



TOHOKU
UNIVERSITY

ISSN 2436-0066

東北大学
サイバーサイエンスセンター

大規模科学計算システム広報

SENAC

Vol.59 No.2 2026-4



Cyberscience
Center

Supercomputing System
Tohoku University

www.ss.cc.tohoku.ac.jp

大規模科学計算システム関連案内

<大規模科学計算システム関連業務は、サイバーサイエンスセンター本館内のデジタルサービス支援課が担当しています。>

<https://www.ss.cc.tohoku.ac.jp/>

階	係・室名	電話番号(内線)* e-mail	主なサービス内容	サービス時間
				平日
一階	利用相談室	022-795-6153 (6153)	計算機利用全般に関する相談	8:30～17:15
		相談員不在時 022-795-3406 (3406)	大判プリンタ、利用者端末等の利用	9:00～21:00
	利用者談話室	(3444)	自販機	8:30～21:00
	展示室(分散コンピュータ博物館)*	*見学希望の方はスーパーコンピューティングサポートユニットまでご連絡ください。	歴代の大型計算機等の展示	9:00～16:00
三階	総務係	022-795-3407 (3407) cc-som@grp.tohoku.ac.jp	総務に関すること	8:30～17:15
	会計係	022-795-3405 (3405) cc-kaikei@grp.tohoku.ac.jp	会計に関すること、負担金の請求に関すること	8:30～17:15
	スーパーコンピューティングサポートユニット	022-795-3406 (3406) cc-uketuke@grp.tohoku.ac.jp	利用手続き、利用相談、講習会、ライブラリ、見学、アプリケーションに関すること	8:30～17:15
		022-795-6252 (6252) cc-sys@grp.tohoku.ac.jp	共同研究、計算機システムに関すること	8:30～17:15
	デジタルプラットフォームユニット	022-795-6253 (6253) i-network@grp.tohoku.ac.jp	ネットワークに関すること	8:30～17:15
四階	研究開発部	022-795-6095 (6095)		
五階	端末機室	(3445)	PC 端末機(X 端末)	

* () 内は東北大学内のみの内線電話番号です。青葉山・川内地区以外からは頭に 92 を加えます。

本誌の名称「SENAC」の由来

昭和 33 年に東北地区の最初の電子計算機として、東北大学電気通信研究所において完成されたパラメロン式計算機の名前で SENAC-1 (SENdai Automatic Computer-1) からとって命名された。

[研究成果]

反応硬化分子動力学によるエポキシ硬化挙動の解析と材料設計への示唆

佐藤雅尚、鈴木謙太、鈴木貴也
 帝人株式会社 炭素繊維事業本部 技術開発部

本報告では、反応硬化分子動力学(MD)法を用いて、エポキシ樹脂 (DGEBA/33DDS 系) の硬化過程における昇温速度の違いが架橋構造および物性へ与える影響を体系的に評価した。スーパーコンピュータを用いて数万原子規模の大規模計算を行い、反応率の進行挙動、分子量分布、アミン消費挙動、ネットワーク形成機構を比較した結果、高速昇温では均一な構造が、低速昇温では局所的に密で粗密のある不均一構造が形成されることを見出した。得られた架橋モデルの物性解析から、低速昇温系では圧縮弾性率が高い一方、引張特性が低いなど、構造不均一性を反映した物性差が確認された。本解析は、硬化条件と構造・物性の相関理解に基づく高性能マトリクス樹脂設計に有用な知見を提供する。

1. はじめに

炭素繊維複合材料(CFRP)に使用するマトリクス樹脂は、エポキシ樹脂をはじめとする三次元架橋高分子がよく用いられる。三次元架橋高分子の物性は、架橋構造を構成するモノマーの化学構造に加えて、成形過程で生じた架橋構造が大きな影響を及ぼすことが知られている。一方で、架橋構造を有する高分子は、構造の複雑性に加えて、溶剤に不溶であることから高分子の構造解析手法として汎用的に用いられている核磁気共鳴(NMR)法やゲル浸透クロマトグラフィー(GPC)などを用いることが困難であり、構造と物性の相関に関しては不明点が多い。

近年、反応硬化分子動力学(MD)計算により、エポキシ樹脂の架橋構造形成過程および架橋構造と物性の相関関係を評価した研究が多く報告されている¹⁻⁴。反応硬化MDを用いることにより、所望の化学構造・硬化条件(硬化温度や昇温速度など)でエポキシ樹脂架橋構造を作成することが可能であり、その構造・物性を評価することにより、高性能なマトリクス樹脂開発に必要な系統的な知見を獲得することが可能である。

本報告では、東北大学の岡部らが報告している反応硬化MD解析手法を用い、計算はスーパーコンピュータAOBA上で実施した。スーパーコンピュータを使用することにより、原子数数万規模の大規模な計算が可能となり、汎用PCを用いた小スケールの解析では評価が困難であった架橋構造の不均一性などの評価が可能となった。以下にその一例を示す。

2. エポキシ樹脂の硬化過程における昇温速度が架橋構造に及ぼす影響の評価

2.1 解析条件

本検討では、汎用のエポキシ樹脂および硬化剤である、2,2-bis(4-glycidyloxyphenyl)propane (DGEBA) および 3,3-diamino diphenyl sulfone (33DDS)を用い、エポキシ:アミンの等量比を1:1とし、DGEBAが720分子、33DDSが360分子、計45720原子系で計算を実施した。用いた分子の化学構造を図1に示す。初めに、1080分子を含む初期構造を作成し、同一構造からスタートして、0.75 °C/fs (低速昇温系)、4.5 °C/fs (高速昇温系)の異なる2条件の昇温速度で昇温し、180 °Cで硬化する条件で解析を実施した。表1に解析条件を示す。

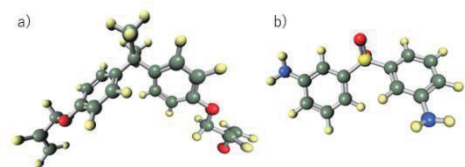


図1 a)DGEBA、b)33DDSの化学構造

表1 MD解析条件まとめ

分子数		総原子数	昇温速度	開始温度	保持温度	保持時間	総サイクル
DGEBA	33DDS		fs/K	K	K	fs	step
720	360	45720	4.5	273.15	453.15	120	161
			0.75	273.15	453.15	120	328

2.2 解析結果

2.2.1 硬化過程の評価

図2にMD解析の計算ステップとエポキシ基の反応率(以下、反応率)および系中温度の関係を示す。図2の反応率に関して、どちらの系も最終的な反応率が80%程度と、反応が同程度進行していることがわかる。次に、ステップ数に対する反応率の増加挙動に着目すると、高速昇温系では、反応系温度が180℃に到達後、50 cycle 付近で反応率の上昇が鈍化しているのに対し、低速昇温系では、昇温過程の170 cycle 付近(150℃付近)で反応率が鈍化した。反応系温度は、分子の運動性・拡散挙動に影響することから、高速昇温系は、低速昇温系と比較して分子運動性が高い状態で反応が進行していることが示唆される。

次に、架橋構造の形成過程を評価した。図3に反応率に対して、反応系中における最大分子の分子量の関係を示す。共通の傾向として、反応率50%程度までは分子量が緩やかに増加し、反応率50~60%付近で分子量が急激に増加、60%以降は分子量が徐々に増加する挙動が観測された。分子量が50~60%程度の急激に分子が成長した領域において、最大分子の重量分率が系全体の50%以上を占め、この近辺でゲル化が生じていると考えられる。この近辺の分子量の増加挙動を比較すると、高速昇温系は分子量が連続的に増加しているのに対して、低速昇温系は分子量が不連続に立ち上がるような挙動が見られる。前述したように、低速昇温系では分子の運動性が低い状態で反応が進行しているため、不均一に分子が成長していると考えられる。図4に、反応率40%、60%における分子量(対数表記)のヒストグラムを示す。高速昇温系では、反応率40%では、 $x=3.0-3.5$ 付近の中程度の分子量の分子が増加し、反応率60%においても同程度の分布を維持した状態で分子が成長している挙動が見られ、反応が均一に進行していると考えられる。一方、低速昇温系では、未反応物($X \leq 2.6$)の割合が高速昇温系と比較して高いにも関わらず、 $x=3.0-3.5$ 付近の中程度の分子量の分子割合が高い挙動を示した。また、反応率60%において、中程度の分子量の分子割合が急激に減少し、分子量が大きな分子が増加していることがわかる。

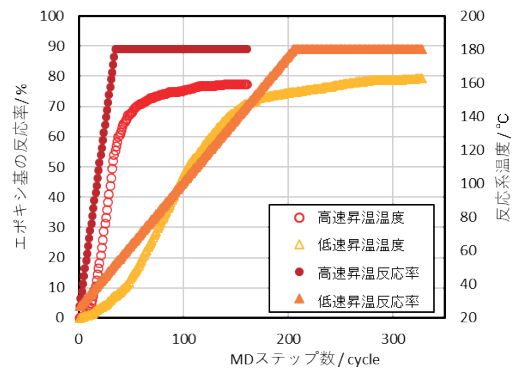


図2 MD解析における、反応温度および反応率の推移

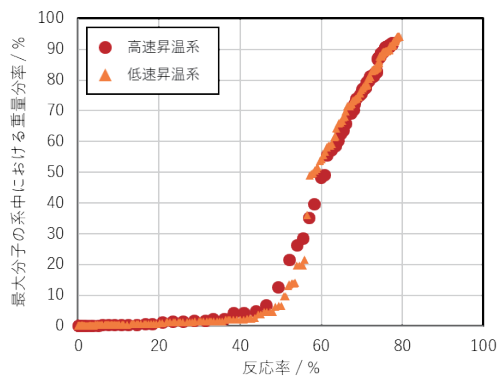


図3 反応率に対する、系中最大分子の反応系内における重量分率の推移

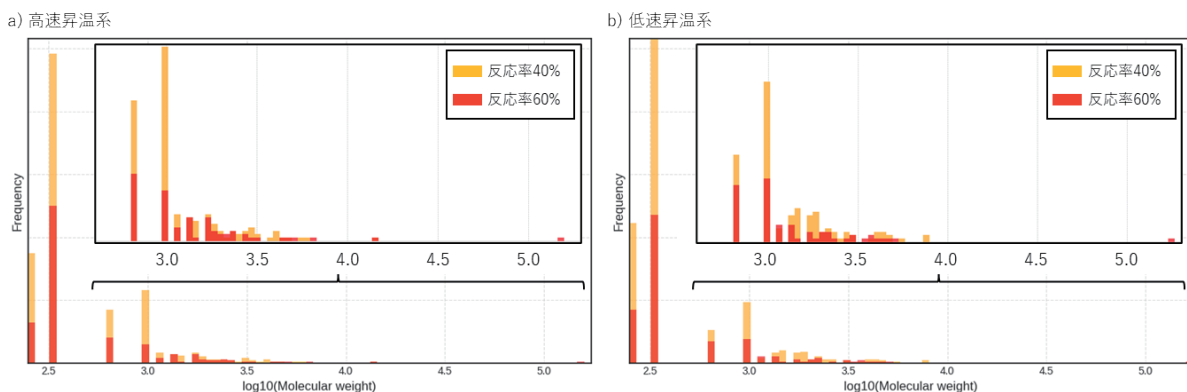


図4 a)高速昇温系、b)低速昇温系の反応率40%、60%における分子量の分布

以上の結果から、高速昇温系では、分子が均一に成長して徐々にネットワークを形成している一方、低速昇温系では、局所的に成長したオリゴマーが連結することで急激にネットワークを形成していることがわかる。

形成したネットワーク構造に関して、詳細に評価するために、硬化点(アミンの活性水素)の消費挙動を評価した。図 5a, b にアミンの反応模式図とエポキシ反応率に対する 1 級、2 級、3 級アミンの分率を示す。図 5a に示すように、1 級→2 級への反応は分子鎖の延長を、2 級→3 級への反応は鎖の分岐を示し、3 級アミンが多い構造程緻密な構造を形成している傾向がある。また、アミンの消費は一方通行であるため、1 級アミンは単調減少、3 級アミンは単調増加し、2 級アミンは 1 級→2 級、2 級→3 級それぞれの反応の速度比によ

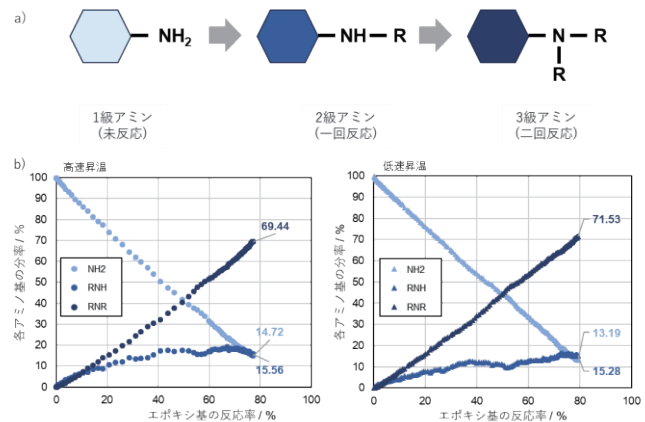


図5 a) アミンの反応模式図、b) エポキシ基の反応率の増加に対する 1 級(NH2)、2 級(RNH)、3 級(RNR)アミンの分率の変化

って挙動が変化するため、2 級アミンの分率の変化から架橋構造の形成過程を予測できる。図 5b において 2 つの系を比較してみると、最終生成物の各アミンの分率に大きな差がない一方、反応途中、特にゲル化近辺である、反応率 40%~50%付近における 2 級アミンの分率が異なっていることがわかる。この違いは最終的に形成される架橋構造が大きく異なっていることを示唆している。高速昇温系は、反応中期において 2 級アミンの割合が 20%程度まで増加後、徐々に減少しており、ある程度分子鎖が成長すると同時に分岐構造を形成していくと推察される。一方で、低速昇温系では、反応中期の 2 級アミンの増加は 10%程度にとどまり、その後、緩やかに増加していく挙動が見える。これは、1 級アミンの反応によって生じた 2 級アミンが直ちに反応し、3 級アミンに変化していることを示唆しており、分岐構造の形成によって架橋構造が発達していることを示唆している。低速反応系では、前述したように系中の分子の運動性が低く、反応率 40%近辺では局所的にオリゴマーが生成・成長する。ここで、アミンの消費挙動を加味すると、形成したオリゴマーは分岐が多く、密に架橋した構造を形成していることが示唆される。

