbf{u} \cdot \nabla \mathbf{u} \quad (6.47)$$

$$\frac{\partial \mathbf{u}}{\partial t} = \frac{1}{\rho} \nabla \cdot (\mu \nabla \mathbf{u}) + \frac{1}{\rho} \mathbf{F}_V \quad (6.48)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho} \nabla P \quad (6.49)$$

式 (6.47) の移流方程式は，以下のように置き換えることで VSIAM3 により解くことができる．

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}) = \phi \nabla \cdot \mathbf{u} \quad (6.50)$$

$$\phi \equiv \begin{pmatrix} u \\ v \\ w \end{pmatrix} \quad (6.51)$$

非圧縮性流体を仮定する場合は，式 (6.50) の右辺は 0 である． $\nabla \cdot \mathbf{u} \neq 0$ の場合は，

$$\phi_C^{n+1} = \phi_C^n - \frac{\Delta t}{V_C} \sum_{f \in \partial V_C} s_f \mathbf{u}_f \cdot \mathbf{n}_f (F_f - \phi_C^n) \quad (6.52)$$

により計算できる．また流体率 F について移流方程式を VSIAM3 および STAA 法により解き，

$$\rho = \rho_{gas} \cdot (1 - F) + \rho_{liq} \cdot F \quad (6.53)$$

$$\mu = \mu_{gas} \cdot (1 - F) + \mu_{liq} \cdot F \quad (6.54)$$

により密度および粘性係数を計算できる．本手法では， u_C , v_C , w_C および F_C が全て格子中心で定義されるコロケート配置の有限体積法を用いる．

非移流項 (6.48) は空間について二次中心差分法，時間についてオイラー一次前進差分法を用いて計算する．また圧力項 (6.49) は，非圧縮性を仮定する場合は両辺に発散をとって



$\nabla \cdot \mathbf{u}^{n+1} = 0$ に代入することにより，ポアソン方程式

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla P^{n+1} \right) = \nabla \cdot \mathbf{u}^* \quad (6.55)$$

が得られる． $\mathbf{u}^*$ は，フラクショナルステップにおいて非移流項 (6.48) を解いた直後の流速 $\mathbf{u}$ を表す．

また $\nabla \cdot \mathbf{u} \neq 0$ を考慮する場合は，非保存型表示のエネルギー方程式

$$\frac{\partial P}{\partial t} + \mathbf{u} \cdot \nabla P = -\rho C_s^2 \nabla \cdot \mathbf{u} \quad (6.56)$$

をフラクショナルステップ分割した

$$\frac{\partial P}{\partial t} = -\mathbf{u} \cdot \nabla P \quad (6.57)$$

$$\frac{\partial P}{\partial t} = -\rho C_s^2 \nabla \cdot \mathbf{u} \quad (6.58)$$

について，式 (6.57) は VSIAM3 による移流計算，また式 (6.58) は式 (6.49) を代入して得られるポアソン方程式

$$\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla P^{n+1} \right) = \frac{P^{n+1} - P^*}{\rho C_s^2 \Delta t} + \nabla \cdot \mathbf{u}^* \quad (6.59)$$

を解く． P^* は，フラクショナルステップにおいて非移流項 (6.57) を解いた直後の圧力 P を表す．この手法は CIP-Combined Unified Procedure (CIP-CUP) 法 [130] とよばれる．非圧縮性のポアソン方程式 (6.55) と比較して，式 (6.59) の右辺第一項が連立方程式行列の優対角化の効果を生むため，圧力方程式解法の収束が早くなり，計算時間の短縮につながる．

流速の発散項 $\nabla \cdot \mathbf{u}^*$ は，以下の式により計算される．

$$(\nabla \cdot \mathbf{u}^*)_C = \frac{1}{V_C} \sum_{f \in \partial V_C} s_f \mathbf{u}_f^* \cdot \mathbf{n}_f \quad (6.60)$$

ここで $\mathbf{u}_f^*$ は計算の安定性を考慮し [131][38]， $\mathbf{u}_C^*$ の線形補間とする．

$$\mathbf{u}_f^* \equiv \frac{1}{2} (\mathbf{u}_C^* + \mathbf{u}_{nb}^*) \quad (6.61)$$

また圧力のラプラシアン項は，ガウスの発散定理を用いて以下のように表す [132]．

$$\left(\nabla \cdot \left(\frac{\Delta t}{\rho} \nabla P^{n+1} \right) \right)_C = \frac{\Delta t}{V_C} \sum_{f \in \partial V_C} \frac{s_f}{\rho_f} (\nabla P^{n+1})_f \cdot \mathbf{n}_f \quad (6.62)$$

式 (6.62) の面 f における圧力勾配 $(\nabla P^{n+1})_f \cdot \mathbf{n}_f$ は以下のように表す [132].

$$(\nabla P^{n+1})_f \cdot \mathbf{n}_f \simeq (\nabla P^{n+1})_f \cdot \mathbf{e}_f + (\nabla P^*)_f \cdot (\mathbf{n}_f - \mathbf{e}_f) \quad (6.63)$$

$$\simeq \frac{P_{nb}^{n+1} - P_C^{n+1}}{|\mathbf{x}_{nb} - \mathbf{x}_C|} + (\nabla P^*)_f \cdot (\mathbf{n}_f - \mathbf{e}_f) \quad (6.64)$$

$$\mathbf{e}_f \equiv \frac{\mathbf{x}_{nb} - \mathbf{x}_C}{|\mathbf{x}_{nb} - \mathbf{x}_C|} \quad (6.65)$$

圧力のポアソン方程式を解いた後は、式 (6.49) により流速を修正する.

6.4 GPGPU および並列処理による高速化

気液二相流解析では、多数の気泡や液滴の運動を正確に計算するためには、非常に多くの計算格子が必要となる. 我々は Graphic Processor Unit (GPU) アクセラレータを用いた高速計算と、Message Passing Interface (MPI) によるメモリ分散型並列処理の併用により、大規模格子計算に対応する任意多面体非構造格子気液二相流ソルバーを開発した. このソルバーでは移流計算のみならず圧力求解を含む全てのフラクショナルステップ計算において GPGPU 化されている. 提案手法は簡素なベクトル計算の集合体であり複雑な条件分岐処理は必要としないため、GPGPU 化は容易であることが利点である. このコードは、NVIDIA 社の GPU デバイスに対応する”CUDA-C”により開発された. VSIAM3 および STAA による二相流ソルバーは、for ループ 1 つの関数モジュールの集合体で全ての定式化をコーディング可能であり、GPGPU による加速に適合する.

表 6.6 に各関数の GPU と CPU の計算時間の比較を示す. このテストで使用された計算モデルは約 145 万格子の衛生陶器製品モデルである. 表 6.6 より、圧力のポアソン方程式の求解の加速率が 6.76 倍と低いものの、それ以外の計算では 11~22 倍の加速率を得ている. 非構造

表 6.6 Comparison of performance between GPU and CPU calculations. GPU calculation was conducted by one NVIDIA GeForce GTX 690 device, while CPU calculation was conducted by one Intel Xeon 3.6 GHz core with the compile command “nvcc -O3 -arch=sm_20”. The numbers of cells, faces and nodes are 1,447,313, 4,390,325, 1,496,242, respectively.

	Functions [second / time step]			
	advection	level set	STAA	pressure
GPU	0.22	0.80	0.28	5.71
CPU	2.43	17.9	3.12	38.6
Rate	11.0x	22.4x	11.1x	6.76x



格子特有の不連続メモリアクセスに対するコード最適化は行っていないが、提案手法の GPU 計算への適用性は良好といえる。

本手法の簡潔さは、MPI 並列化の実装の容易性にも寄与する。VSIAM3 および STAA 法はコンパクトなステンシルの手法であり、一つの格子の計算に必要な近接格子のステンシル幅は 1 である。したがって並列計算における分割領域境界の通信は 1 格子幅分で済むため、通信量は例えば WENO 等のワイドステンシル手法と比較して相対的に少ないことがスケーラビリティに対する利点である。

領域分割方法として、第 4 章で提案した直交格子における「境界不整合領域分割法」を非構造格子に拡張する。すなわち以下の手順で三次元領域分割を行う。

1. X, Y および Z 軸方向に分割された仮想の直交格子=バケットを用意する。
2. 各非構造格子を最も近いバケットに振り分ける。
3. 分割された各領域の格子数が最も均等に近くなるように Z 軸方向についてバケットを M_z 個に分割する。
4. M_z 個に分割された各領域で、Y 軸方向についてバケットを M_y 個に分割する。
5. $M_y \times M_z$ 個に分割された各領域で、X 軸方向についてバケットを M_x 個に分割する。

領域間の通信は MPI と GPU カーネル演算を並行して行うオーバーラップ通信をプログラム全体にわたって適用する。通信の隠蔽化により、並列化効率の向上を図る。

図 6.9 に、非圧縮性流体計算において通常最も計算量が多い圧力ポアソン方程式について、CUDA-C および MPI で加速した PCG 求解のベンチマーク結果を示す。用いたデータは、格子数 2,279,568 の衛生陶器製品モデルである。システム 1 は 1 ノードに 2GPU を搭載し、InfiniBand QDRx4 でネットワーク接続した標準的なクラスタシステムであり、システム 2 は 1 ノードに 8GPU を搭載したファットノードシステムである。本ベンチマークでは 228 万メッシュと比較的小規模なモデルを利用しているが、システム 1 では 8GPU までのスケーラビリティはほぼ理想値が得られており、提案手法は妥当といえる。一方システム 2 では、スケーラビリティは 8GPU で約 5.8 倍とシステム 1 よりも低いが、計算格子数が少ないことを考慮すると実用レベルと考えられる。システム 2 のスケーラビリティが低い理由として、8GPU の各プロセスが領域間通信する際に、サーバーのリングバスや PCI-Express 3.0 バスなどの内部バスに一斉にアクセスすることによる競合が原因と推測される。本 8GPU サーバーは一般的な 1CPU8 コアであり、8 コア共有のリングバスのアーキテクチャも一般的な PC と共通である。PCI-Express バスは 4 組を 8 プロセスで共有している。詳細な原因を今後検討する必要がある。