これまでの結果から推察される昇温速度の違いによる架橋構造の形成過程の違いのイメージを図 6 にまとめる。高速昇温系では、反応率 40%近辺まで反応系中で分子鎖の延長と分岐が同時に起こり、反応率 60%近辺(ゲル化)均一に後も均一な架橋構造が発達していると考えられる。一方で、低速昇温系では、反応率 40%近辺まで分子が成長していく過程における温度の低さ(≒分子運動性の低さ)から、局所的に発生した分子が分岐構造を主として密な構造を形成する。その後、系の温度が十分に上昇した段階で、密な構造同士が緩やかに連結することで、粗密のある不均一な構造が形成されると考えられる。

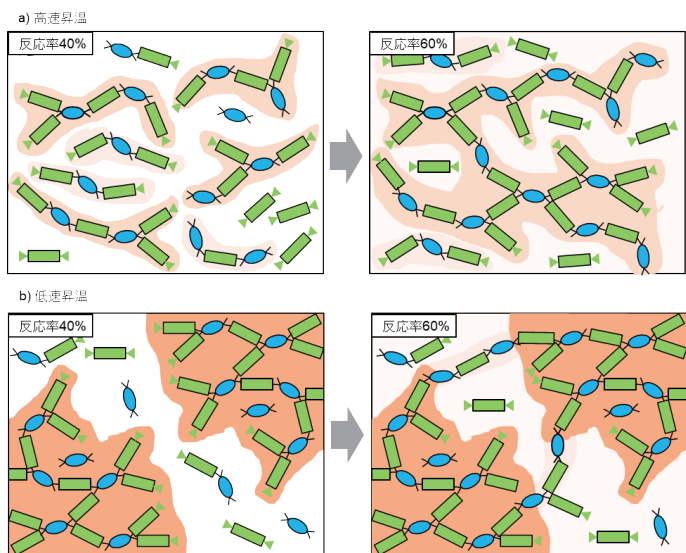


図6 a) 高速昇温系、b) 低速昇温系におけるネットワークの形成イメージ

表2 硬化後物性まとめ

	体積			自由体積分率		CTE			3軸引張	1軸圧縮		
	硬化前	硬化後	硬化収縮	反応後	緩和後	Tg	α glass	α rubber	弾性率	弾性率	降伏歪	降伏応力
	nm ³		%	%		°C	1/K		GPa	GPa	%	MPa
高速昇温系	525.59	508.49	3.253	0.366	0.082	230.5	6.05E-05	2.43E-04	4.143	4.354	0.153	324.5
低速昇温系	525.59	508.00	3.348	0.343	0.095	207.1	6.03E-05	2.11E-04	3.993	4.696	0.151	330.5

2.2.2 力学物性評価

反応硬化 MD によって得られた架橋モデルに対し、構造最適化・構造緩和を実施した後、熱物性、力学物性の解析を実施した。解析結果を表 2 に示す。

初めに、硬化反応の前後における体積の変化から、硬化収縮を計算した。その結果、高速昇温系と比較して、低速昇温系の硬化収縮が僅かではあるが大きい値を示した。これは低速昇温系の方が反応過程において密な構造を形成していることを示唆している。また、反応硬化 MD 直後と構造緩和を行った後の架橋モデル内の自由体積分率を比較すると、反応硬化 MD 直後では低速昇温系内の自由体積が小さい値を示したが、構造緩和後ではその関係が逆転した(図 7)。この結果は、低速昇温系において不均一な架橋構造が形成されていることを示唆するものである。

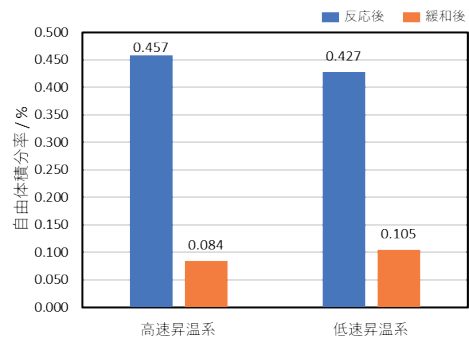


図7 構造緩和前後の自由体積分率変化

熱膨張率 (CTE) 解析から、それぞれの系のガラス転移温度 (Tg) および Tg 前後における線膨張係数 (α) を評価した。Tg に関して、低速昇温系が高速昇温系と比較して低い値を示した。また、ガラス状態の線膨張係数 (α glass) はどちらの系も同程度の値を示した一方で、ゴム状態の線膨張係数 (α rubber) は高速昇温系よりも低速昇温系の方が顕著に低い値を示した。以上の結果は、低速昇温系の系内に緻密な構造を有しているとともに、架橋構造に粗密が生じていることを支持するものである。

力学物性は、1軸圧縮試験および3軸引張試験の解析により行った(図 8)。1軸圧縮試験で得られた弾性率 (E) および降伏応力は、低速昇温系の方が高速昇温系と比較して大きな値を示した。一方で、3軸引張試験においては、高速昇温系の方が低速昇温系と比較して大きな値を示した。低速昇温系は、架橋構造内に局所的に密な(硬い)部分を有しており、圧縮過程において高い弾性率・降伏応力を示したと考えられる。一方、引張過程においては、架橋構造が疎な部分が容易に変形することにより、低い弾性率を示したと考えられる。また、降伏ひずみを比較すると、低速昇温系の方が低い値を示した。また、降伏後のひずみ軟化挙動に着目すると、構造全体として考えた場合、剛直だが脆い構造であることが示唆される。この結果は、定性的ではあるが実験により得られた値と同様の傾向を示した。

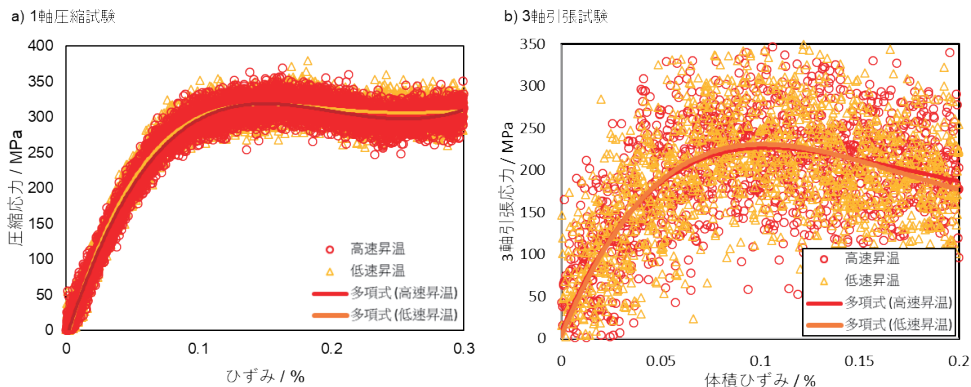


図8 a) 1軸圧縮試験、b) 3軸引張試験のS-Sカーブ

3. まとめと今後の展望

今回の解析では、分子の初期配置、等量が完全一致した系で、異なる硬化条件で反応を行い、得られた架橋構造および物性の違いを比較した。最終的に得られた架橋高分子のエポキシ基反応率、1級、2級、3級アミンの存在比に大きく差はなかったが、その構造の形成過程において顕著な差が生じ、それによって物性が大きく変化することを見出した。

近年、成型方法の多様化、製品形状の複雑化により、材料物性だけでなく多様な硬化条件への適合性も求められることから、樹脂の硬化メカニズムや物性と構造の相関関係などの基礎的な知見が、高効率・迅速な材料開発には不可欠である。本解析により、硬化条件による材料物性の変化を系統的にとらえることができれば、実際の製品製造過程で生じるトラブルを事前に把握することも可能であり、開発コスト低減に寄与すると考えられる。

また、解析に用いる分子の化学構造(置換基の有無や反応点の結合位置など)を変化させて解析を行った結果、分子構造の違いに起因して架橋構造の形成挙動が変化する様子も観測されている(現在、詳細な解析を実施中)。これらの知見は、高性能樹脂の分子設計の指針を与えるものになると予想される。

謝辞

本研究は、東北大学サイバーサイエンスセンターのスーパーコンピュータを利用して得られた成果である。研究の遂行にあたり、同センター関係各位には有益なご指導ならびに多大なご協力を賜った。ここに深く感謝の意を表する。また、本研究の推進にあたっては、東北大学 グリーン未来創造機構 グリーンクロステック研究センター 川越吉晃准教授より、貴重なご助言とご指導をいただいた。ここに記して謝意を表する。

なお、本研究の一部は、内閣府総合科学技術・イノベーション会議の戦略的イノベーション創造プログラム (SIP) 「サーキュラーエコノミーシステムの構築」(研究推進法人: 独立行政法人環境再生保全機構) (JPJ012290) の支援を受けて実施したものである。

参考文献

- [1] Okabe et al, Curing reaction of epoxy resin composed of mixed base resin and curing agent: Experiments and molecular simulation, *Polymer*, 54(17), 4660-4668, 2013
- [2] Shirasu et al, Amine/epoxy stoichiometric ratio dependence of crosslinked structure and ductility in amine-cured epoxy thermosetting resins, *J. Appl. Polym. Sci.*, 138(23), 50542-50542, 2021
- [3] Kawagoe, Kikugawa et al., Thermoset resin curing simulation using quantum-chemical reaction path calculation and dissipative particle dynamics, *Soft Matter*, 17(28), 6707-6717, 2021
- [4] Kawagoe et al, Uncovering the Mechanism of Size Effect on the Thermomechanical Properties of Highly Cross-Linked Epoxy Resins, *J. Phys. Chem. B*, 126(13), 2593-2607, 2022

[共同研究成果]

有機超塩基触媒を用いる炭素-酸素結合変換

— フォスファゼン塩基触媒によるヒドロキシベンジルアルコールの変換反応 —

笹本 大空¹、是永 敏伸²、重野 真徳^{1,3}¹東北大学大学院薬学研究科、²岩手大学理工学部化学・生命理工学科、³JST さきがけ

本研究ではフォスファゼン塩基 *t*-Bu-P2 を用いたヒドロキシベンジルアルコール類のベンジル位水酸基の触媒的変換反応を対象として、反応機構の理論的考察を実施した。

1. はじめに

ヒドロキシベンジルアルコールおよびその類縁化合物は、アドレナリンやバンコマイシンをはじめとする生物活性分子に広く分布している(図1)。そのため、これらの化合物群を標的とする変換反応の開発は創薬化学分野において重要な研究課題に位置付けられる。現在までに報告されている反応では、ヒドロキシベンジルアルコールを出発物質として高反応性のキノンメチド中間体を経由する方法が主である。これらの反応は、水酸基の低い脱離能を補うためにルイス酸やブレンステッド酸を用いる酸性条件が必要であった。一方で、中性/塩基性条件下での反応はほとんど報告されておらず、基質適用範囲や官能基許容性の観点から従来法に代わる新しい反応の開発が望まれている。^{1,2}

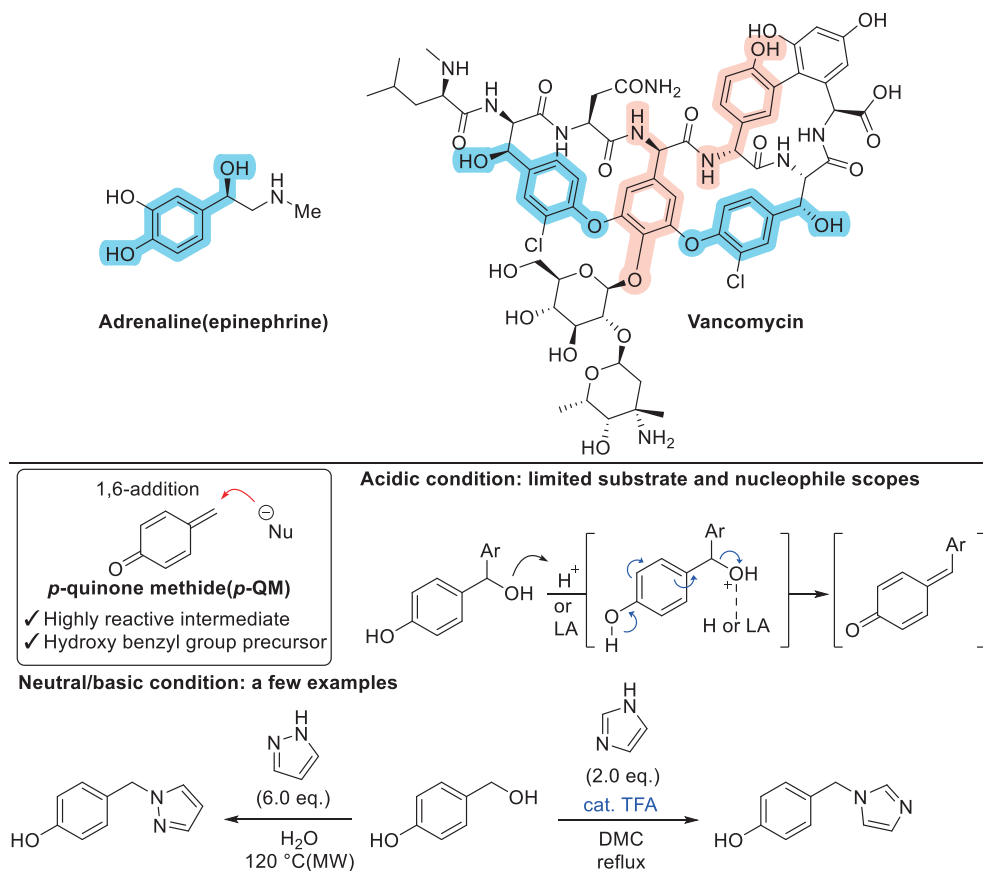


図1 ヒドロキシベンジルアルコール部位を有する生物活性分子とキノンメチドの発生法

フォスファゼン($R_3P=NR'$)塩基は、共役系が広がった化学構造を有し、それに起因する強力なブレンステッド塩基性(超塩基性とも呼ばれる)および対アニオン活性化効果を示す。これらの特徴を活用することで、様々な分子変換反応を実現してきた。³⁻⁷ 最近、種々の求核剤存在下、ヒドロキシベンジルアルコールに対してフォスファゼン塩基 *t*-Bu-P2 を作用させると、ベンジル位水酸基の置換反応が円滑に進行することを見出した(図 2)。⁸ そこで、今回、本置換反応の反応機構を明らかにするべく、密度汎関数理論(DFT)を用いた計算を実施した。

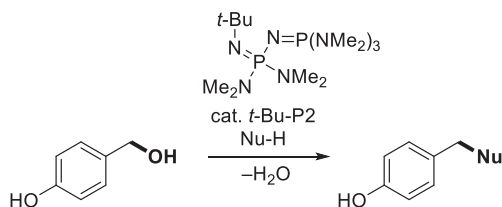


図 2 ヒドロキシベンジルアルコール類の触媒的置換反応

2. 方法・結果 : DFT 計算に基づく反応機構の考察

本研究では、東北大学サイバーサイエンスセンターの Gaussian 16 (revision C.01 および revision C.02)パッケージソフトを利用して DFT 計算を実施した。

対象とする反応の推定反応機構を図 3 に示す。まず、*p*-ヒドロキシベンジルアルコール(1)の脱プロトン化が進行し、続いてベンジル位水酸基の脱離によってキノンメチドが生成すると考えられる。ここで、水酸基は脱離能が低いいため、ヒドロキシドとして脱離するとは考えにくい。そのため、本反応では、反応系内の酸性プロトンによる活性化を伴って脱離すると予想される。その後、反応系中に存在する求核剤 2 によって目的物 3 が生成する。

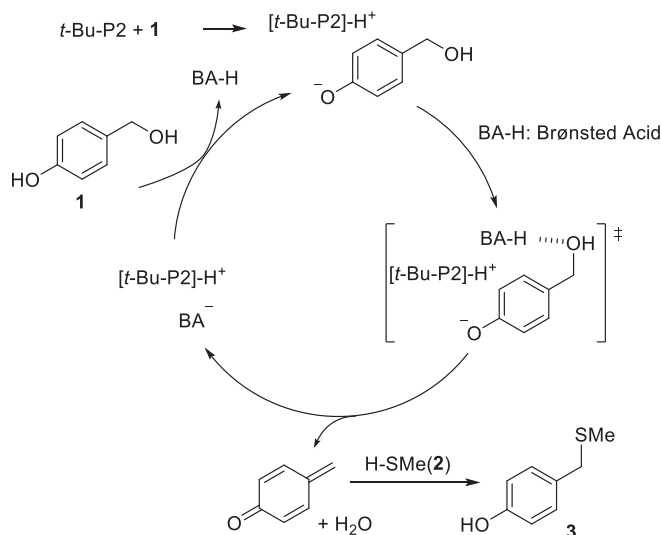


図 3 本触媒反応における推定される反応機構

本仮説に基づき、対カチオンによる活性化を伴う経路の遷移状態探索と反応のエネルギープロファイルを実施した(図 4)。なお、遷移状態の探索および構造最適化には ω B97X-D/6-311+G(2d,p)の組み合わせを用い、溶媒効果として CPCM 法(1,4-dioxane)を採用した。本反応の律速段階はキノンメチドの生成過程(Int-2 \rightarrow TS-1 \rightarrow Int-3)であり、その活性化障壁は 29.8 kcal/mol と比較的高い結果となった。なお、この過程は可逆反応であるが、続くチオール 2 の

付加反応によって安定な目的物へと変換される。

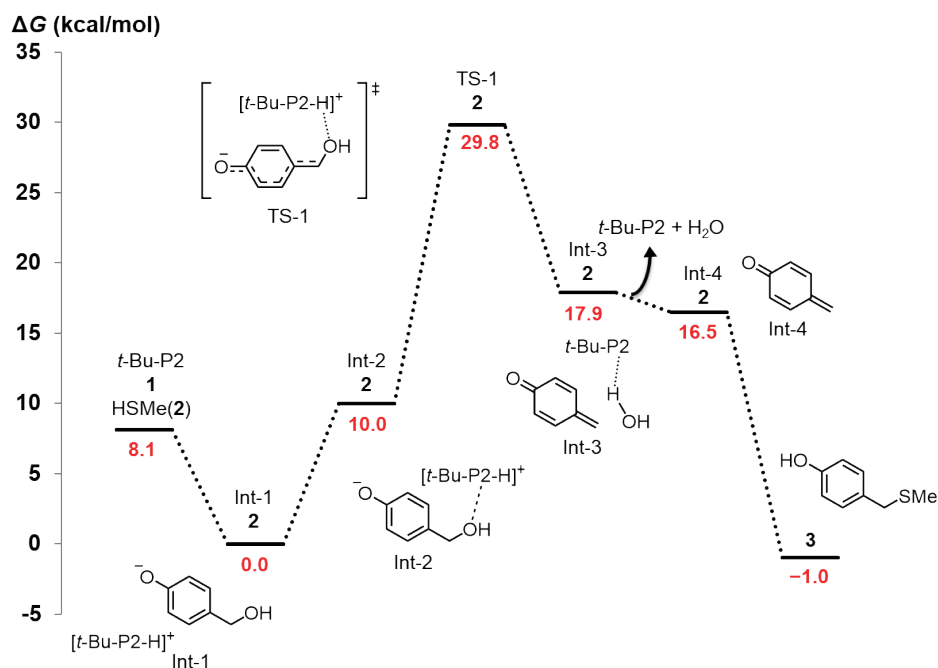


図4 対カチオンが水酸基を活性化する反応経路

次に、反応系中に存在するブレンステッド酸、具体的にはフェノール部位や求核剤のチオール部位が遷移状態に関与する経路を探索した(図 5,6)。出発物質である **1** が関与する反応の活性化障壁は、最も低い 26.9 kcal/mol であった一方で、メタンチオール(**2**)が関与する反応では 32.9 kcal/mol の最も高い活性化障壁を与えた。この差は、脱離する水酸基部位との水素結合の強さに起因すると考えられる。すなわち、電気陰性度に基づく O-H(酸素-水素)結合の方が S-H(硫黄-水素)結合よりも分極しているため、チオールよりもフェノールの水素結合ドナー性が高いと考えられる。したがって、本反応の円滑な進行にはフェノール性水酸基の水素結合が重要な役割を果たすことが示唆された。実際、分子間水素結合を一部阻害する極性非プロトン性溶媒中で反応を実施すると、反応の効率が低下する傾向にあった(図 7)。

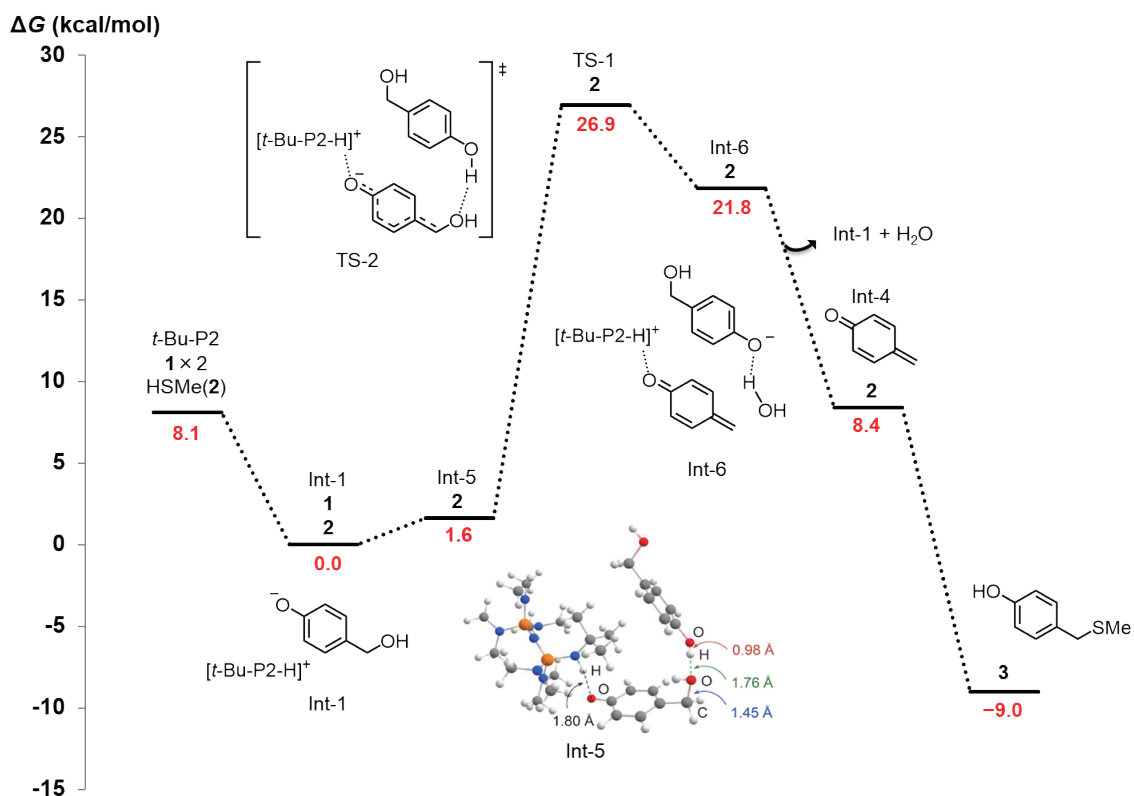


図5 出発原料1が水酸基を活性化する反応経路

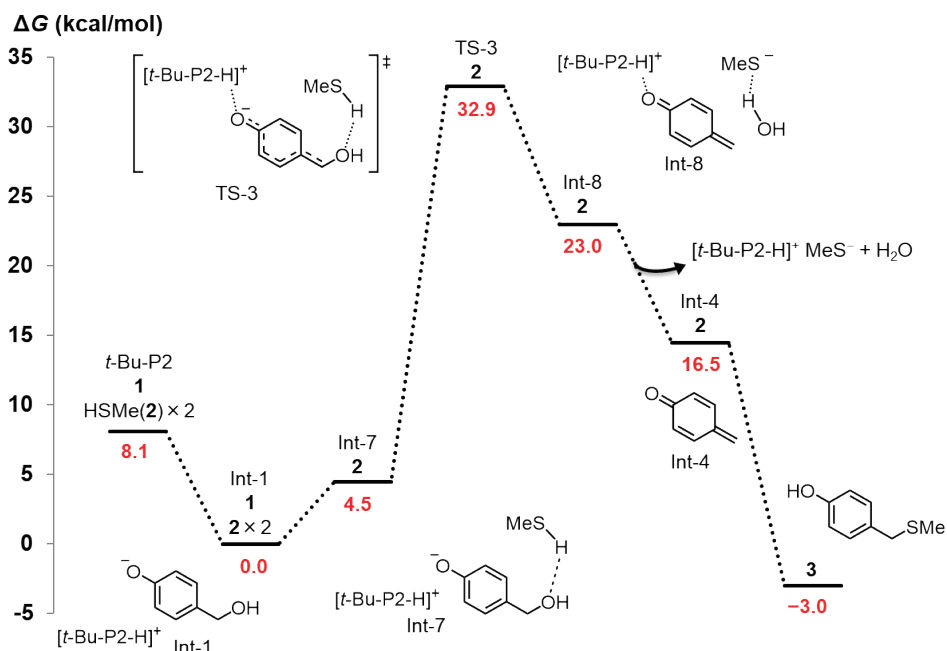


図6 求核剤2が水酸基を活性化する反応経路

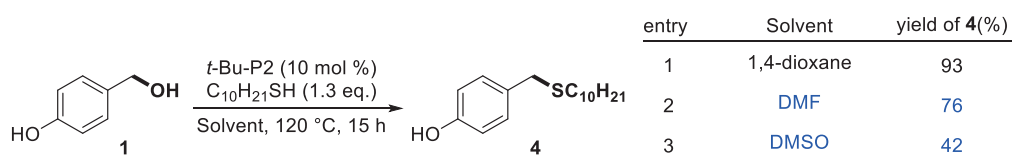


図7 反応溶媒による目的化合物4の収率変化

3. おわりに

本稿では DFT 計算に基づいてフォスファゼン塩基 *t*-Bu-P2 を用いたヒドロキシベンジルアルコール類の変換反応について、遷移状態探索とエネルギープロファイルを実施した。反応の円滑な進行には出発原料のフェノール性水酸基の水素結合が重要であることが示唆された。

謝辞

本研究は、東北大学サイバーサイエンスセンターのスーパーコンピュータを利用することで実現することができた。また、研究にあたっては同センター関係各位に有益なご指導とご協力をいただくと共に、令和6年度若手・女性研究者支援課題として利用負担金の一部をご支援いただいた。この場をお借りして御礼申し上げます。

参考文献

- [1] S. González-Pelayo, L. A. López. Microwave-Assisted Generation and Capture by Azoles of *ortho*-Quinone Methide Intermediates under Aqueous Conditions, *Eur. J. Org. Chem.*, 6003–6007, 2017.
- [2] A. S. Myshko, *et al.* Trapping of thermally generated *ortho*- and *para*-quinone methides by imidazoles and pyrazoles: a simple route to green synthesis of benzopyrone-azole hybrids and their evaluation as α -glucosidase inhibitors, *RSC Adv.*, 14, 27809–27815, 2024.
- [3] M. Shigeno, *et al.*, Phosphazene Base *t*Bu-P4 Catalyzed Methoxy–Alkoxy Exchange Reaction on (Hetero)Arenes, *Chem. Eur. J.*, 25, 6077–6081, 2019.
- [4] M. Shigeno, *et al.*, Organic Superbase *t*-Bu-P4 Catalyzes Amination of Methoxy(hetero)arenes, *Org. Lett.*, 21, 5505–5508, 2019.
- [5] M. Shigeno, *et al.*, Catalytic Amination of β -(Hetero)arylethyl Ethers by Phosphazene Base *t*-Bu-P4, *Org. Lett.*, 21, 6695–6699, 2019.
- [6] M. Shigeno, *et al.*, Organic superbase *t*-Bu-P4-catalyzed demethylations of methoxyarenes, *Org. Chem. Front.*, 9, 3656–3663, 2022.
- [7] M. Shigeno, *et al.*, Catalytic Concerted S_NAr Reactions of Fluoroarenes by an Organic Superbase, *J. Am. Chem. Soc.*, 146, 32452–32462, 2024.
- [8] O. Sasamoto, *et al.*, Phosphazene Base *t*-Bu-P2-Catalyzed Substitution Reactions of *para*-Hydroxybenzyl Alcohols, *Asian J. Org. Chem.*, 14, e00407, 2025.