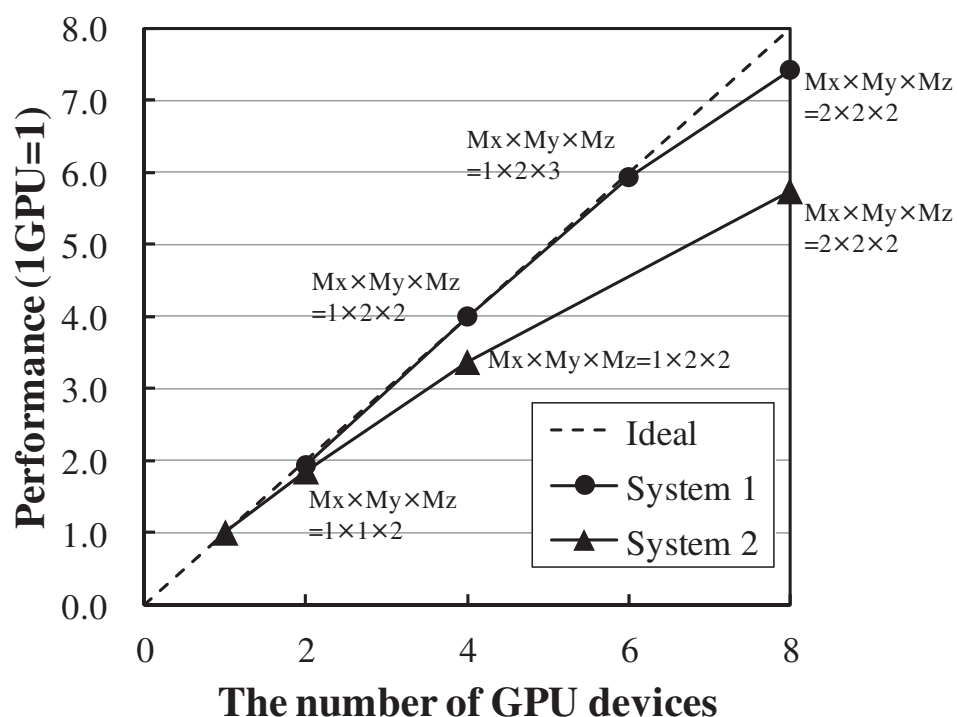


図 6.9 The results of PCG calculation scalability test on a sanitary ware product model. System 1: GeForce GTX 690, 2GPU / node $\times$ 4 nodes. System 2: Tesla K20x, 8GPU / node $\times$ 1 node. System 1 is “Thin node cluster”, while system 2 is “Fat node system”. The numbers of cells, faces and nodes are 2,279,568, 6,925,847 and 2,368,243, respectively.



6.5 数値計算結果

6.5.1 キャビティフローテスト

はじめに VSIAM3 流体ソルバーの基本的な精度検証として, Ghia のキャビティフロー問題 [134] を実施した. 二次元矩形領域 $[-1, 1] \times [-1, 1]$ において, 全ての壁面に NO SLIP 条件を与え, さらに $y = 1$ の壁面に X 軸方向流速 $u = 1$ を与えた. 単相の流体において, $Re = 1000$ および 5000 に相当する粘性係数を与え, 流れが定常に到達した際の流速分布を検証する. 計算格子はこれまでの検証と同様に, 三角形の均等格子を用いる.

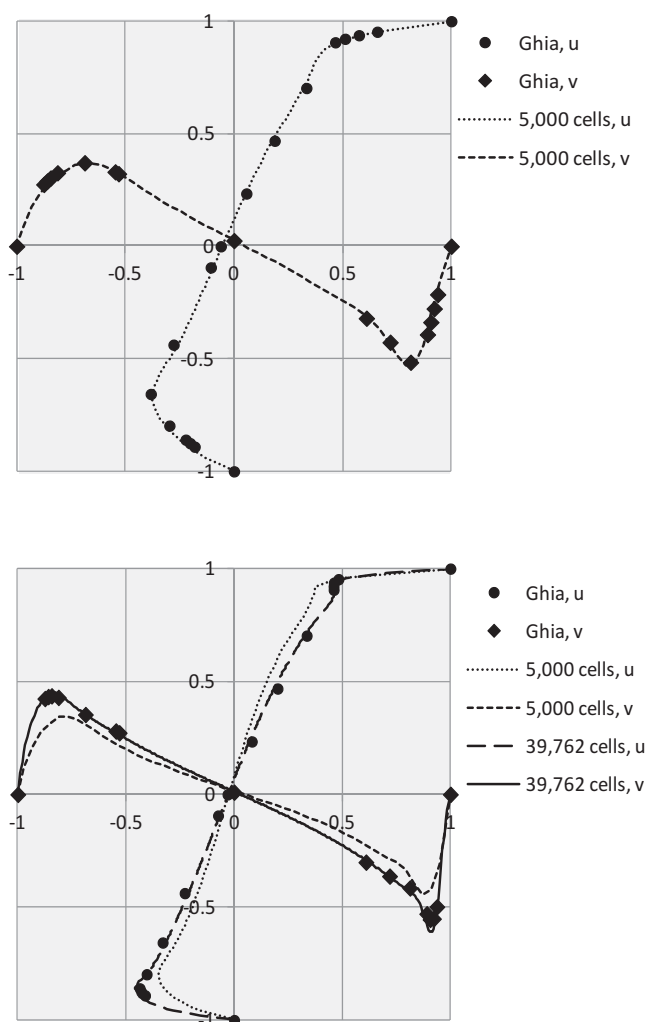


図 6.10 The results of lid-driven vortex flow benchmark test. Upper and lower panels are the velocity profiles respectively for $Re = 1000$ and 5000 cases.

図 6.10 に Ghia の結果との比較を示す．まず $Re = 1000$ では，わずか 5000 格子の計算においても Ghia の結果に収束した．したがってこの計算結果は妥当と考えられる．また $Re = 5000$ では，5000 格子を用いた計算において明らかに計算精度が悪いことがわかる．これは $Re = 1000$ と比較して高レイノルズ数流れであることから，計算格子が粗い場合の数値粘性が計算結果に大きく影響をおよぼしたためと考えられる．しかしながら，移流計算精度検証のときと同じ 39,762 格子を用いた計算結果では，Ghia の結果に収束しており，VSIAM3 流体ソルバーの基本精度に問題はないと考えられる．

6.5.2 ダム崩壊流れテスト

気液二相流ソルバーの精度検証として，Martin & Moyce の二次元ダム崩壊問題 [135] を用いた．初期設定として，図 6.11 に示すように $4a \times 4a$ の二次元矩形空間の左下隅に幅 a ，高さ $2a$ の水柱を配置した． $a = 0.05715$ m である．水と空気の密度は，各々 996.62, 1.1763 kg/m³ とした．粘性係数は，各々 8.544×10^{-4} , 1.862×10^{-5} Pa·s とした．格子は図 6.11 下に示すように，Rudman ベンチマークテストと同じ均等三角形格子を用いた．壁面は 4 面とも全て NO SLIP 条件とし，表面張力や濡れ性は考慮しない．

図 6.12 にシミュレーション結果を示す．崩壊した水柱は右方に進み，右側面の壁に衝突後，壁面に沿って上昇するが，勢いは止まらずやがて天井に衝突し，天井を伝って左方に進んだ後，勢いを失って落下しているようすが分かる．この過程で初期の水柱は薄膜，液滴に変形していき，落下時には巻き込み気泡まで生成する複雑な現象を再現できている．

気液界面は終始同等の界面幅を維持しており，STAA 法により界面の数値拡散を効果的に抑制していることが分かる．さらに水と空気の密度比が約 900 倍と大きく，計算安定性の観点からは厳しい条件にも関わらず，水面や液滴の不自然な動きや振動などは見られず計算は終始安定していることが分かった．

また図 6.13 に，水柱が右方に前進するときの液面先端位置および水柱左端の液面高さ位置の時系列変化を，Martin & Moyce の実験結果と併せて示す．液面先端位置については，実験結果よりも若干シミュレーション結果の方が右方に前進する速度が速いものの，概ねよく一致していることが分かる．この数値計算では濡れ性は考慮していないが，実験では何らかの濡れ性により水膜が前進する速度が妨げられていると推測され，そのような不確定性を除外して計算結果を観測すると，ほぼ妥当な計算結果といえる．

また液面高さの時系列変化で見ると，計算結果は実験とよく一致していることが分かる．これは先ほどの液膜先端とは異なり，液面高さの変化に対する壁面濡れ性条件がおよぼす影響が小さいためと推測される．以上の結果から，提案計算手法および計算ソルバーの妥当性が示された．



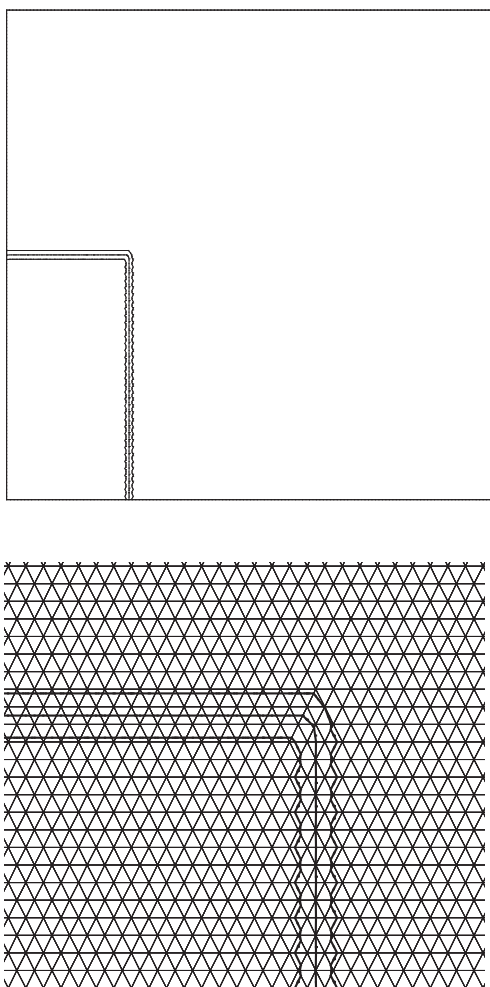


図 6.11 The initial profile of dam break benchmark test. Three solid lines in upper figure indicate $F = 0.05, 0.5$ and 0.95 contours. Lower figure shows the mesh structure near the corner of liquid.