[共同研究成果]

分子動力学計算による抗 iPS 細胞抗体と

糖鎖リガンドの相互作用様式の同定

大野 詩歩 (東北医科薬科大学 分子生体膜研究所)

抗体と糖鎖の相互作用を明らかにするための一般的な手法として X 線結晶構造解析があるが、糖鎖は結晶化が困難であることなどの問題点がある。そこで本研究では、未分化 iPS 細胞抗体-糖鎖複合体モデルの構築および相互作用解析による抗体の糖鎖認識機構の解明を目的として、分子動力学計算を実施した。

1. 序論

抗体のリガンド結合様式を原子レベルで解析する手法として、X 線結晶構造解析が広く用いられている。しかし、糖鎖は電子密度が低く、像を得ることが困難な傾向にある。近年、計算化学の分野は急速に発展しており、その代表例として AI を用いたタンパク質モデリングシステムである AlphaFold が挙げられる。しかし、AlphaFold では、学習データとなる糖鎖の結晶構造が限られているなど、未だ発展途上であり、糖鎖を含むタンパク質の構造予測精度は低い。また、抗体や糖鎖の動的挙動を調べるにはモデル構築だけでなく、分子動力学 (MD) 計算が不可欠である。本研究では、抗糖鎖抗体とリガンド糖鎖の複合体モデルを構築し、その結合様式を計算化学と NMR を組み合わせて、解明した。

2. 方法

抗体-糖鎖複合体の構築を行うため、リガンド糖鎖として lacto-*N*-fucopentaose I (LNFP I) (Fuc α (1-2)Gal β (1-3)GlcNAc β (1-3)Gal β (1-4)Glc)、抗体は R-17F および R-13E を用いた [1]。ドッキングモデルの構築にあたり、PDB に登録されている LNFP I の結晶構造を抽出し、グリコシド結合まわりの二面角の傾向を抽出し、安定なコンホメーションを調査した。R-17F および R-13E 抗体の Fv 領域の立体構造モデルは、AlphaFold2 により作成した [2]。CDR は IMGT により定義した。リガンドは結晶構造から抽出した (PDB ID: 5VX5)。作成した抗体に対し、構造最適化のため MD シミュレーションを行った。MD シミュレーションは、力場に CHARMM-GUI を用い、溶媒モデルは TIP3P を用いた [3]。計算ソフトには GENESIS 1.7.0 [4, 5] を用いた。Steepest Descent 法を用いて 10,000 ステップのエネルギー最小化を実施し、100 ps の間昇温し、nVT アンサンブル下で平衡化を行った。抗体のみのシミュレーションでは、この昇温過程において重原子を固定した。その後、nPT アンサンブル下で平衡化を実施した。この段階の初めの 100 ps の間は重原子を固定し、その後さらに 2000 ps にわたり主鎖原子および C α 原子を固定した。最終的に 300 K、タイムステップ 2 fs の条件下で nPT アンサンブルにおいて 100 ns の平衡化を実行した。シミュレーションのボックスサイズは、糖鎖-抗体複合体に対しては 78-83 Å \times 78-83 Å \times 78-83 Å とし、糖鎖のみのシミュレーションでは 60 Å \times 60 Å \times 60 Å とした。複合体のエピトープを実験的に同定するため、化学合成した LNFP I と R-17F あるいは R-13E の混合溶液に対して、STD-NMR を測定した。LNFP I の NMR 信号の帰属には 1D- ^1H 、2D DQF-COSY、CLIP-COSY、HOHAHA、NOESY、1D- ^{13}C 、 ^1H - ^{13}C HSQC、 ^1H - ^{13}C HMBC を用いた。これらの結果を踏まえ、LNFP I と R-17F および R-13E 抗体とのドッキングシミュレーションを行った。ドッキングシミュレーションは、HADDOCK2.4 を用い STD-NMR とデータベース解析の結果を組み込んで行った [6, 7]。得られたドッキングポーズのクラスタリングを行い、妥当と思われる複合体構造

の最適化のために MD シミュレーションを実施した。

3. 結果・考察

PDB に登録されている LNFP I のグリコシド結合まわりの各二面角を求めたところ、その分布は非常に限局されていたことから、LNFP I の構造は柔軟性が乏しいことが推察された(図 1)。

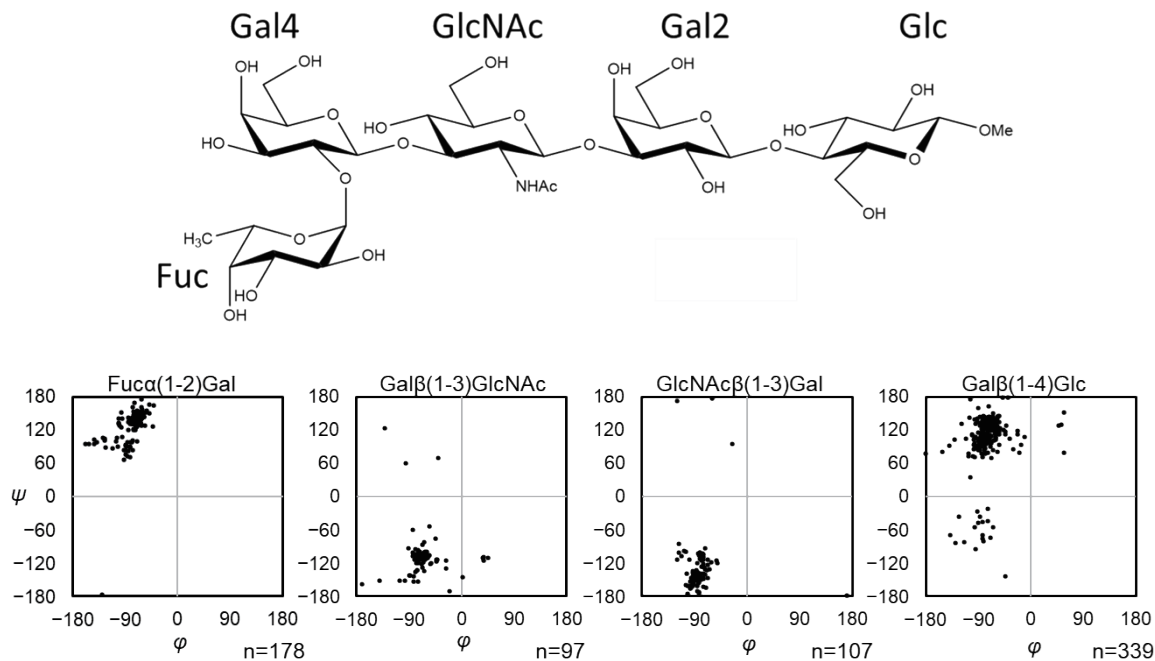


図 1. PDB に登録されている LNFP I の二面角の分布。六員環の単糖 (i) と単糖 (i-1) が 1-4 結合でつながっている場合、 ϕ (O5-C1(i)-O4(i-1)-C4(i-1))、 ψ (C1(i)-O4(i-1)-C4(i-1)-C3(i-1)) と定義した。

2次元 NMR により LNFP I の NMR シグナルの帰属を行い、各抗体と LNFP I の STD-NMR の解析を行った。その結果、R-17F および R-13E はいずれも LNFP I の非還元末端側を共通のエピトープとして認識していることが示唆された。また、STD-NMR のシグナルは R-13E の方が R-17F よりも強かったため、R-13E と LNFP I の結合・解離は R-17F の場合と比べて速いことが示唆された (図 2)。

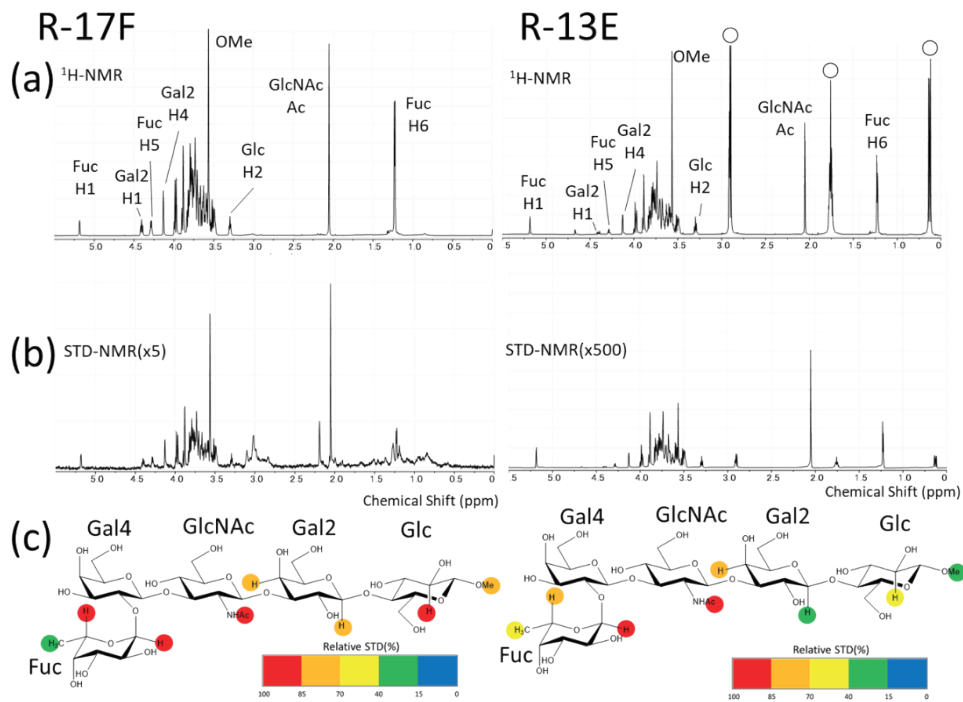


図 2. R-17F または R-13E と LNFP I の STD-NMR 解析。(a) リガンド糖鎖のみで測定したスペクトル。○は化学シフトの基準物質 DSS に由来するシグナル。(b) STD-NMR 測定で得た差スペクトル。(c) 各プロトンの強度比。強度比は(差スペクトルの強度)/(コントロールスペクトルの強度)として算出した。強度比が一番大きい GlcNAc の Ac プロトンを 100% とし、色分けを行った。STD-NMR のシグナル強度は R-17F においてはコントロールの 5 倍、R-13E においてはコントロールの 500 倍にした。

LNFP I-抗体複合体モデルを作成するため、ドッキングシミュレーションを行った。さらに糖鎖は多様なコンホメーションを取り得るため、静的構造解析のみでは認識機構の理解は不十分である。そこで、得られた複合体の最適化のため、100 ns の MD シミュレーションを行った。その結果、R-17F-LNFP I 複合体は結合を維持していたが、R-13E-LNFP I 複合体は 100 ns のシミュレーション時間の間、結合を維持することができなかった(図 3)。

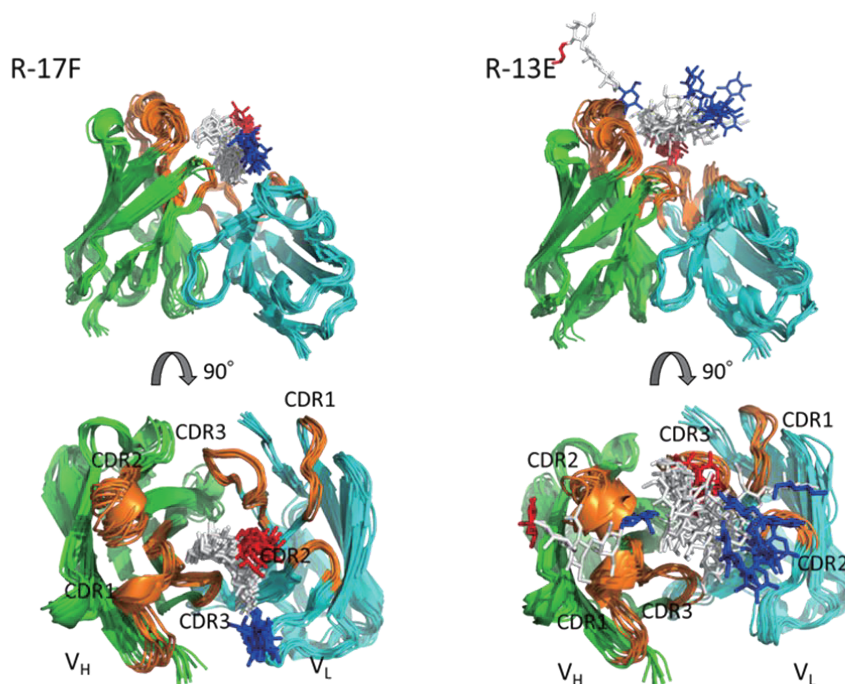


図 3. MD シミュレーション(100 ns)における複合体の構造。各スナップショットは 10 ns ごとに抽出した構造である。緑は H 鎖、水色は L 鎖、橙色は CDR、赤は Fuc、青は Glc を示す。

H 鎖と L 鎖のどちらが結合安定性に寄与しているかを評価するため、(1) H 鎖は R-17F の配列であり、L 鎖は R-13E の配列を有するモデルと、(2) H 鎖は R-13E の配列であり、L 鎖は R-17F の配列を有するモデルの 2 つに対し MD シミュレーションを行ったところ、(2) のモデルのみ結合を維持していた。したがって結合には R-17F の L 鎖のアミノ酸残基がリガンドとの結合に重要であることが示された。

次に、遊離型および複合体状態の抗体に対する分子動力学シミュレーションにおいて、R-13E および R-17F のアミノ酸の RMSF 値を計算することにより、LNFP I 複合体における抗体の動的評価を行った。遊離型と比較して、R-13E の L 鎖の CDR3 の RMSF 値は LNFP I への結合に伴い減少した。対照的に、遊離型と複合体状態の R-17F では、この領域の RMSF 値に有意な差は認められなかった(図 4)。

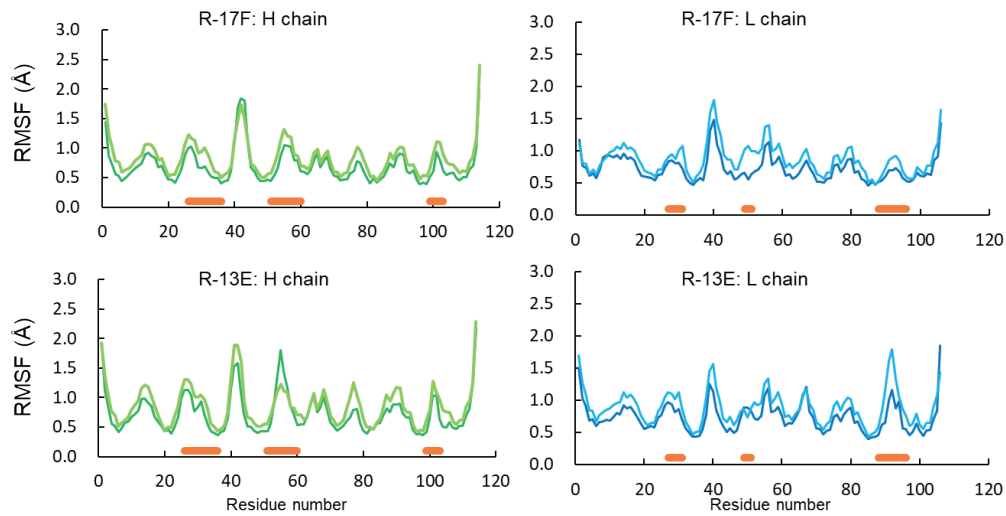


図 4. 分子動力学シミュレーション中における抗体残基の RMSF 解析。

上は R-17F、下は R-13E について、アミノ酸残基単位で算出した RMSF プロットを示す。RMSF 値は、R-17F では 100 ns、R-13E では 50 ns のシミュレーションを用いて計算した。抗体単独状態の RMSF 値は、R-17F を黄緑色、R-13E を水色で示し、抗体-リガンド複合体の RMSF 値は、それぞれ緑色および青色で示した。RMSF 値は、独立した 3 回のシミュレーション (run 1-3) の平均値である。相補性決定領域 (CDR) はオレンジ色で示した。

R-13E および R-17F の糖鎖結合部位をアミノ酸残基レベルで明らかにするため、抗体のアミノ酸の溶媒露出度 (SASA) を算出し、糖鎖との接触部位を評価した。その結果、R-17F は CDR 領域を介して LNFP I と広範囲に接触していることを示した。対照的に、R-13E は CDR 領域と LNFP I との相互作用が R-17F と比較して少ないことが明らかになった。特に、L 鎖の CDR2 と LNFP I との相互作用はごくわずかであった。次に、MD 計算中の L 鎖の CDR2 および CDR3 領域の挙動を検討した (図 5)。

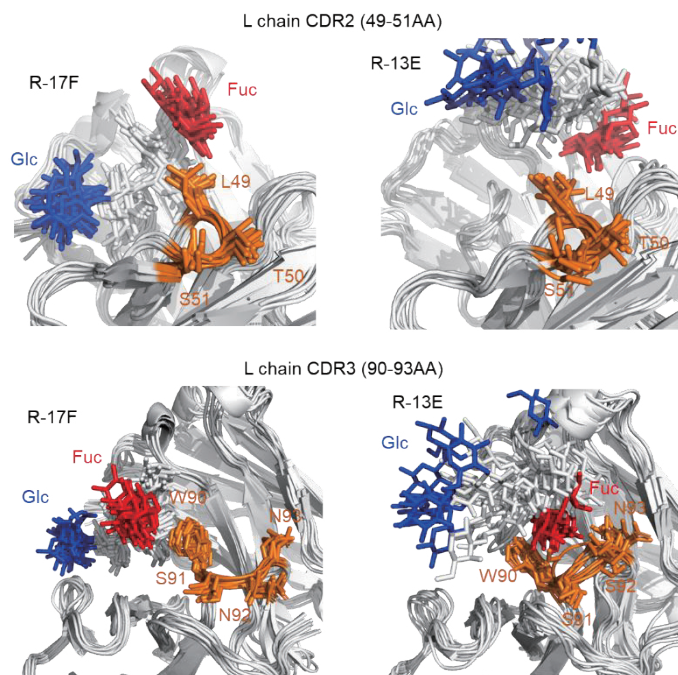


図5. R-17F-LNFP I 複合体と R-13E-LNFP I 複合体が異なる領域の構造の拡大図。(A) L鎖 (L) CDR2 周辺構造 (アミノ酸: 49-51)。(B) L鎖 CDR3 周辺構造 (アミノ酸残基 90-93)。100 ナノ秒シミュレーションにおける抗体-LNFP I 複合体の重ね合わせスナップショットを示す。主要アミノ酸残基を含む構造を10 ナノ秒ごとに抽出。Fuc 残基は赤、Glc 残基は青で表示。

R-17F-LNFP I 複合体では、CDR2 領域のアミノ酸残基はシミュレーション全体を通じて一貫して類似した立体構造が観察できた。これに対し、R-13E-LNFP I 複合体における同領域のアミノ酸残基はフレキシブルな配向を示した。同様に、R-17F の CDR3 領域内のアミノ酸残基は安定した立体構造をとったが、R-13E ではフレキシブルな配向を示した。特に、R-13E における 91-92 残基のアミノ酸は著しく不規則な運動を示した。これらの結果から、LNFP I が R-13E の結合ポケットに適切に適合できず、その結果として CDR 領域のアミノ酸がフレキシブルな配向を示すことが示唆された。

最後に、MD 計算により得られた R-17F-LNFP I 複合体の構造的妥当性を検証するため、LNFP I の還元末端から脂質部分を延長し、糖鎖-脂質構造体を脂質二重層モデルに埋め込むことで、脂質二重層環境下における複合体モデルを構築した。その結果、抗体分子と脂質二重層との間に立体的衝突は認められなかった (図6)。

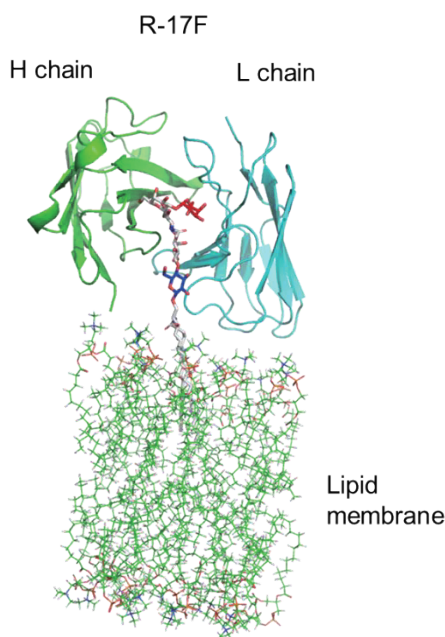


図 6. 脂質二重膜上の LNFP I 含有糖脂質と複合体を形成した R-17F 抗体のモデル。モデルは、100 ナノ秒時点の分子動力学シミュレーションから得られた R-17F-LNFP I 複合体を用いて構築された。抗体-糖脂質複合体が脂質膜に埋め込まれたモデルを生成するため、LNFP I の還元末端から脂質鎖を延長した。

これは、本研究において構築した R-17F-LNFP I 複合体モデルが、脂質膜環境下においても物理的に妥当な配置をとり得ることを示しており、本モデルの構造的妥当性を支持するものである。

4. 結論

本研究は、困難とされる糖鎖-抗体複合体の構築と、その結合様式の解明を計算化学と NMR を組み合わせることで明らかにした。この手法は他の組み合わせの抗体-糖鎖複合体に対しても応用でき、新規抗体デザインに貢献することができる。

謝辞

本研究は、東北大学サイバーサイエンスセンターのスーパーコンピュータを活用することで実現することができた。また、研究遂行にあたり、同センターの関係各位より貴重なご指導とご協力を賜ったことに深く感謝申し上げます。さらに、MD シミュレーションにご協力いただいた李秀栄博士（国立研究開発法人 医薬基盤・健康・栄養研究所）、ならびに NMR 測定に多大なご支援をいただいた 松木智之様、佐藤慎一様（東北医科薬科大学） に対し、心より感謝の意を表す。

参考文献

[1] H. Nakao, et al., Characterization of novel antibodies that recognize sialylated keratan

- sulfate and lacto-*N*-fucopentaose I on human induced pluripotent cells: comparison with existing antibodies, *Glycobiology*, **33**, 150-164 (2023)
- [2] J. Jumper, R. Evans, A. Pritzel, et al. Highly accurate protein structure prediction with AlphaFold, *Nature*, **596**, 583–589 (2021)
- [3] F.A. Momany and R.Rone, Validation of the general purpose QUANTA ®3.2/CHARMm® force field, *J. Comput. Chem.*, **13**, 888–900 (1992)
- [4] C. Kobayashi, J. Jung, Y. Matsunaga, T. Mori, T. Ando, K. Tamura, M. Kamiya, and Y. Sugita, GENESIS 1.1: A hybrid-parallel molecular dynamics simulator with enhanced sampling algorithms on multiple computational platforms, *J. Comput. Chem.*, **38**, 2193-2206 (2017)
- [5] J. Jung, T. Mori, C. Kobayashi, Y. Matsunaga, T. Yoda, M. Feig, and Y. Sugita, GENESIS: a hybrid-parallel and multi-scale molecular dynamics simulator with enhanced sampling algorithms for biomolecular and cellular simulations, *Wiley Interdiscip Rev. Comput. Mol. Sci.*, **5**, 310-323 (2015)
- [6] C. Dominguez, R. Boelens, and Alexandre M. J. J. Bonvin, HADDOCK: A Protein–Protein Docking Approach Based on Biochemical or Biophysical Information, *J. Am. Chem. Soc.*, **125**, 1731-1737 (2003)
- [7] S.J. de Vries, A.D.J. van Dijk, M. Krzeminski, M. van Dijk, A. Thureau, V. Hsu, T. Wassenaar and A.M.J.J. Bonvin, HADDOCK versus HADDOCK: New features and performance of HADDOCK2.0 on the CAPRI targets, *Proteins.*, **69**, 726-733 (2007)

[共同研究成果]

液滴の帯電挙動を考慮した VOF シミュレーション

松川 嘉也：東北大学大学院工学研究科
 酒田 一帆：東北大学大学院工学研究科
 宮川 泰明：弘前大学大学院理工学研究科
 齋藤 泰洋：九州工業大学大学院工学研究院
 岡部 孝裕：弘前大学大学院理工学研究科
 城田 農：弘前大学大学院理工学研究科
 大黒 正敏：八戸工業大学工学部機械工学科
 福野 純一：本田技研工業株式会社完成車生産技術部
 青木 秀之：東北大学大学院工学研究科

1. 背景

自動車製造における塗装工程は自動車の商品価値を高める重要な役割を担っており、回転霧化塗装機が広く利用されている。本塗装機では、塗料をベルカップと呼ばれる回転体の遠心力によって微粒化した後に、印加した電場により塗料液滴の塗着効率を高めている。我々の研究グループでは回転霧化塗装機の塗装品質および塗着効率の向上を目的として、電場下における液体の微粒化挙動や車体表面への塗着挙動を支配する因子について調査を行っている。最近では、特に塗料の帯電量や電荷の分布が微粒化および塗着に大きな影響を与えていると考えている。これまで、液体が外部電場の影響により変形するマクロ的な現象に関して幅広く実験的検討が行われてきた[1-5]。しかし、液体の帯電挙動や内部の電荷の分布に代表されるミクロ的な現象に着目した検討は僅少である。その理由の一つに、電荷という肉眼では確認が不可能な対象に対して実験的検討が困難であることが挙げられる。そのため、数値シミュレーションを用いた検討が有効である。加えて、液体のダイナミックな変形を伴う現象の場合 volume of fluid (VOF)法を用いた混相流解析の枠組みで電磁気計算を行った例が複数存在する[6-13]。これまでに多種多様な電磁気計算モデルが考案されてきたが、Leaky dielectric model (LDM)に基づくもの[14]が多い。液体の電氣的緩和時間が液体の流動の特性時間よりも十分小さい場合に、電荷が二相間に瞬時に蓄積すると仮定するモデルであるが、LDMによる近似的取り扱いが研究者により異なる。一方で、回転霧化塗装機から射出される塗料のようにダイナミックな変形を伴う系において、LDMの適用の可否には疑問が残る。そこで、本研究では液体の変形を高精度に解析しながら液体の電磁氣的現象を十分に表現可能なモデルの構築を目指し、支配方程式の選定および液滴の帯電挙動に関する検証問題に取り組んだ。