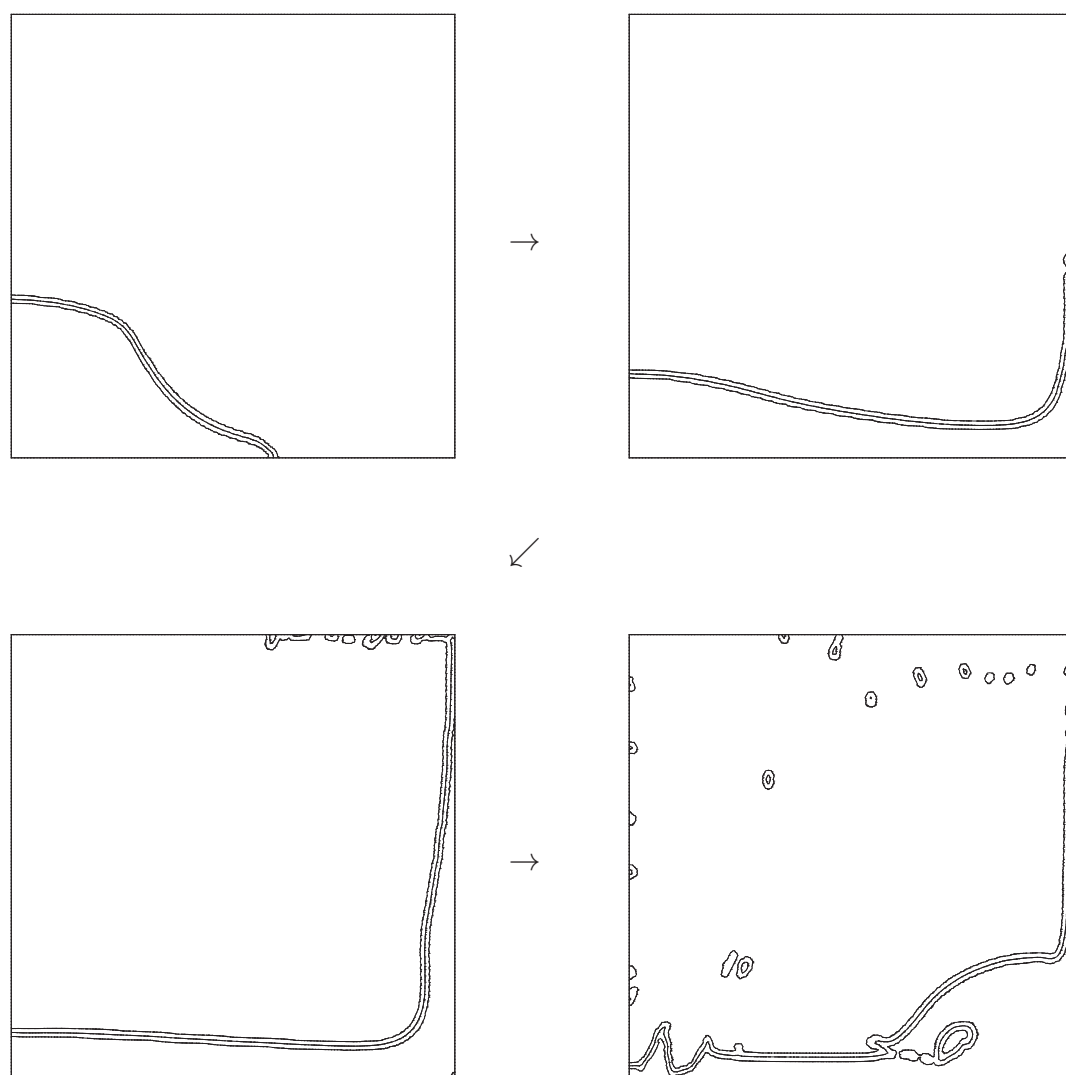


図 6.12 The results of dam break benchmark test. Three solid lines indicate $F = 0.05, 0.5$ and 0.95 contours. Four panels show the results after 0.10, 0.20, 0.30 and 0.50 second, respectively.

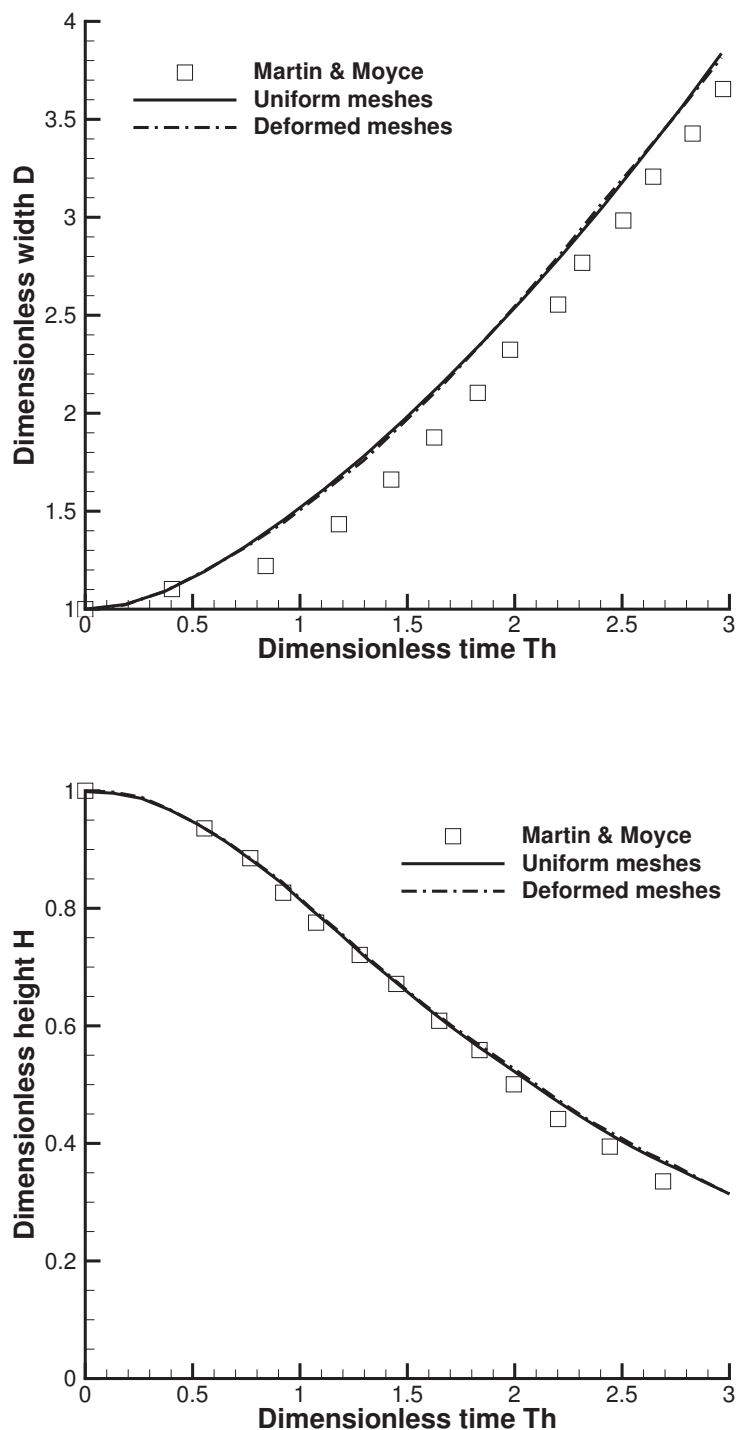


図 6.13 The comparison of results of dam break benchmark test between present and Martin & Moyce's experiment. Upper is the profile of the front position of water, where $T_h \equiv t\sqrt{2g/a}$, $D \equiv x/a$. Lower is the profile of the height of water column, where $T_h \equiv t\sqrt{2g/(2a)}$, $H \equiv y/(2a)$.

6.5.3 衛生陶器シミュレーションへの適用

衛生陶器表面の薄膜流れの解像度を向上させつつ全体の格子数の増加を抑制するため、

1. 計算領域全体をほぼ均等な非構造格子で分割する.
2. 陶器表面境界部に、1 ミリ以下のプリズムメッシュを数層挿入する.

というメッシュ構造により、効率的に表面近傍の解像度を向上させる方法を採用した. この方針を元に作成した非構造格子衛生陶器便鉢モデルを図 6.14 に示す. 点線は分割領域の境界線を表し、 $2 \times 4 = 8$ 分割としている.

シミュレーション結果を図 6.15 に示す. 陶器壁面の境界条件は全て NO SLIP 条件としている. また表面張力および濡れ性は考慮していない. 図 6.15 は衛生陶器を鉛直上方から下方に見下ろした方向にて表示している. 図の上方より一定流量の洗浄水を導水路を通して陶器の便鉢部に流入させている. 導水路の吐水口から便鉢部の開放領域に吐水されると、水柱は陶器表面に薄膜状に広がりながら陶器を反時計回りに旋回し、一周して吐水口からの水膜に合流していく. 図 6.15 では、 $3/4$ 周までの計算結果を示している. このシミュレーション結果からは、便鉢部において

1. 吐水口に近い左下面で水が薄膜状に広がっている.

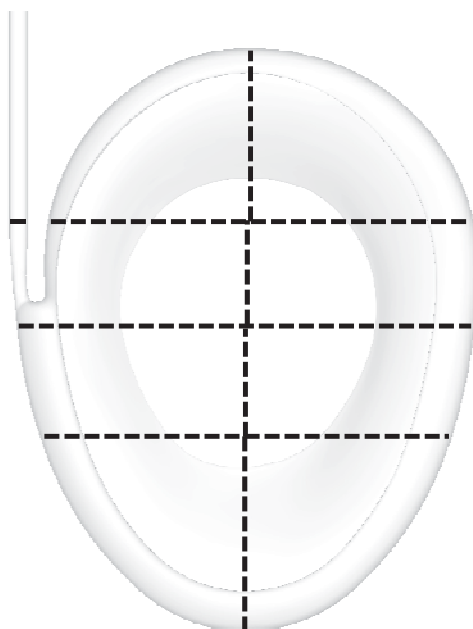


図 6.14 The simulation model of a sanitary ware.

2. 右下面では、吐水が遠心力により便鉢部側面に押し付けられるため、水の広がり作用が弱い。
3. 吐水が右上面に到達すると、遠心力が弱まり再び水が面状に広がっている。

ということが分かる。また 3/4 周の図の点線部において、水膜の厚み方向の断面を図 6.16 に示す。水膜の厚みはある程度の分布を持っているが、わずかに境界層格子数メッシュ分のかなり薄い場所があることが分かる。

図 6.17 に、3/4 周時の実験結果との比較結果を示す。中央の図は提案手法、下の図が従来の直交格子での計算を表す。直交格子では 0.5 ミリの細分格子を用いてスーパーコンピュータ TSUBAME2 で計算している一方、提案手法では最小 0.2 ミリの薄膜プリズム格子と 1.5 ミリ程度の非構造格子を組み合わせたモデルを小規模 PC クラスタにて計算しており、計算格子数は直交格子の 1/10 以下である。

実験結果の色が濃いところは水膜の厚いところを示しており、便鉢部の右上部が最も水膜が厚い。この傾向は提案手法でも同様であるが、従来の直交格子では明らかに水膜の広がり是不十分であり、提案の非構造格子手法により精度が向上していることが分かる。また便鉢部左下面は、従来手法ではほとんど薄膜が再現できていないが、提案手法では実験結果に近い結果が得られている。さらに中央上部の水膜先端部の形状も提案手法の結果は実験によく符合していることが分かる。便鉢部右下面は薄膜が破断し液滴状に分裂しているが、図 6.16 の左断面図から推測されるように、該当部分は境界層格子数メッシュ以下の水膜厚みになって格子解像度の限界に達し、数値計算上の誤差が現れていると考えられる。

以上の結果より、提案の非構造格子計算手法により、従来の直交格子計算手法と比較して大幅に少ない格子数を用いても、定性的に薄膜流れ精度が大幅に改善することが分かった。ただし薄膜流れの定量的精度評価のためには、今後 PIV 等を用いた流速分布や水膜厚みの実機測定手法の確立および詳細な計算結果検証が必要である。

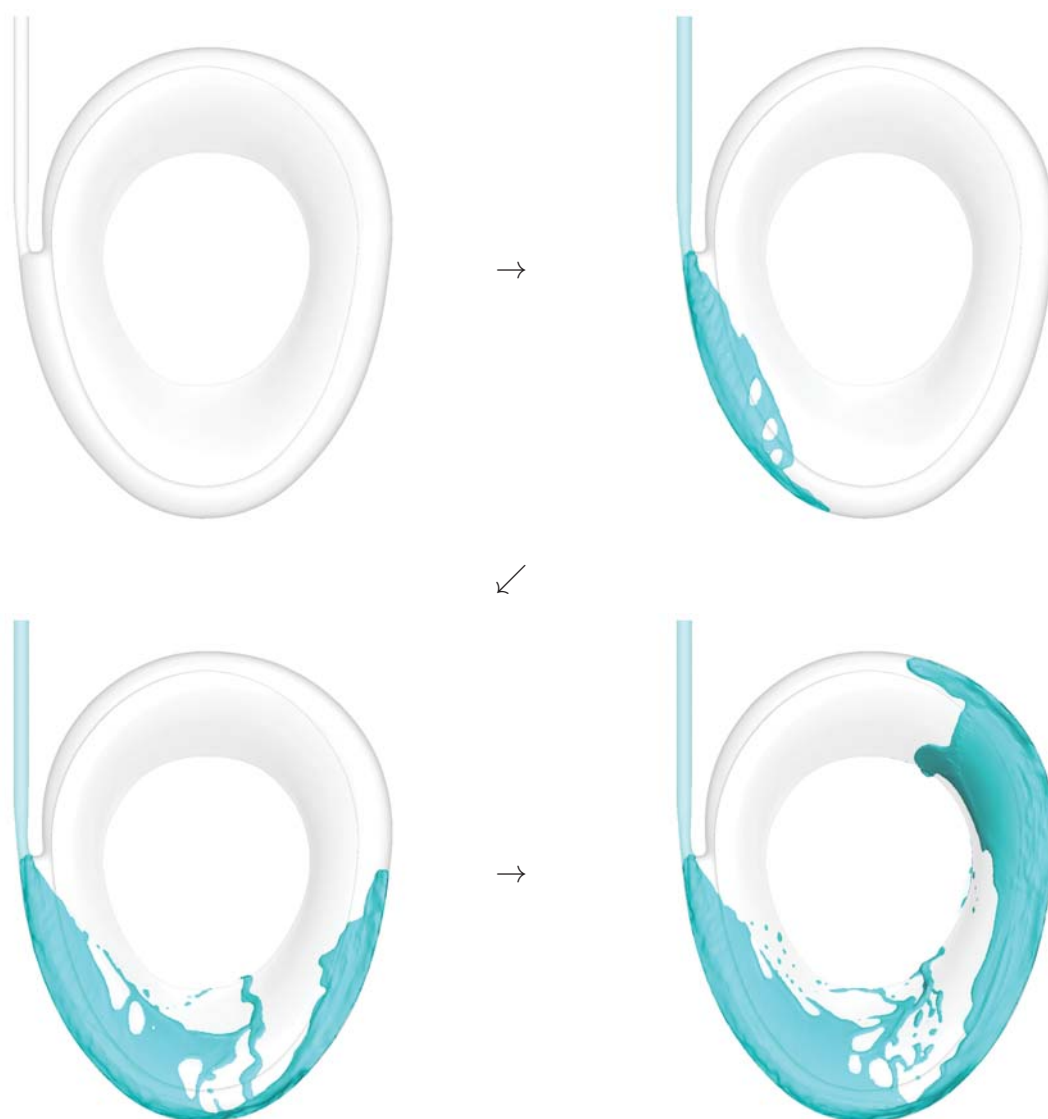


図 6.15 The results of thin water film flow on the surface of a sanitary ware.

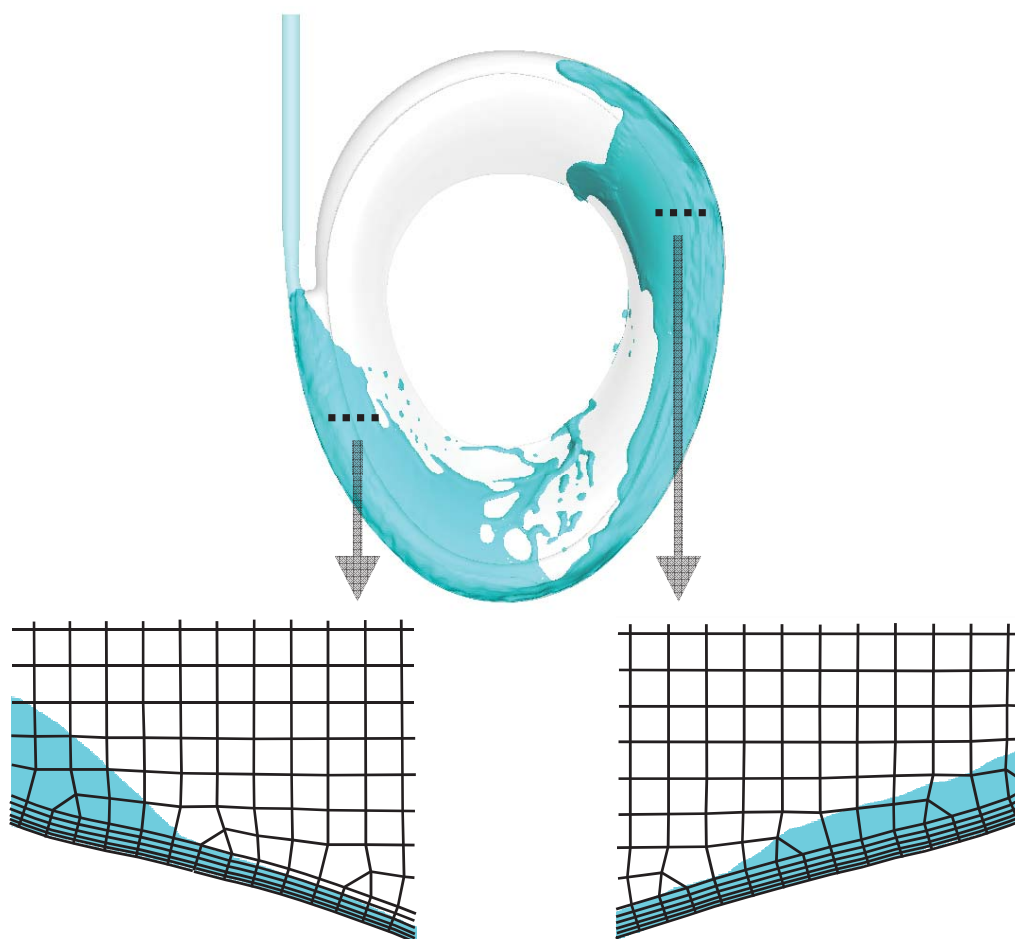


図 6.16 The results of thin water film flow on the surface of a sanitary ware.



図 6.17 The comparison of the result of present method (middle) and our existing method using Cartesian grid (lower) with the experimental result (upper).