2. 支配方程式

2.1. 電氣流体力学における電磁気の基礎式

Tomar *et al.*は解析対象の二相の誘電性および導電性の組み合わせに対して制限を課した上で、VOF法を用いた電氣流体力学シミュレーションを行った。流体運動の支配方程式は、Navier-Stokes方程式およびVOF関数の移流方程式である。解析対象の二相の組み合わせを絶縁体-絶縁体とした場合、電氣的緩和時間 $t_e(= \epsilon_0 \epsilon_r / \sigma)$ が流体運動の特性時間 $t_f(= \rho d^2 / \mu)$ よりも十分に長い場合液体内部の電荷の移動は考慮しない。ここで、 ϵ_0 、 ϵ_r 、 σ 、 ρ 、 d および μ は真空の誘電率、連続相の比誘電率、電氣導電率、流体の密度、代表長さおよび流体の粘度を表す。絶縁性液体に外部から電場を印加すると、外部電場の影響を打ち消すために液体内部の分子が分極し液体内部に新たな電場が発生する。電磁気現象の支配方程式は、ガウスの法則から導かれる電位ポテンシャル ϕ のラプラス方程式である。

$$\nabla \cdot (\epsilon_0 \epsilon_r \nabla \phi) = 0 \quad (1)$$

完全な絶縁性流体の場合、電荷の存在を無視するため電荷に関する基礎式は存在しない。また、電磁気現象は電気と磁気が相互に影響を及ぼす。一方で、電気流体力学では磁氣的緩和時間 $t_m (= \mu_M \sigma d^2)$ は電氣的緩和時間 $t_e (= \epsilon_0 \epsilon_r / \sigma)$ よりも十分に短いことから磁場の影響は無視できる。ここで、 μ_M は透磁率を表す。そのため、Maxwell 方程式における磁束に関する項を含む基礎式を解かない。式(1)から得られた電位ポテンシャルをもとに式(2)を用いて電気力 \mathbf{F}_e を算出し、運動量保存式の外力項に組み込むことで流体運動と電磁気現象をカップリングする。

$$\mathbf{F}_e = \nabla \cdot \mathbf{T}_e = -\frac{1}{2} \mathbf{E} \cdot \nabla \mathbf{E} (\epsilon_0 \epsilon_r) \quad (2)$$

\mathbf{E} および E は電場ベクトルおよび電場の大きさを表す。電場ベクトルは電位ポテンシャルの勾配で表される(式(3))。絶縁体は電荷を保持しない。そのため、電気力には分子内の双極子に作用する分極力のみが含まれ電荷に作用するクーロン力は無視される。

$$\mathbf{E} = -\nabla \phi \quad (3)$$

一方で、解析対象を導電体-導電体とした場合、液体内部の電荷の移動を考慮しなければならない。Tomar *et al.* [15]は Leaky dielectric model (LDM)を用いることで電荷の移流を部分的に考慮した。LDMはTaylorによって提唱された[16]。液体が導電性を持つ場合、電氣的緩和時間 t_e が流体運動の特性時間 t_m よりも数オーダー短いことから電荷が二相界面に瞬時に蓄積すると仮定したモデルである。電磁気現象の支配方程式は式(4)のガウスの法則から導かれる電位ポテンシャルのポアソン方程式と式(5)で表される電流密度ベクトルの保存を表す式である。

$$\nabla \cdot (\epsilon_0 \epsilon_r \nabla \phi) = -q \quad (4)$$

$$\nabla \cdot \mathbf{J} = \nabla(\sigma \mathbf{E}) = 0 \quad (5)$$

ここで、 q および \mathbf{J} は電荷密度および電流密度ベクトルを表す。式(5)は式(6)で表される時間依存を考慮する電荷密度の輸送方程式から導かれ、LDMを適用して非定常項を省略した式である。

$$\frac{Dq}{Dt} + \nabla \cdot \mathbf{J} = 0 \quad (6)$$

式(4)および(5)から得られた電位ポテンシャルおよび電荷密度を用いて、式(7)から運動量保存式の外力項に組み込む電気力を求める。

$$\mathbf{F}_e = \nabla \cdot \mathbf{T}_e = -\frac{1}{2} \mathbf{E} \cdot \nabla \mathbf{E} (\epsilon_0 \epsilon_r) + q \mathbf{E} \quad (7)$$

絶縁体-絶縁体の場合と異なり液体内部に電荷が存在することから、電気力として分極力に加えてクーロン力が追加されている。

Lopez-Herrera *et al.* [17]はTomar *et al.*の手法を拡張した。一切の仮定を用いずに電磁気現象を計算することで、解析対象の電氣的物性値に依存せずに解析が可能であることを報告した。流体運動の支配方程式はTomar *et al.* [15]と同様である。電磁気現象の支配方程式は、式(4)の電位ポテンシャルのポアソン方程式と式(8)で表される電荷密度の輸送方程式である。

$$\frac{\partial q}{\partial t} + \nabla \cdot (q \mathbf{u} - \sigma \nabla \phi) = 0 \quad (8)$$

2.2. 流体力学に関する支配方程式

既往のVOF法の枠組みで電気流体力学シミュレーションを行った例を概観した。本章の最終的な目的は、電氣的緩和時間の差異を計算しながら液滴内部の電荷分布を計算可能な電気流体力学シミュレーション手法の確立である。既往の手法では、電位ポテンシャルのラプラス方程式やポアソン方程式を第一に解くことで系内の電位ポテンシャルの分布を計算する。その後、電荷密度に関する輸送方程式を先に求めた電位ポテンシャルの分布を用いて解くことにより、新たな電荷

密度の分布を計算している．一方で，実現象においては電位ポテンシャルと電荷は相互に影響を及ぼす．よって，電位ポテンシャルのポアソン方程式および電荷密度の輸送方程式は同時に解く必要があると考える．また，これら 2 式を同時に解くことにより，電位ポテンシャルのポアソン方程式に含まれる誘電率 ε および電荷密度の移流方程式に含まれる電気導電率 σ の寄与が同時に計算される．すなわち，電氣的緩和時間の影響が計算可能となる．そこで，本手法では電磁気現象の基礎式である電位ポテンシャルのポアソン方程式および電荷密度の輸送方程式を有限体積法に基づき離散化し，同時に行列ソルバーに与えることで，電位ポテンシャルと電荷密度の解を同時に求める手法を提案する．これ以降，本手法を **Coupled solver** と呼称する．

流体運動の支配方程式は，式(9)および(10)から成る Navier-Stokes 方程式および式(11)で表される VOF 関数の移流方程式である．

$$\nabla \cdot \mathbf{u} = 0 \quad (9)$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) = -\frac{1}{\rho} \nabla p + \nabla \cdot \left(\frac{\mu}{\rho} \nabla \mathbf{u} \right) + \mathbf{F} \quad (10)$$

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u}f) = 0 \quad (11)$$

ここで， \mathbf{u} ， t ， ρ ， p ， μ ， \mathbf{F} ，および f はそれぞれ，速度ベクトル，時間，密度，圧力，粘度，外力ベクトルおよび VOF 関数である．電磁気現象の基礎式は，式(4)および式(8)である．誘電率および電気導電率の平均化手法についてシグモイド関数を用いた平均化の手法を採用した．

$$\frac{1}{\sigma} = \frac{S(f)}{\sigma_1} + \frac{1-S(f)}{\sigma_2} \quad (12)$$

$$\frac{1}{\varepsilon} = \frac{S(f)}{\varepsilon_1} + \frac{1-S(f)}{\varepsilon_2} \quad (13)$$

ここで，シグモイド関数は以下の式で表される．二相界面付近で明確に気相セルと液相セルを表せるように， $a = 20$ とした．

$$S(f) = \frac{1}{1 - e^{-af}} \quad (a > 0) \quad (14)$$

これにより，二相界面付近での計算格子で誘電率および電気導電率を明確に分断される．実現象では気相において導電は生じず液相のみで導電が生じる．液滴内部において電荷が存在するように数値解析的な工夫を施した．**Coupled solver** における式(4)および式(8)の取り扱いについて説明する．式(4)および式(8)は互いに影響を及ぼすため二式を同時に満たすように連立させて解く．有限体積法に基づいて離散化する．

$$\begin{pmatrix} \mathbf{A}_1 & \mathbf{C}_1 \\ \mathbf{A}_2 & \mathbf{C}_2 \end{pmatrix} \begin{pmatrix} \phi \\ q \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \quad (15)$$

ここで， \mathbf{A}_i ， b_i および \mathbf{C}_i ($i = 1, 2$)は離散化係数である． $i = 1$ のとき式(4)から得られた離散化係数を， $i = 2$ のとき式(8)から得られた離散化係数を示す．得られた離散化方程式を式(15)の行列形式に格納し，疎行列ソルバーで解いた．得られた電位ポテンシャルおよび電荷密度から分極力ベクトルとクーロン力ベクトルの和で表される電気力ベクトル \mathbf{F}_e を算出し，式(10)の外力項 \mathbf{F} に組み込んだ． \mathbf{E} は電位ポテンシャルの勾配から算出される電場ベクトルである．**Coupled solver** を採用することにより，二式を満たすように電位ポテンシャルおよび電荷密度が算出される．また，二式に含まれる誘電率および電気導電率も同時に計算されるため，電氣的緩和時間の影響も正確に計算することが出来る．

3. 解析条件

3.1. 電気流体力学における電磁気の基礎式

電場が印加された解析領域に単一液滴を配置し、液滴の変形度合いを実験値および既往の研究例と比較した。本検証問題では Huang *et al.*の解析条件を模擬した[18]。疑似的な2次元の解析対象を図1に示す。 $x=0$ において対称境界であると仮定し、1/2領域を解析領域とした。流体運動に関する境界条件として解析領域の南面を対称境界条件とし、その他の面をすべりあり境界とした。また、電磁気に関する変数に対して境界条件を与える必要がある。解析領域東面をアースし西面に正の電位ポテンシャルを与えることで x 方向に一様な電場を発生させた。液滴の存在下では、液滴の誘電率の影響で電場は一様ではなくなり、液滴の誘電率の大きさに比例して液滴内部の電位勾配が増大する。北面の境界条件として電位ポテンシャルの勾配をゼロとした。ここで、電位ポテンシャルの境界条件を与えることと電荷密度の境界条件を与えることが同義であることを述べておく。 $10\text{ mm} \times 3\text{ mm} \times 0.3\text{ }\mu\text{m}$ の3次元解析領域を等間隔格子により x, y および z 方向に $1000 \times 300 \times 3$ 分割した。初期条件として、直径 $R = 1.5\text{ mm}$ の液滴をその中心が $(x, y, z) = (5\text{ mm}, 0\text{ mm}, 0\text{ mm})$ に配置した。液滴にはシリコンオイルの物性値を与え、周囲を水で満たした。詳細な液体の物性値に関して表1に記す。

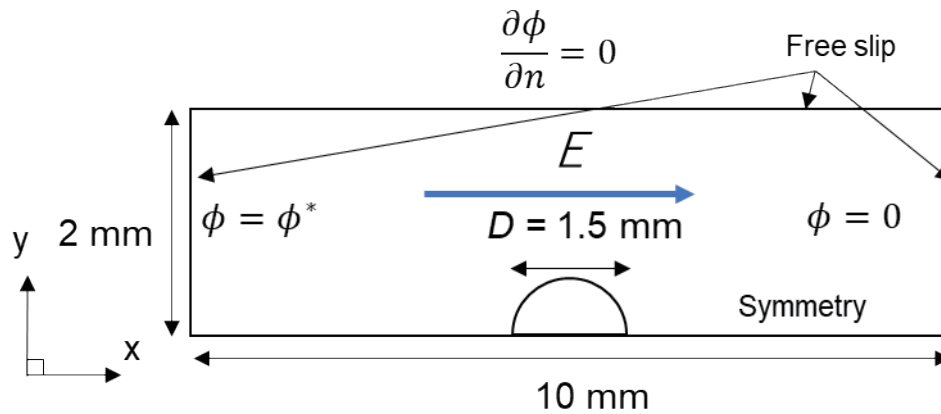


図1 解析対象

表1 シリコンオイルと水の物性値

	密度 [kg/m ³]	粘度 [mPa s]	比誘電率 [-]	電気伝導度 [S/m]	界面張力係数 [mN/m]
シリコンオイル (液滴)	960	870	2.26	8.0×10^{-14}	26.7
水 (連続相)	1000	1.005	82	2.0×10^{-2}	

支配方程式を有限体積法に基づき離散化した。時間進展に Crank-Nicolson 法を用いた。運動量保存式の移流項には TVD スキームを適用し流束制限関数には Min-Mod 関数を用いた。拡散項の離散化には二次精度中心差分法を用いた。VOF 関数の移流項の離散化に flux-blending interface-capturing schemes (FBICS) [19]を適用した。FBICS は bounded downwind (BD)スキームと high resolution (HR)スキームを適宜利用する離散化スキームであり、界面の再構築を必要としない。電荷密度の輸送方程式の移流項には運動量保存式の場合と同様に TVD スキームを適用し、流束制限関数には Min-Mod 関数を用いた。一方の、電気導電項の離散化に二次精度中心差分法を用いた。圧力速度結合解法として SMAC[20]を用い、圧力補正值のポアソン方程式は AMGS[21]を用いて解いた。その他の基礎式は AGMG[22–24]を用いて解いた。界面張力 \mathbf{F}_s は二相の密度による重みづけを施した CSF [25,26]モデルから算出した。

4. 結果と考察

図2に各 Ca_E における液滴の変形度合いの時間変化に関するグラフを示す。 $Ca_E = 0.00$ [-]すなわち $\phi^* = 0.00$ kVの場合、液滴には電気力が作用しないため液滴は界面に作用する界面張力および圧力のつり合いにより初期の形状を維持した。 $Ca_E = 0.05$ [-]すなわち $\phi^* = 2.98$ kVの場合、時間の進展に伴い変形度合いは緩やかに増加したが、解析時間内に定常には達しなかった。変形度合いが正であることから液滴は解析領域の水平方向に扁平したことがわかる。解析終了時における変形度合いの値は**1.106**となった。 $Ca_E = 0.10$ [-]すなわち $\phi^* = 4.22$ kVの場合、解析初期において変形度合いは大きく増加したが、時間の進展にともない約0.05 s付近で増加の度合いが小さくなった。最終的に液滴の形状は定常に達していると定性的に判断し、最終的な変形度合いの値は**1.233**となった。 $Ca_E = 0.05$ の条件における結果と比較すると、最終的な液滴形状の変形度合いが増大している。これは電場強度の増大に伴い液滴に作用する電気力の強度が増大したためである。 $Ca_E = 0.15$ [-]すなわち $\phi^* = 5.17$ kVの場合、解析初期において変形度合いは $Ca_E = 0.10$ [-]の場合と比較してより急激に増加した。時間の進展に伴い約0.05 sを経過すると液滴の形状は定常に達したと定性的に判断した。最終的な変形度合いの値は**1.292**となった。 $Ca_E = 0.10$ [-]の条件における結果と同様に電場強度の増大に伴い液滴に作用する電気力の強度が増大した結果、液滴は大きく変形した。 $Ca_E = 0.20$ [-]、 $\phi^* = 5.97$ kVの場合、解析初期において変形度合いは $Ca_E = 0.15$ [-]の場合と比較してより急激に増加した。また、約0.02 s経過後、変形度合いの変化はいったん緩やかになったが0.04 s経過後に増加に転じ、最終的に定常へと達した。これまでの計算条件で見られなかった、変形度合いに二度の特異的な変化が観察された。一方で、最終的な変形度合いの値は**1.420**となった。本検証問題において最も強力な電場を印加したため、液滴は最も大きく変形した。