6.6 第 6 章のまとめ

超節水型の衛生陶器における陶器表面薄膜流れを精度よくシミュレートすることを目的とし、物体境界面に沿った高精細な境界層格子を含む任意多面体非構造格子の三次元気液二相流シミュレーション手法を開発した。任意多面体は従来の四面体、六面体、ピラミッド五面体およびプリズム五面体のみで構成される非構造格子とは異なり、格子生成の自由度が高く、全体のメッシュ数を削減できるなどの利点がある。また直交格子において物体境界近傍のみを細分化する Mesh Refinement (MR) 法と比較しても、境界層厚み方向のみ細分する提案手法はメッシュ数を少なくすることができると考えられる。

任意多面体における移流計算手法として、第 3 章の直交格子における UTI-VSIAM3 を任意多面体非構造格子に拡張する新しい VSIAM3 定式化を提案した。非構造格子 VSIAM3 では、従来の非構造格子有限体積法で広く用いられているセル中心値 (体積積分平均値) のみによるシングルモーメント法とは異なり、セル界面の面積積分平均値も併せて更新するマルチモーメント法が用いられる。体積積分平均値と面積積分平均値を組み合わせることで二次補間関数を構築する。この補間式を各面のフラックス計算に用いることで、一般的手法である TVD スキームと比較して大きく計算量を増加させることなく、効率的に精度を向上させることができた。各種の移流計算ベンチマークテストにより、定性的かつ定量的に TVD スキームより高精度であることを示した。VSIAM3 の計算量を定量的に把握するため、1 回の移流計算ステップで実行される演算数やデータ移動量および FLOP 値を測定したところ、1 計算ステップでは TVD スキームの 1.5 倍前後と判明した。TVD スキームはオイラー手法であり通常は 1 タイムステップあたり 2~3 回の反復が行われる一方、VSIAM3 は通常セミラグランジェ計算であり反復は必要ないため、1 タイムステップでの計算時間は、VSIAM3 の方が短い傾向にあることが判明した。したがってマルチモーメント法であるにも関わらず計算効率が高いことを示した。

また気液界面捕獲手法として、第 3 章の直交格子における STAA 法を任意多面体非構造格子に拡張する新しい STAA 法定式化を提案した。VSIAM3 による流体率移流計算で生じた数値拡散を、気液界面方向に人工的に逆拡散させる際に、複雑な非構造格子形状を考慮することなく、隣接格子における中心点座標と気液界面方向ベクトルから数値拡散分配比を決定する簡易なアルゴリズムを見出した。移流計算ベンチマークテストにより、効果的に数値拡散を補正し、かつ VSIAM3 との組み合わせにより既存の代数的 VOF 法よりも高精度な結果が得られることを示した。

VSIAM3 と STAA 法を組み合わせた GPGPU 気液二相流ソルバーを開発した。非構造格子では不連続なメモリアクセスとなることは避けられないが、NVidia CUDA プラットフォーム上では、CPU コードと比較して平均 10 倍程度の加速率を達成し、非構造格子における

GPGPU 利用に問題はないことを示した。また並列計算手法として、第 3 章の直交格子で提案した境界不整合領域分割法を非構造格子に拡張する手法を提案した。仮想直交格子 (=バケット) に各非構造格子を割り振り、バケットに対して境界不整合領域分割法を適用することで、極めて容易に均等に近い領域分割を実現した。8GPU までの小規模クラスタにて、小規模の衛生陶器シミュレーションのテスト計算を行い、ほぼ理想的な並列化効率を得られることを示した。

開発したソルバーを用い、キャビティフローおよびダム崩壊ベンチマークテストを実施し、妥当な計算精度およびロバスト性を示した。ダム崩壊ベンチマークでは、気液界面は終始コンパクトに保持され、STAA 法の数値拡散補正が効果的に機能していることが分かった。衛生陶器シミュレーションに適用し、従来の直交格子では大容量メッシュをスパコンで計算しても再現できなかった衛生陶器表面薄膜流れにおいて、わずか 1/10 程度の任意多面体非構造格子を用いて小規模 GPU クラスタで計算することにより、定性的に精度を大きく改善することができた。以上の評価結果から、提案した計算手法および並列 GPGPU ソルバーのフレームワークは妥当であることを示した。



第 7 章

結論

本研究では、衛生陶器における水や空気および固体を含む混相流のシミュレーションを高精度かつ高速に計算する数値計算手法、フレームワークおよびシミュレーションコードの開発を行った。まず計算手法では、既存の保存型 CIP 法 (CIP-CSL) の計算量を低減した VSIAM3 有限体積法に基づく移流計算手法を改良した。VSIAM3 は、VIA とよばれる、既存の有限体積法で利用される格子中心値 (格子の体積積分平均値) の他に、SIA とよばれる格子界面の面積積分値を併せて変数値として保持・更新することで、VIA と SIA との組み合わせによる 1 格子内での高次補間関数を構築できる。このように複数のモーメント (物理量関連値) を保持する手法をマルチモーメント法とよぶ。

直交格子では、既存の VSIAM3 が X 軸 $\rightarrow Y$ 軸 $\rightarrow Z$ 軸の順に一次元移流計算を 3 回計算する次元分割法に対し、提案の UTI-VSIAM3 では三次元の移流計算を各格子界面の法線方向および接線方向の 2 ステップに分解して計算する非次元分割法を採用した。次元分割型の VSIAM3 が次元対称性を持たない定式化であることに対し、UTI-VSIAM3 は本質的に次元対称な定式化であることが大きな違いである。これにより、UTI-VSIAM3 を非直交格子に拡張することが可能になる。直交格子での移流計算テストでは、既存の VSIAM3 と同等の結果を保持し、一般手法として市販ソフトウェアなどで広く用いられている MUSCL と比較して高精度な結果が得られた。本手法を Navier-Stokes 方程式における運動量 (流速) の移流項の計算に適用することで、流体計算の精度向上を見込める。

次に水と空気の気液界面を高精度に捕獲する計算手法として、独自の気液界面数値拡散補正スキームである STAA 法を UTI-VSIAM3 と組み合わせる新しい代数的 VOF 法を提案した。STAA 法では、まず流体率 (VOF 関数) からレベルセット関数を高速に生成し、次にレベルセット関数をもとに気液界面法線ベクトルの計算および数値拡散補正対象格子の選別を行い、気液界面法線方向に逆数値拡散フラックスを生成して、気液界面近傍の数値拡散を除去する。この手法により移流計算スキームでの気液界面形状情報を消失させることなく高精度に数値拡散補正が可能となる。ベンチマークテストでは様々な既存の移流計算スキームと STAA を組

み合わせて検証したところ、UTI-VSIAM3 との組み合わせが最も精度が良いことが分かった。またこの組み合わせにおいては、THINC/WLIC や THINC/SW などの既存手法と比較して高精度な移流計算結果が得られることを示した。

以上の提案手法を有限体積法に組み込んだ直交格子多相流計算ソルバーを開発した。計算手法はフラクショナルステップ法を用いた。ダム崩壊ベンチマークテストや気泡上昇問題テストでは問題ない計算精度が得られた。この手法を衛生陶器やキッチンの水の流れのシミュレーションに組み込み、妥当な結果が得られた。

次に、上記のフレームワークを拡張し、GPGPU コード化による計算加速化および領域分割法による計算並列化を行った。UTI-VSIAM3 および STAA 法は単純なベクトル演算の集合で構成されるため、GPGPU 化および並列化は容易に対応できることが分かった。また NCDDM とよぶ境界不整合の三次元領域分割方法を提案し、さらにオーバーラップ通信を実装することにより、GPGPU 計算において高い並列化効率が実現できることを実証した。スパコン TSUBAME による動作検証では約 1 億メッシュの計算で 150GPU で 100 倍程度の並列化効率を実現し、微細メッシュでの高精度な計算が実用的な計算時間で実施できるようになった。

このフレームワークの改良として、運動量の保存性を向上させる改良移流計算手法を提案した。本手法は VSIAM3-CM とよばれる。一般に非圧縮性流れの計算では非保存型の Navier-Stokes 方程式が用いられるが、衛生陶器のシミュレーションでは、洗浄時の水はねが実現象よりも過剰に評価されることが分かっている。この原因は運動量の保存性が満たされていないため、気相側の流速が数値拡散により液相側に伝播し、不自然な微小液滴の動きを発生させるためと考えられる。そこで従来の流速に関する移流方程式を解く方法ではなく、元の Navier-Stokes 方程式の運動量 (=密度 × 流速) に関する輸送方程式を解く方法を考案した。通常は密度の計算は VOF 関数から計算されるが、提案手法では、密度の輸送方程式を運動量の輸送方程式と同じ移流計算手法 (VSIAM3) により一体的に計算し、密度と運動量の一貫性を保持することで、運動量を密度で除算して得られる流速の計算安定化を図った。ただしこの密度は物理量としては用いられず、毎時間ステップに VOF 関数から初期設定される。本提案手法を様々なテスト計算に適用し、従来の非保存型解法による微小液滴の不自然な動きが解消されることを示した。衛生陶器シミュレーションに適用し、従来手法で発生していた洗浄時の飛沫がなくなり、実現象に適合するようになった。

最後に、超節水型の衛生陶器の流れにおいて高精度なシミュレーションが必要とされている。衛生陶器表面の薄膜流れに対応可能な新しい非構造格子二相流シミュレーション手法を提案した。複雑曲面形状に沿った薄膜流れは、先述の直交格子計算ではもっとも困難な計算であり、また局所細分化格子の利用では衛生陶器シミュレーションにおいて計算格子数の飛躍的増大が見込まれる。そこで格子数を大幅に増加させることなく表面流れの精度向上を図るため非構造格子の利用を検討した。



まず UTI-VSIAM3 の非構造格子への拡張では、SIA を用いてグリーン・ガウスの定理から仮想上流点の値を計算することにより、直交格子と本質的に同一な一次元保存型 CIP 法を、各格子界面法線方向に対し構築できることを見出した。さらに格子界面において流速を法線方向と接線方向に分解し、接線方向については UTI-VSIAM3 と同様な定式化を構築した。本手法を非構造格子 VSIAM3 とよぶ。UTI を冠しないのは、そもそも非構造格子では次元分割という概念がないからである。しかしながら定式化としては直交格子での VSIAM3 を非構造格子に拡張したものではなく、UTI-VSIAM3 を拡張したものである。非構造格子 VSIAM3 を規則的および不規則的な三角形格子における移流計算ベンチマークテストに適用し、代表的な既存手法である TVD スキームよりも高精度な計算結果が得られることを実証した。一方、計算量の TVD スキームとの比較を実施したところ、通常 TVD スキームと併用される TVD-RK2 もしくは TVD-RK3 による高次時間積分法においては、セミラグランジェ法の非構造格子 VSIAM3 の方が計算時間が長いとはいえない結果となった。これは非構造格子 VSIAM3 では直接フラックスの計算を行い、この値をそのまま SIA として保存しているため効率的な計算となっていること、またセミラグランジェ法を用いているため、TVD-RK2 や TVD-RK3 で行われる反復計算が必要ないことが原因として挙げられる。したがって非構造格子 VSIAM3 は計算精度および計算効率の両面から既存手法よりも優れた手法であるといえる。

次に STAA 法の非構造格子への拡張では、格子体積に対する数値拡散補正フラックスではなく、各格子中心点に対する逆数値拡散フラックスという概念を提案し、気液界面法線方向と各隣接格子点方向から数値拡散の分配を行う簡易なアルゴリズムを開発した。非構造格子 VSIAM3 のときと同様の移流計算ベンチマークテストを実施し、規則格子および不規則格子ともに良好な結果が得られた。既存の VOF/PLIC や CICSAM および MULES といった代表的な VOF 法の非構造格子移流計算プログラムを作成し、非構造格子 VSIAM3+STAA との比較を行ったところ、気液界面形状の保持性においては最も良い結果が得られた。STAA 法では、気液界面近傍における数値拡散はある程度の管理されたメッシュ幅の内側において許容しているため、 L_1 誤差の結果では VOF/PLIC に及ばない部分はあったものの、他の代数的 VOF 法よりは明確に L_1 誤差が小さい結果が得られた。

非構造格子におけるダム崩壊流れのベンチマークテストでは規則格子および不規則格子ともに妥当な結果が得られた。気液界面の数値拡散の拡大も見られず、STAA 法による数値拡散補正が有効に機能していることを示した。

またこの非構造格子有限体積法フレームワークを GPGPU 並列計算に拡張した。直交格子と同様の GPGPU コード化および境界不整合領域分割法を適用可能であり、直交格子と同様の GPGPU 加速化および並列化効率が得られた。したがって、非構造格子における GPGPU 計算機の適用性は構造格子同様に有効であり、非構造格子特有の不規則なメモリアクセスが GPU 加速化におよぼす影響は大きくないことを実証した。

開発した非構造格子二相流計算手法を超節水型の衛生陶器の薄膜流れシミュレーションに適

用した．実機での流れとの比較では，薄膜流れの上方から見たフローパターンでは従来の直交格子のスパコンでの計算結果と比較して，わずか 8GPU の GPU サーバーでの非構造格子計算により 1/10 以下の格子数で大幅に精度が向上したことを実証した．直交格子では表現できなかった吐水口近傍の薄膜の広がりがある程度再現できるようになっており，境界層格子を挿入した非構造格子において直交格子より定性的に大きく精度を改善した．

以上本研究では，実製品における気液混相流を高精度かつ高効率に計算するための移流計算手法および並列 GPGPU 流体計算フレームワークを提案・検証・構築した．本成果は，第 3～4 章の構造格子移流計算手法「UTI-VSIAM3」および気液界面捕獲手法「STAA」の提案と検証および実製品への適用として [136][137] に，また第 6 章の非構造格子移流計算手法「VSIAM3」および気液界面捕獲手法「STAA」の提案と検証および実製品への適用として [138] にまとめ，公開した．本フレームワークを適用することにより，複雑な流路構造を持つ衛生陶器製品において，無数の微細気泡やミリ単位の薄膜流れが混在する複雑な気液二相流れを再現するという非常に挑戦的な課題を解決できるようになり，節水化を初めとする製品性能の向上や試作試験費用の削減および開発期間の短縮に貢献すると考えられる．また提案の多相流シミュレーション手法は，他の産業界の製品への適用のみならず，原子力発電所や河川，海洋の流体シミュレーションなど，様々な異分野への応用が可能と考えられる．

今後の研究課題としては，開発したフレームワークを板状薄膜流れや管内流れなどの基本的な二相流検証問題や衛生陶器薄膜流れ等様々な製品の多相流シミュレーションに適用し，定量的な計算精度や計算速度，計算効率について検証を行う必要がある．提案手法の計算精度の改良については，非構造格子 VSIAM3 においてより高次の補間関数を用いた高精度化が考えられる．現状では CIP-CSL による上流側の値による再構築が行われているが，下流側の値も併用することで，より数値粘性の小さい計算スキームを構築できると考えられる．ただし実用計算において必須な計算のロバスト性をどのように確保するかは難しい課題と予想される．



参考文献

- [1] L. B. Lucy. A numerical approach to the testing of the fission hypothesis. *Astro. J.*, Vol. 82, No. 12, p. 1013, 1977.
- [2] R. A. Gingold and J. J. Monaghan. Smoothed particle hydrodynamics : theory and application to non-spherical stars. *Mon. Not. R. astr. Soc.*, Vol. 181, p. 375, 1977.
- [3] J. P. Morris, P. J. Fox, and Zhu Y. Modeling low reynolds number incompressible flows using SPH. *J. Comput. Phys.*, Vol. 136, p. 214, 1997.
- [4] S. Koshizuka and Y. Oka. Moving-particle semi-implicit method for fragmentation of incompressible fluid. *Nuclear Science and Engineering*, Vol. 123, p. 421, 1996.
- [5] S. Koshizuka. Current achievement and future perspectives on particle simulation technologies for fluid dynamics and heat transfer. *J. Nucl. Sci.*, Vol. 48, p. 155, 2011.
- [6] H. Y. Yoon, S. Koshizuka, and Y. Oka. A mesh-free numerical method for direct simulation of gas-liquid phase interface. *Nuclear Science and Engineering*, Vol. 133, p. 192, 1999.
- [7] A. M. Moqaddam, S. S. Chikatamarla, and I. V. Karlin. Simulation of binary droplet collisions with the entropic lattice boltzmann method. *Phys. Fluids*, Vol. 28, No. 022106, 2016.
- [8] D. L. Brown, K. H. Luo, R. Liu, and P. Lv. Binary droplet collision simulations by a multiphase cascaded lattice boltzmann method. *Phys. Fluids*, Vol. 26, No. 022106, 2014.
- [9] 高田尚樹, 三澤雅樹. 二相流界面追跡シミュレーションのためのフェーズフィールドモデルと格子ボルツマンスキームの検討. 第 18 回数値流体力学シンポジウム 講演番号 D4-2, 2004.
- [10] T. Nomura. Ale finite element computations of fluid-structure interaction problems. *Comput. Methods App. Mech. Eng.*, Vol. 112, p. 291, 1994.
- [11] S. Takagi, Y. Matsumoto, and H. Huang. Numerical analysis of a single rising bubble using boundary-fitted coordinate system. *Japan Soc. Mech. Eng., B*, Vol. 61, No. 586, p. 1976, 1995.
- [12] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *J. Comput. Phys.*, Vol. 100, p. 25, 1992.
- [13] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawachi, W. Tauber, J. Han, S. Nas, and Y. Jan. An front-tracking method for the computations of multiphase flow. *J. Comput. Phys.*, Vol. 169, p. 708, 2001.
- [14] 大山峻幸, 伊井仁志, 杉山和靖, 竹内伸太郎, 高木周, 松本洋一郎. フロント・トラッキング法を用いた自由界面と剛体壁における単一気泡の反発現象の解析. *Japanese J. Multiphase Flow*, Vol. 26, No. 1, p. 60, 2012.
- [15] S. Osher and J. A. Sethian. Fronts propagating with curvature dependent speed : Algo-

- rithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, Vol. 79, p. 12, 1988.
- [16] S. Osher and R. P. Fedkiwy. Level set methods: An overview and some recent results. *J. Comput. Phys.*, Vol. 39, p. 463, 2001.
 - [17] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flows. *J. Comput. Phys.*, Vol. 114, p. 146, 1994.
 - [18] C. W. Hirt and B. D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.*, Vol. 39, p. 201, 1981.
 - [19] W. F. Noh and P. Woodward. Slic (simple line interface calculation). *Proceedings of fifth international conference on Numerical Methods in Fluid Dynamics*, 1976.
 - [20] D. L. Youngs. Time-dependent multi-material flow with large fluid distortion. *Numerical Methods for Fluid Dynamics*, p. 273, 1982.
 - [21] 越塚誠一. 粒子法. 丸善, 2005.
 - [22] W. G. Hoover. 粒子法による力学 連続体シミュレーションへの展開. 森北出版, 2008.
 - [23] T. A. Gronnelov. Smoothed particle hydrodynamics for astronomical model demonstration. A project of SPH and astronomical model on Greenleaf, 2010.
 - [24] E. S. Lee, C. Moulinec, R. Xu, D. Violeau, D. Laurence, and P. Stansby. Comparisons of weakly compressible and truly incompressible algorithms for the sph mesh free particle method. *J. Comput. Phys.*, Vol. 227, p. 8417, 2008.
 - [25] J. Hardy, Y. Pomeau, and O. de Pazzis. Time evolution of a two-dimensional model system. i. invariant states and time correlation functions. *J. Math. Phys.*, Vol. 14, p. 1746, 1973.
 - [26] 薦原道久, 高田尚樹, 片岡武. 格子気体法・格子ボルツマン法. コロナ社, 1999.
 - [27] Y. H. Qian, D. d'Humieres, and P. Lallemand. Lattice bgk models for navier-stokes equation. *Europhysics letters*, Vol. 17, No. 6, p. 479, 1992.
 - [28] M. R. Swift, W. R. Osborn, and J. M. Yeomans. Lattice boltzmann simulation of nonideal fluids. *Physical Review Letters*, Vol. 75, No. 5, p. 830, 1995.
 - [29] M. R. Swift, E. Orlandini, W. R. Osborn, and J. M. Yeomans. Lattice boltzmann simulations of liquid-gas and binary fluid systems. *Physical Review E*, Vol. 54, No. 5, p. 5041, 1996.
 - [30] D. L. Youngs. An interface tracking method for a 3d eulerian hydrodynamics code. *Technical Report, 44/92/35, AWRE*, 1984.
 - [31] T. Inamuro, T. Ogata, S. Tajima, and N. Konishi. A lattice boltzmann method for incompressible two-phase flows with large density differences. *J. Comput. Phys.*, Vol. 198, p. 628, 2004.
 - [32] 稲室隆二. 二相系格子ボルツマン法の最近の研究動向. *J. Heat Trans. Soc. Japan*, Vol. 55, No. 233, p. 8, 2016.
 - [33] G. S. Jiang and C. W. Shu. Efficient implementation of weighted eno schemes. *J. Comput. Phys.*, Vol. 126, p. 202, 1996.
 - [34] R. Borges, M. Carmona, B. Costa, and W. S. Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *J. Comput. Phys.*, Vol. 227, p. 3191, 2008.
 - [35] Y. C. Chang, T. Y. Hou, R. Merriman, and S. Osher. A level set formulation of Eulerian interface capturing methods for incompressible fluid flows. *J. Comput. Phys.*, Vol. 124, p.