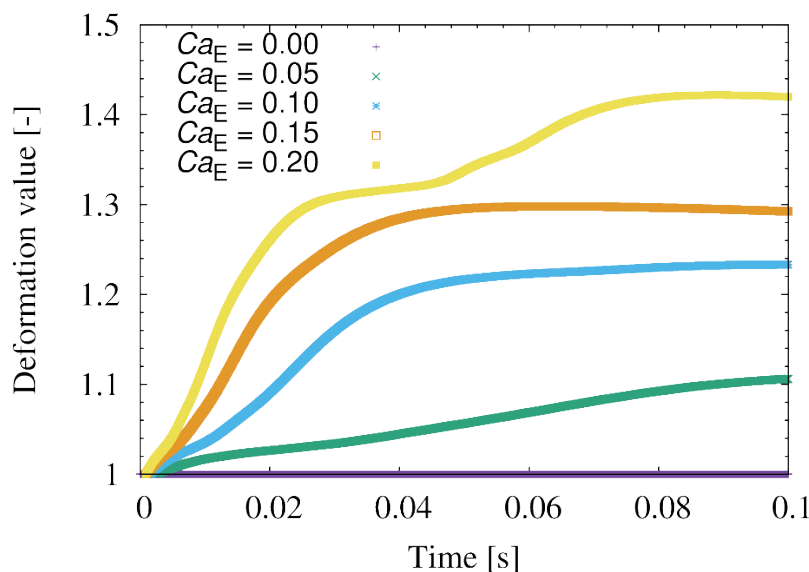


図2 変形の時間変化. $Ca_E = 0.00, 0.05, 0.10, 0.15$ and 0.20 .

$Ca_E = 0.05$ [-]の条件を代表させて、解析領域内部の電磁気関連のプロファイルに着目することで液滴が解析領域の水平方向に扁平する機構に関して考察する。図3に解析領域内部の(a)電場強度、(b)分極力の大きさ、(c)電荷密度および(d)クーロン力の大きさの分布を示す。(a)電場強度の分布に着目すると、液滴の水平方向の界面付近に強力な電場が印加されている一方で、液滴の鉛直方向の界面付近では電場の強度が減衰している。これは、二相間の誘電率の差による影響である。液滴の水平方向の界面において、誘電率が大きく変化する一方で液滴の鉛直方向では誘電率の変化が小さいためである。(b)分極力の大きさの分布に着目すると液滴の水平方向の界面に強力な分極力が作用している。一方で、液滴の鉛直方向には微弱な分極力が作用している。これは、分極

力が液滴の界面方向に作用し、その大きさが電場強度に依存するためである。以上より、誘電率および電場の影響で液滴は水平方向に牽引されることが分かった。続いて、(c)電荷密度の分布に着目すると、液滴表面に電荷が対を成して蓄積している。これは、誘電率が異なる液滴が分極したことを表している。外部の電場の影響を打ち消すために液滴内部において、外部電場とは反対方向の内部電場が生じる。電荷の偏りが生じた場合にクーロン力が作用する。最後に(d)クーロン力の大きさの分布に着目すると、電荷密度が分布している界面において電場の大きさに比例したクーロン力が液滴に作用していることが分かる。クーロン力ベクトルの作用方向は電場ベクトルの方向に一致するため、液滴を水平方向に牽引するようにクーロン力が作用している。以上より、分極力とクーロン力の水平方向成分が液滴に作用することで、液滴は水平方向に扁平し、変形度合いの値が増大した。

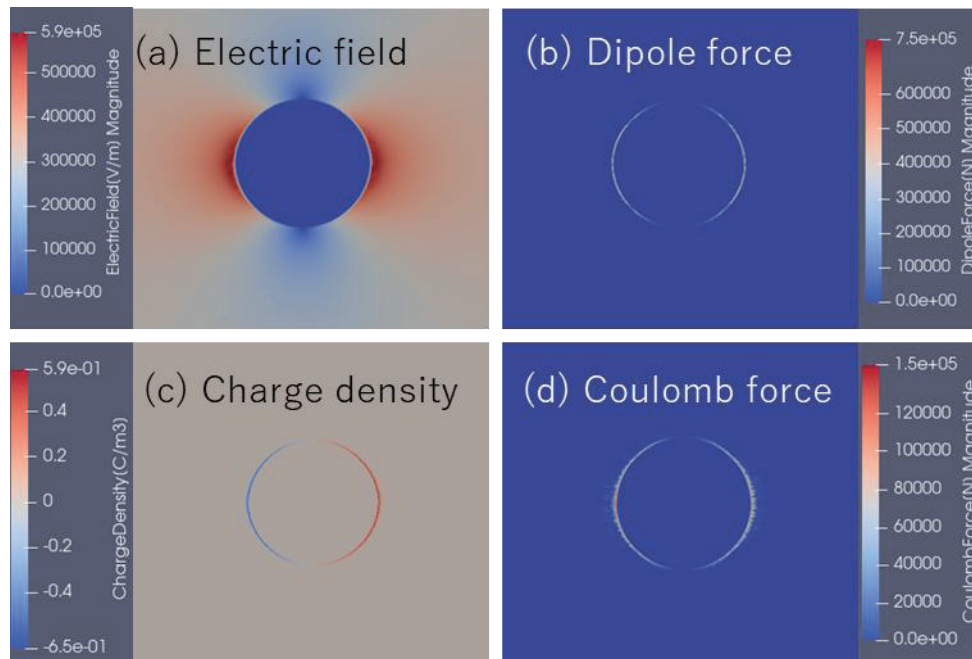


図3 $Ca_E = 0.05$ [-] and $t = 0.01$ s における (a) 電場強度、(b) 双極子力、(c) 電荷密度、(d) クーロン力の分布

謝辞

本研究は、東北大学サイバーサイエンスセンターのスーパーコンピュータを利用することで実現することができた。また、研究にあたっては同センター関係各位に有益なご指導とご協力をいただいた。

参考文献

- [1] T. S, C. RG, M. SG, Electrohydrodynamic deformation and burst of liquid drops, *Phil Trans Roy Soc Lond. Ser Math Phys Sci.* 269 (1971) 295–310.
- [2] R.L. Grimm, J.L. Beauchamp, Dynamics of Field-Induced Droplet Ionization: Time-Resolved Studies of Distortion, Jetting, and Progeny Formation from Charged and Neutral Methanol Droplets Exposed to Strong Electric Fields, *J. Phys. Chem. B.* 109 (2005) 8244–8250.
- [3] Y.-M. Jung, H.-C. Oh, I.S. Kang, Electrical charging of a conducting water droplet in a dielectric fluid on the electrode surface, *J. Colloid Interface Sci.* 322 (2008) 617–623.
- [4] P. Di Marco, R. Kurimoto, G. Saccone, K. Hayashi, A. Tomiyama, Bubble shape under the action of electric forces, *Exp. Therm. Fluid Sci.* 49 (2013) 160–168.

- [5] Y. Tian, Y. Liu, Z. Peng, C. Xu, D. Ye, Y. Guan, X. Zhou, W. Deng, Y. Huang, Air entrapment of a neutral drop impacting onto a flat solid surface in electric fields, *J. Fluid Mech.* 946 (2022) A21.
- [6] J.M. López-Herrera, S. Popinet, M.A. Herrada, A charge-conservative approach for simulating electrohydrodynamic two-phase flows using volume-of-fluid, *J. Comput. Phys.* 230 (2011) 1939–1955.
- [7] S. Sunder, G. Tomar, Numerical investigation of a conducting drop’s interaction with a conducting liquid pool under an external electric field, *Eur. J. Mech. - BFluids.* 81 (2020) 114–123.
- [8] A. Zhang, Y. Wang, D. Sun, S. Yu, B. Yu, Y. Li, Development of a VOF+LS+SPP method based on FLUENT for simulating bubble behaviors in the electric field, *Numer. Heat Transf. Part B Fundam.* 71 (2017) 186–201.
- [9] J.M. López-Herrera, A.M. Gañán-Calvo, S. Popinet, M.A. Herrada, Electrokinetic effects in the breakup of electrified jets: A Volume-Of-Fluid numerical study, *Int. J. Multiph. Flow.* 71 (2015) 14–22.
- [10] B. Nath, G. Biswas, A. Dalal, Influence of electric field on deformation of a drop in shear flow, *Phys. Fluids.* 31 (2019) 042102.
- [11] M. Rahmanpour, R. Ebrahimi, A. Pourrajabian, Numerical simulation of two-phase electrohydrodynamic of stable Taylor cone–jet using a volume-of-fluid approach, *J. Braz. Soc. Mech. Sci. Eng.* 39 (2017) 4443–4453.
- [12] Z. Wang, Q. Dong, Y. Zhang, J. Wang, J. Wen, Numerical Study on Deformation and Interior Flow of a Droplet Suspended in Viscous Liquid under Steady Electric Fields, *Adv. Mech. Eng.* 6 (2014) 532797.
- [13] G. Tomar, D. Gerlach, G. Biswas, N. Alleborn, A. Sharma, F. Durst, S.W.J. Welch, A. Delgado, Two-phase electrohydrodynamic simulations using a volume-of-fluid approach, *J. Comput. Phys.* 227 (2007) 1267–1285.
- [14] D.A. Saville, ELECTROHYDRODYNAMICS: The Taylor-Melcher Leaky Dielectric Model, *Annu. Rev. Fluid Mech.* 29 (1997) 27–64.
- [15] G. Tomar, D. Gerlach, G. Biswas, N. Alleborn, A. Sharma, F. Durst, S.W.J. Welch, A. Delgado, Two-phase electrohydrodynamic simulations using a volume-of-fluid approach, *J. Comput. Phys.* 227 (2007) 1267–1285.
- [16] D.A. Saville, ELECTROHYDRODYNAMICS: The Taylor-Melcher Leaky Dielectric Model, *Annu. Rev. Fluid Mech.* 29 (1997) 27–64.
- [17] J.M. López-Herrera, S. Popinet, M.A. Herrada, A charge-conservative approach for simulating electrohydrodynamic two-phase flows using volume-of-fluid, *J. Comput. Phys.* 230 (2011) 1939–1955.
- [18] X. Huang, L. He, X. Luo, H. Yin, D. Yang, Deformation and coalescence of water droplets in viscous fluid under a direct current electric field, *Int. J. Multiph. Flow.* 118 (2019) 1–9.
- [19] Y.Y. Tsui, S.W. Lin, T.T. Cheng, T.C. Wu, Flux-blending schemes for interface capture in two-fluid flows, *Int. J. Heat Mass Transf.* 52 (2009) 5547–5556.
- [20] A.A. Amsden, F.H. Harlow, A simplified MAC technique for incompressible fluid flow calculations, *J. Comput. Phys.* 6 (1970) 322–325.
- [21] <https://hpcl.info.kogakuin.ac.jp/docs/amgs/>, (accessed on 2026/3/1).
- [22] Y. Notay, An aggregation-based algebraic multigrid method, *Electron. Trans. Numer. Anal.* 37 (2010) 123–146.
- [23] A. Napov, Y. Notay, An algebraic multigrid method with guaranteed convergence rate, *SIAM J. Sci. Comput.* 34 (2012) A1079–A1109.
- [24] Y. Notay, Aggregation-based algebraic multigrid for convection-diffusion equations, *SIAM J. Sci. Comput.* 34 (2012) A2288–A2316.
- [25] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (1992) 335–354.
- [26] D.B. Kothe, R.C. Mjolsness, M.D. Torrey, RIPPLE: A Computer Program for Incompressible Flows with Free Surfaces, (1991).

[共同研究成果]

直交格子積み上げ法を用いた五芒星グレイン形状を有する

固体燃料ロケット内部流れ場の数値流体解析

小川 秦一郎, 奥 汐梨, 佐々木 大輔

大阪公立大学 大学院工学研究科 航空宇宙海洋系専攻 航空宇宙工学分野

固体燃料ロケットは構造の簡素さや運用性, コスト面での利点から, 小型衛星打ち上げ用ロケットとしての利用が期待されている. しかし, 固体燃料ロケットの燃焼室内部では, 不安定な流れ場に起因する圧力振動が発生し, ロケットの推進性能や搭載機器に悪影響を及ぼす可能性がある. これらの現象を解明するためには, 燃焼室内流れ場の詳細な解析が重要であるが, 実機を用いた燃焼試験では多大なコストと労力を要するため, 数値流体解析の活用が有効である. 本研究では, 3次元 Building-Cube Method (BCM) ソルバーの妥当性を検証することを目的として, 星型グレイン形状を有する五芒星固体燃料ロケット内部流れ場を対象とした数値解析を実施した. 推進剤の燃焼進行に伴う燃焼室形状の変化を模擬した複数の解析モデルを用いて, 内部流れ場の特徴およびロールトルクの変化を評価した. 解析結果より, 五芒星グレイン壁面から中心方向へ流入する流れ場構造が形成されることが確認でき, 推進剤壁面噴出モデルの有効性が明らかとなった. また, ノズル出口部のロールトルクの比較結果より, 燃焼開始直後にロールトルクが最大となり, 燃焼の進行に伴い減少する傾向が得られた. この傾向は先行研究の解析結果および飛行実験結果と良好に一致した. さらに, 五芒星スロット形状を表す l/d とロールトルクの関係についても, 先行研究と同様に指数関数的関係が得られた. 以上の結果より, 本研究で用いた3次元 BCM ソルバーは, 星型グレイン形状を有する固体燃料ロケット内部流れ場およびロールトルクの変化を適切に再現できることが示され, 本解析手法の妥当性が確認された.