- 449, 1996.
- [36] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved level set method for incompressible two-phase flows. *Computers and Fluids*, Vol. 27, No. 5-6, p. 663, 1998.
 - [37] E. Marchandise, J. F. Remacle, and N. Chevaugeon. A quadrature-free discontinuous Galerkin method for the level set equation. *J. Comput. Phys.*, Vol. 212, p. 338, 2006.
 - [38] J. H. ファーティガー, M. ペリッチ. コンピュータによる流体力学. 丸善出版, 2012.
 - [39] W. E. Lorensen and H. E. Cline. Marching cubes : A high resolution 3d surface construction algorithm. *Computer Graphics*, Vol. 21, No. 4, p. 163, 1987.
 - [40] 石井智海, 安田孝美, 横井茂樹, 鳥脇純一郎. マーチングキューブ法の改良アルゴリズムについて. *グラフィクスと CAD*, Vol. 60, No. 9, p. 63, 1992.
 - [41] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *J. Comput. Phys.*, Vol. 141, p. 112, 1998.
 - [42] R. Scardovelli and S. Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *J. Comput. Phys.*, Vol. 164, p. 228, 2000.
 - [43] 功刀資彰. 自由界面を含む多相流の直接数値解析法. 日本機械学会論文集 B 編, Vol. 63, No. 609, p. 1576, 1997.
 - [44] F. Xiao, Y. Honma, and K. Kono. A simple algebraic interface capturing scheme using hyperbolic tangent function. *Int. J. Numer. Methods fluid*, Vol. 48, p. 1023, 2005.
 - [45] F. Xiao, S. Ii, and C. G. Chen. Revisit to the thinc scheme : A simple algebraic VOF algorithm. *J. Comput. Phys.*, Vol. 230, p. 7086, 2011.
 - [46] K. Yokoi. Efficient implementation of thinc scheme : A simple and practical smoothed VOF algorithm. *Int. J. Comput. Phys.*, Vol. 226, p. 1985, 2007.
 - [47] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, Vol. 49, p. 357, 1983.
 - [48] P. K. Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM. J. Numer. Anal.*, Vol. 21, No. 5, p. 995, 1984.
 - [49] B. Cockburn and C. W. Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws ii: General framework. *Math. Comput.*, Vol. 52, p. 411, 1989.
 - [50] B. Cockburn, S. Y. Lin, and C. W. Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws iii: One-dimensional systems. *J. Comput. Phys.*, Vol. 84, p. 90, 1989.
 - [51] B. Cockburn, S. Hou, and C. W. Shu. Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws iv: The multidimensional case. *Math. Comput.*, Vol. 54, p. 545, 1990.
 - [52] A. Harten, B. Enquist, S. Osher, and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *J. Comput. Phys.*, Vol. 71, p. 231, 1987.
 - [53] T. Sonar. On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: Polynomial recovery, accuracy and stencil selection. *Comput. Methods Appl. Mech. Eng.*, Vol. 140, p. 157, 1997.
 - [54] R. P. Brent. *Algorithms for minimization without derivatives*. Dover Books on Mathematics, 2013.

- [55] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [56] C. W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.*, Vol. 77, p. 439, 1988.
- [57] Y. Wei, L. Shu-hong, and W. Yu-lin. An unsplit lagrangian advection scheme for volume of fluid method. *J. Hydrodynamics*, Vol. 22, p. 73, 2010.
- [58] C. S. Wu and D. L. Young. Interface reconstruction with split lagrangian advection for two-dimensional interfacial flows. *J. Mechanics*, Vol. 29, No. 2, p. 253, 2013.
- [59] J. E. Pilliod and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *J. Comput. Phys.*, Vol. 199, p. 465, 2004.
- [60] Y. Renardy and M. Renardy. Prost : A parabolic reconstruction of surface tension for the volume -of-fluid method. *J. Comput. Phys.*, Vol. 183, p. 400, 2002.
- [61] O. Ubbink and R. I. Issa. A method for capturing sharp fluid interfaces on arbitrary meshes. *J. Comput. Phys.*, Vol. 153, p. 26, 1999.
- [62] B. P. Leonard. A stable and accurate convective modeling procedure based on quadratic upstream interpolation. *Comput. Methods App. Mech. Eng.*, Vol. 19, p. 59, 1979.
- [63] T. Waclawczyk and T. Koronowicz. Comparison of CICSAM and HRIC high-resolution schemes for interface capturing. *J. Theoretical and Applied Mechanics*, Vol. 46, No. 2, p. 325, 2008.
- [64] S. S. Deshpande, L. Anumolu, and M. F. Trujillo. Evaluating the performance of the two-phase flow solver interfoam. *Computational Science and Discovery*, Vol. 5, No. 014016, 2012.
- [65] S. Ii, K. Sugiyama, S. Takeuchi, S. Takagi, Y. Matsumoto, and F. Xiao. An interface capturing method with a continuous function : The thinc method with multi-dimensional reconstruction. *J. Comput. Phys.*, Vol. 231, p. 2328, 2012.
- [66] B. Xie and F. Xiao. Toward efficient and accurate interface capturing on arbitrary hybrid unstructured grids : The thinc method with quadratic surface representation and gaussian quadrature. *J. Comput. Phys.*, Vol. 349, p. 415, 2017.
- [67] M. Sussman and E. G. Puckett. A coupled level set and volume-of-fluid method for computing 3d and axisymmetric incompressible two-phase flows. *J. Comput. Phys.*, Vol. 162, p. 301, 2000.
- [68] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *J. Comput. Phys.*, Vol. 187, p. 110, 2003.
- [69] Z. Wang, J. Yang, B. Koo, and F. Stern. A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves. *Int. J. Multiphase Flow*, Vol. 35, p. 227, 2009.
- [70] T. Yabe, R. Tanaka, T. Nakamura, and F. Xiao. An exactly conservative semi-lagrangian scheme (CIP-CSL) in one dimension. *Monthly Weather Review*, Vol. 129, p. 332, 2001.
- [71] F. Xiao and T. Yabe. Completely conservative and oscillation-less semi-lagrangian schemes for advection transportation. *J. Comput. Phys.*, Vol. 170, p. 498, 2001.
- [72] F. Xiao, T. Yabe, X. Peng, and H. Kobayashi. Conservative and oscillation-less atmospheric transport schemes based on rational functions. *J. Geophysical Research Atmosphere*, Vol.



- 107, pp. ACL 2–1, 2002.
- [73] T. Nakamura, R. Tanaka, T. Yabe, and K. Takizawa. Exactly conservative semi-lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique. *J. Comput. Phys.*, Vol. 174, p. 171, 2001.
- [74] K. Takizawa, T. Yabe, and T. Nakamura. Multi-dimensional semi-lagrangian scheme that guarantees exact conservation. *Comput. Phys. Commun.*, Vol. 148, p. 137, 2002.
- [75] F. Xiao. Unified formulation for compressible and incompressible flows by using multi-integrated moment method : one-dimensional inviscid compressible flow. *J. Comput. Phys.*, Vol. 195, p. 629, 2004.
- [76] F. Xiao, R. Akoh, and S. Ii. Unified formulation for compressible and incompressible flows by using multi-integrated moments ii : multi-dimensional version for compressible and incompressible flows. *J. Comput. Phys.*, Vol. 213, p. 31, 2006.
- [77] F. Xiao, A. Ikebata, and T. Hasegawa. Numerical simulations of free-interface fluids by a multi integrated moment method. *Comput. Struct.*, Vol. 83, p. 409, 2005.
- [78] B. Van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. *J. Comput. Phys.*, Vol. 23, p. 276, 1977.
- [79] B. Van Leer. Towards the ultimate conservative difference scheme. v. a second order sequel to godunov’s method. *J. Comput. Phys.*, Vol. 32, p. 101, 1979.
- [80] 藤井孝蔵. 流体力学の数値計算法. 東京大学出版会, 1994.
- [81] P. Lax and B. Wendroff. Systems of conservation laws. *Commun. Pure Appl. Math.*, Vol. 13, p. 217, 1960.
- [82] M. Rudman. Volume-tracking methods for interfacial flow calculations. *Int. J. Numer. Methods Fluids*, Vol. 24, p. 671, 1997.
- [83] J. U. Brackbill, D. B. Kothe, and C. A. Zemach. A continuum method for modeling surface tension. *J. Comput. Phys.*, Vol. 100, p. 335, 1992.
- [84] K. Yokoi. A practical numerical framework for free surface flows based on CLSVOF method, multi-moment methods and density-scaled CSF model: Numerical simulations of droplet splashing. *J. Comput. Phys.*, Vol. 232, p. 252, 2013.
- [85] A. J. Chorin. Numerical solution of the navier-stokes equations. *Math. Comp.*, Vol. 22, p. 745, 1968.
- [86] A. J. Chorin. On the convergence of discrete approximations to the navier-stokes equation. *Math. Comp.*, Vol. 23, p. 341, 1969.
- [87] J. Kim and P. Moin. Application of a fractional-step method to incompressible navier-stokes equations. *J. Comput. Phys.*, Vol. 59, p. 308, 1985.
- [88] L. S. Caretto, A. D. Gosman, S. V. Patankar, and D. B. Spalding. Two numerical methods for three-dimensional boundary layers. *Comp. Methods Appl. Mech. Eng.*, Vol. 1, p. 39, 1972.
- [89] S. V. Patankar. *Numerical heat transfer and fluid flow*. Hemisphere Publishing Corporation, 1980.
- [90] S. V. Patankar. A calculation procedure for two-dimensional elliptic situations. *Num. Heat Transfer*, Vol. 4, p. 409, 1981.
- [91] J. P. Van Doormal and G. D. Raithby. Enhancements of the simple method for predicting

- incompressible fluid flows. *Numer. Heat Transfer*, Vol. 7, p. 147, 1984.
- [92] R. I. Issa. Solution of the implicitly discretized fluid flow equations by operator-splitting. *J. Comput. Phys.*, Vol. 62, p. 40, 1985.
- [93] J. G. Hnat and J. D. Buckmaster. Spherical cap bubbles and skirt formation. *Phys. Fluids*, Vol. 19, p. 182, 1976.
- [94] M. Sussman and P. Smereka. Axisymmetric free boundary problems. *J. Fluid Mech.*, Vol. 341, p. 269, 1997.
- [95] C. W. Hirt. Volume-fraction techniques : powerful tools for wind engineering. *Proc. 1st Int. Comput. Wind Eng.*, Vol. 46-47, p. 327, 1993.
- [96] F. Xiao. A computational model for suspended large rigid body in 3d unsteady viscous flow. *J. Comput. Phys.*, Vol. 155, p. 348, 1999.
- [97] TSUBAME 2. <http://tsubame.gsic.titech.ac.jp/en/tsubame2-system-architecture>.
- [98] 金田康正. 並列数値処理-高速化と性能向上のために-. コロナ社, 2010.
- [99] 金田行雄, 笹井理生, 石井克哉. 計算科学のための並列計算. 共立出版, 2014.
- [100] R. Barrett, F. T. Chan, J. Donato, M. Berry, and J. Demmel. 反復法 Templates (応用数値計算ライブラリ). 朝倉書店, 1996.
- [101] 河村哲也, 平野広和, 池川昌弘, 川原睦人, 登坂宣好. 非圧縮性流体解析 (数値流体力学シリーズ). 東京大学出版会, 1995.
- [102] G. Tryggvason, R. Scardovelli, and S. Zaleski. *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*. Cambridge University Press, 2011.
- [103] M. Rudman. A volume-tracking method for incompressible multifluid flows with large density variations. *Int. J. Numer. Methods Fluids*, Vol. 28, p. 357, 1998.
- [104] M. Raessi and H. Pitsch. A level-set based methodology for modeling interfacial flows characterized by large density ratios. *ILASS-Americas 22nd Annual Conference*, 2010.
- [105] S. Ghods and M. Herrmann. A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods. *IOP Publishing Phys. Scr.*, Vol. 88, No. T155, 2013.
- [106] M. Bussmann, D. B. Kothe, and J. M. Sicilian. Modeling high density ratio incompressible interfacial flows. *Proceedings of ASME 2002 Fluids Engineering Division Summer Meeting*, No. 31125, 2002.
- [107] M. Raessi and H. Pitsch. Modeling interfacial flows characterized by large density ratios with the level set method. *Center for Turbulence Research*, 2009.
- [108] M. Raessi and H. Pitsch. Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method. *Computers and Fluids*, Vol. 63, p. 70, 2012.
- [109] M. Bussmann, D. B. Kothe, and J. M. Sicilian. Modeling high density ratio incompressible interfacial flows. *Proc. ASME 2002 Fluids Engineering Division Summer Meeting*, 2002.
- [110] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, Vol. 53, p. 484, 1984.
- [111] M. J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, Vol. 82, p. 64, 1989.
- [112] D. F. Martin and P. Colella. A cell-centered adaptive projection method for the incompress-



- ible Euler equations. *J. Comput. Phys.*, Vol. 163, p. 271, 2000.
- [113] S. J. Owen. A survey of unstructured mesh generation technology. IMR, 1998.
- [114] H. Si and K. Gartner. Meshing piecewise linear complexes by constrained delaunay tetrahedralizations. IMR, 2005.
- [115] 谷口健男. FEM のための要素自動分割. 森北出版, 1992.
- [116] B. van Leer. Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme. *J. Comput. Phys.*, Vol. 14, p. 361, 1974.
- [117] P. L. Roe. Characteristic-based schemes for the euler equations. *Ann. Rev. Fluid Mech.*, Vol. 18, p. 337, 1986.
- [118] G. D. van Albada, B. van Leer, and W. W. Rovers. A comparative study of computational methods in cosmic gas dynamics. *Astronomy and Astrophysics*, Vol. 108, p. 76, 1982.
- [119] D. J. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. *16th AIAA Computational Fluid Dynamics Conference*, p. 23, 2003.
- [120] M. S. Darwigh and F. Moukalled. Tvd schemes for unstructured grids. *Int. J. Heat Mass Transfer*, Vol. 46, No. 4, p. 599, 2003.
- [121] 肖鋒, 伊井仁志, 小野寺直幸. 計算流体力学. コロナ社, 2009.
- [122] V. Dyadechko and M. Shashkov. Reconstruction of multi-material interfaces from moment data. *J. Comput. Phys.*, Vol. 227, p. 5361, 2008.
- [123] H. Takewaki, A. Nishiguchi, and T. Yabe. Cubic-interpolated pseudo particle method (CIP) for solving hyperbolic equations. *J. Comput. Phys.*, Vol. 61, p. 261, 1985.
- [124] H. Takewaki and T. Yabe. The cubic-interpolated pseudo particle (CIP) method - application to nonlinear and multi-dimensional hyperbolic equations. *J. Comput. Phys.*, Vol. 70, p. 355, 1987.
- [125] T. Yabe and T. Aoki. A universal solver for hyperbolic-equations by cubic-polynomial interpolation 1. one-dimensional solver. *Comput. Phys. Commun.*, Vol. 66, p. 219, 1991.
- [126] T. Yabe, F. Xiao, and T. Utsumi. The constrained interpolation profile method for multi-phase analysis. *J. Comput. Phys.*, Vol. 169, p. 556, 2001.
- [127] R. Tanaka, T. Nakamura, and T. Yabe. Constructing exactly conservative scheme in a non-conservative form. *Comput. Phys. Commun.*, Vol. 126, p. 232, 2000.
- [128] X. Deng, B. Xie, and F. Xiao. Multimoment finite volume solver for euler equations on unstructured grids. *AIAA Journal*, Vol. 55, No. 8, p. 2617, 2017.
- [129] H Choi and P Moin. Effects of the computational time step on numerical solutions of turbulent flow. *J. Comput. Phys.*, Vol. 113, p. 1, 1994.
- [130] T. Yabe and P. Y. Wang. Unified numerical procedure for compressible and incompressible fluid. *J. Phys. Soc.*, Vol. 60, p. 2105, 1991.
- [131] C. M. Rhie and W. L. Chow. Numerical study of the turbulent flow part an airfoil with trailing edge separation. *AIAA Journal*, Vol. 21, No. 11, p. 1525, 1983.
- [132] H. Jasak. Error analysis and estimation for finite volume method with applications to fluid flow. *Citeseer*, 1996.
- [133] M. Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *J. Comput. Phys.*, Vol. 227, p. 2674, 2008.
- [134] U. Ghia, K. N. Ghia, and C. T. Shin. High-re solutions for incompressible flow using the

- navier-stokes equations and a multigrid method. *J. Comput. Phys.*, Vol. 48, p. 387, 1982.
- [135] J. C. Martin and W. J. Moyce. An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Mathematical and Physical Sciences*, Vol. 244, No. 882, p. 312, 2008.
- [136] A. Ikebata and F. Xiao. GPU-accelerated large-scale simulations of interfacial multiphase fluids for real-case applications. *Computers and Fluids*, Vol. 141, p. 235, 2016.
- [137] 池端昭夫, 肖鋒. 大規模並列 GPU 計算の衛生陶器多相流シミュレーションへの適用. 情報処理学会誌 コンピューティングシステム, Vol. 9, No. 4, p. 1, 2016.
- [138] 池端昭夫, 清水友哉, 肖鋒. 保存型 CIP 法による多相流の効率的な非構造格子数値計算モデル. Transaction of JSCES, 2018 年 1 月受理済み, 2018.



謝辞

本研究の推進にあたり、指導教員として、また長年の当社との共同研究責任者として、熱心なご指導と数々の貴重なご助言、さらに衛生陶器流体シミュレーション技術の進展に多大なるご貢献を頂きました肖鋒 教授に甚大なる感謝の意を表します。

本研究において、「CIP 法による多相流シミュレーション手法」に対する数々の貴重なご助言、また「並列 GPGPU 計算手法の適用」にあたり東京工業大学スーパーコンピュータ「TSUBAME」の産業利用へ導いて頂き、さらに利用過程で多くの貴重なご助言を頂きました青木尊之 教授に甚大なる感謝の意を表します。

さらには、この技術の核を成す「CIP 法」の発明をされ、私ども末端の数値計算技術者に論文、ご講演などを通して積極的に内容の公開をしていただき、CIP 法の研究着手時には数々の貴重なご助言や励ましの言葉を頂きました矢部孝 名誉教授に、この場をお借りしまして甚大なる感謝の意を表します。

またお忙しい中、審査員を引き受けて頂きました奥野喜裕 教授、末包哲也 教授、長崎孝夫 准教授に深く感謝いたします。

肖研究室におきまして、「非構造格子流体計算」に関して有意義なディスカッションをして頂きました OB の謝彬 氏に感謝申し上げます。また「気液界面捕獲手法」に関して有意義なディスカッションをして頂きました研究者の銭龍根 氏に感謝いたします。研究の推進および本論文の作成にあたり、ソフトウェアの使用方法、データの整理、文書の整形、文章のチェックなどの技術的な点を支援して頂きました OB の孫紫堯 氏、M2 の橋口翔 氏、M1 の清水友哉 氏、宮川直 氏に感謝いたします。

また慣れない東京での3年間の研究生活において、日常の交流を通して精神的な支えとなっていた頂きました肖研究室 OB の荻原一幾 氏、中山裕貴 氏、稲場智 氏にこの場を借りて深く感謝を申し上げます。