1. はじめに

近年, 小型衛星の需要拡大に伴い, その打ち上げ能力を有する小型ロケットへの関心が高まっている. とりわけ, 構造の簡素さや運用性, コスト面での優位性から, 固体燃料ロケットは小型衛星打ち上げ用のロケットとしての利用が期待されている. しかし, 固体燃料ロケットの燃焼室内部では, 不安定な流れ場によって圧力振動[1]が発生し, これによりロケットの推進性能だけではなく人工衛星などのペイロードにも振動により悪影響を及ぼすことが知られており, 振動による機器の不具合を防止するために振動抑制装置を搭載する必要がある[2]. しかし, これらの対策はロケット全体の重量増加につながるため, 燃焼室内における圧力振動の発生メカニズムを解明し, その抑制を図ることが重要な課題となっている.

従来, 圧力振動の発生メカニズムを解明するために, 小型スケールモデルを用いた燃焼実験が実施されてきた[3]. しかし, 小型モデルでは燃焼室内の形状変化や流動構造など, 実物大の固体燃料ロケットと同一の現象を再現することが困難である. さらに, 固体燃料ロケットの燃焼過程ではスラッグ (slag) と呼ばれるアルミナ粒子が放出されることが知られている[4, 5]. このスラッグは固体燃料ロケットのサイズによらず同程度の大きさで生成されるため, その影響を適切に評価する実物大モデルでの検証が必要となる. しかし, 実物大の固体燃料ロケットを用いた燃焼試験は, 多大な費用と労力を要するため, 近年では, 数値流体解析 (CFD) を活用することで, 開発コストの削減と効率的なメカニズム解明が期待されている.

現在, 航空宇宙分野を含めた幅広い分野で流体の数値計算が行われているが, 一般的に数値計算に用いられる格子は, 物体の形状に沿って作成される構造格子や非構造格子が主流である. 例えば, 固体燃料ロケットの燃焼室内の CFD 解析についても, 構造格子を用いた解析[4, 6]が多く実施されてきている. しかし, 従来手法の場合には, 複雑な形状の格子を作成するために多くの

時間を要することや、アルゴリズムが複雑化するため空間高次スキームの構築が困難である等の課題がある。これらの課題を解決する方法の一つに、直交格子積み上げ法 (Building-Cube Method, BCM) がある。BCM は、物体の形状にかかわらず、計算領域を Cube と呼ばれる立方体で区切ることによって格子を生成するため、格子生成の負荷を抑えることができる。さらに、本手法は空間高次精度化が容易であり、渦構造を高精度に捉えることが可能である。加えて、移動境界への対応が容易であるため、固体燃料ロケットにおける推進剤燃焼に伴う推進剤形状の変化を模擬する移動境界問題への適用が期待されている。

先行研究において 2 次元 BCM ソルバーや 3 次元 BCM ソルバーを用いた解析が報告されている [7-9]。しかし、これまでの先行研究においては円筒形の固体燃料ロケット燃焼室を対象とした解析モデルで解析が行われている [9]。しかし、固体燃料ロケットの燃焼室の形状は、星型のグレイン形状が主流であり、より現実的な固体燃料ロケット燃焼室内部の流体解析を実施するためには、星型グレイン形状での解析を実施する必要がある。

そこで本研究では、3 次元 BCM ソルバーの妥当性を検証することを目的として、星型グレイン形状を有する五芒星型固体燃料ロケット内部流れ場を対象とした数値解析を実施した。推進剤の燃焼進行に伴う燃焼室形状の変化を再現した複数の解析モデルを構築し、各モデルにおける内部流れ場の特徴およびその変化を評価した。さらに、解析結果からノズル出口におけるロールトルクを算出し、先行研究の数値解析結果および飛行実験結果と比較することで、3 次元 BCM ソルバーの妥当性を検証した。

2. 数値解析手法

本解析手法の支配方程式は、3 次元圧縮性 RANS (Reynolds-Averaged Navier-Stokes) 方程式である。乱流モデルは、Spalart-Allmaras (SA) モデルを用いている。時間積分法は LU-SGS (Lower-Upper Symmetric Gauss-Seidel) 陰解法、空間高次精度化は MUSCL (Monotonic Upstream-centered Scheme for Conservation Laws) 法、時間精度は 3 次精度 MUSCL 法、離散化手法は Cell 中心有限体積法、非粘性流束は SLAU (Simple Low Dissipation Advection Upstream Splitting Method) 法である。

BCM は、モデル形状に対して立方体の Cube に分割して格子生成する。分割した Cube は、1 つの Cube につき一片を指定した数で等間隔に分割し Cell を生成する。物理量の交換には、各 Cube に Overlap cell と呼ばれる 3 層の Cell が存在しており、その Cell を用いて物理量を交換している。例えば、大きい Cube から小さい Cube への物理量の交換は、最も近い Cell から 0 次外挿する。一方、小さい Cube から大きな Cube への物理量の交換は 1 次線形内挿する。

BCM は格子生成や高次精度化が容易など多くのメリットがある一方で、物体の壁面に傾斜があるような物体の場合には、物体と解析領域の境界面が階段状に表現されてしまう課題がある。この課題を解決するために、本研究では壁近傍の Cell で物理量をモデル化することで物体形状を変化させることなく境界条件を適用する手法である Ghost cell と Image point を使用した Immersed Boundary Method (IBM)[10]を用いて計算精度の向上を図っている。図 1 に示す通り、Inner cell のうち Fluid cell に接している Wall cell を Ghost cell と定義する。Wall cell から最小の Cell 幅の 1.75 倍に仮想的な点である Image point を作成し、その周囲の 27 点のうち Fluid cell の値から距離に応じて重みづけを行い仮想点での物理量を計算する。そして、その物理量をもとに壁面の圧力などの物理量を決定する。

Image point における物理量 q_{IP} は、それぞれの Cell 内に対応する物理量 q_i を用いて以下で表される。

$$q_{IP} = \sum_{i=1}^{27} \omega_i q_i \text{mask}(i) \quad (1)$$

mask(i) = 1 : Fluid cell

mask(i) = 0 : Inner cell

ここで、 ω_i は i 番目の Cell の中心座標との距離を h_i として、以下の式で表される。

$$\omega_i = \frac{h_i^{-2}}{\sum_{i=1}^{27} h_i^{-2}} \quad (2)$$

固体燃料ロケットモデルを解析する際には、推進剤を模擬した壁面（推進剤壁面）と推進剤のない物体を模擬した壁面（固体壁）の 2 種類の壁面条件が存在する。ここで、推進剤壁面とは、壁面からガス噴射（流入条件）が生じる条件であり、固体壁とは、壁面からガス噴射が生じない粘性壁面のことである。そこで、本研究では、IBM を用いて 2 種類の壁面モデル[11]を固体燃料ロケットモデルに設定している。以下に 2 種類の壁面モデルの Ghost cell の物理量の算出方法を示す。推進剤壁面領域と固体壁領域は、初期条件の段階で任意に指定することができる。

(a) 固体壁

仮想点での物理量 q_{IP} を用いて、Ghost cell 内の速度ベクトル \mathbf{V}_{GC} 、圧力 p_{GC} 、密度 ρ_{GC} を以下のように決定する。

$$\mathbf{V}_{GC} = \mathbf{V}_{IP} - \left(1 + \frac{d_{IP}}{d_{GC}}\right) (\mathbf{V}_{IP} \cdot \mathbf{n}) \mathbf{n} \quad (3)$$

$$p_{GC} = p_{IP} \quad (4)$$

$$\rho_{GC} = \rho_{IP} \quad (5)$$

ここで、 \mathbf{n} は物体表面に対する法線ベクトル、 d_{IP} は物体表面から仮想点までの距離、 d_{GC} は物体表面から Ghost cell の中心までの距離である。

(b) 推進剤壁面

推進剤壁面モデルの GC の物理量は以下の式を用いて算出した。

$$\mathbf{V}_{GC} = V_{IN} \mathbf{n} \quad (6)$$

$$p_{GC} = p_{IN} \quad (7)$$

$$\rho_{GC} = \rho_{IN} \quad (8)$$

流入条件における密度は、理想気体の状態方程式 ($p = \rho RT$) を用いて算出した。

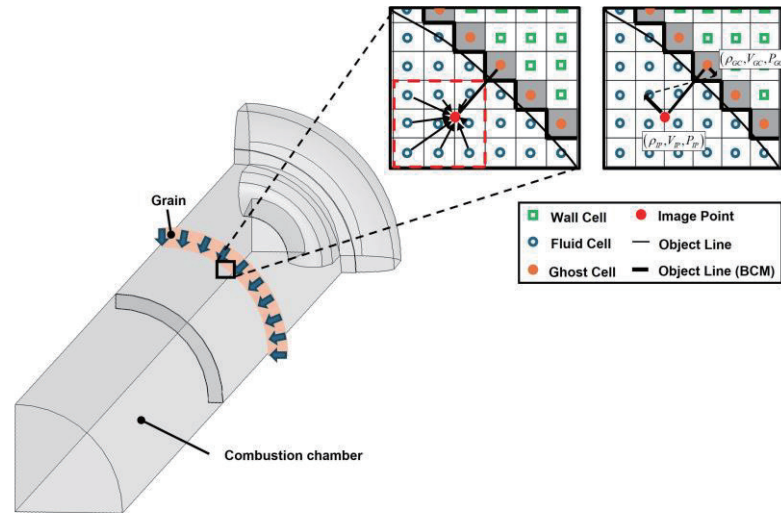


図1 固体燃料ロケットモデルと埋め込み境界法

3. 解析条件

本研究で用いた五芒星型固体燃料ロケットモデルと境界条件を図2に示す。燃焼室の長さは5041.14 mm、ノズル部分の長さは1097.6 mm、ノズル出口部の直径は1016 mmとした。燃焼室は断面が五芒星型となっている。図2(b)に示すModel 1からModel 4の4種類の形状は、固体燃料の燃焼に伴う燃焼室領域の変化を再現したモデルである。Model 1は燃焼開始0.28秒後のモデル、Model 2は燃焼開始2.02秒後のモデル、Model 3は燃焼開始4.12秒後のモデルである。また、Model 4については、形状変化を比較するための参考として、五芒星モデルの推進剤が完全に燃焼し円筒形状になった場合のモデルを想定している。境界条件は、五芒星グレイン領域(赤色の領域)は流入境界条件を設定し、ノズル壁面(緑色の領域)は粘性壁条件を設定し、ノズル出口(青色の領域)は流出境界条件を設定している。

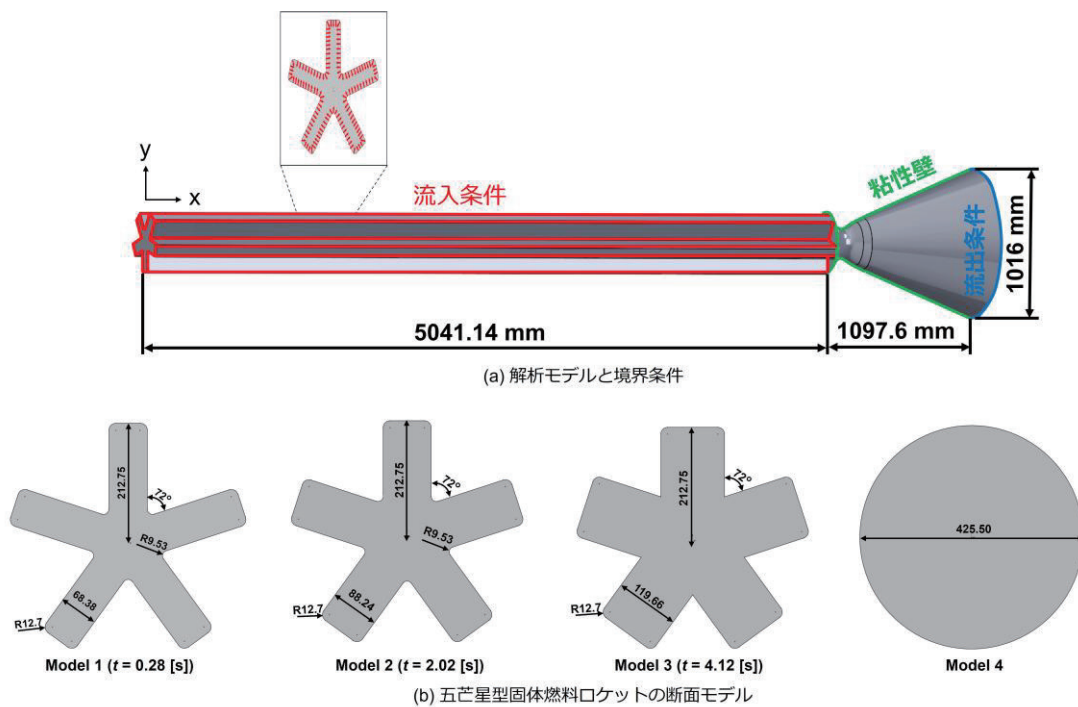


図2 五芒星型固体燃料ロケットモデルと境界条件

本研究で用いた解析格子情報を表 1 に示す。格子の総数を減らし計算負荷を低減するために、ノズル出口部の中心付近の格子は、燃焼室内部の格子よりも Cube サイズを大きくしている。Cell 数については、各 Cube を 16×16×16 で分割して生成している。

表 1 各モデルの格子情報

モデル	総 Cube 数	総 Cell 数	最小格子幅, mm
Model 1	10,950	44,851,200	48
Model 2	11,584	47,448,064	48
Model 3	63,845	261,509,120	24
Model 4	11,508	47,136,768	48

本研究の流入・流出境界条件を表 2 に示す。先行研究の解析条件[12]と同じ条件に設定している。本解析では、化学反応を伴わない非燃焼数値シミュレーションのため、流入するガスは高温の空気を模擬している。レイノルズ数は 2.51×10^8 であり、モデル長さを代表長さとして計算している。

表 2 解析条件

条件	数値
主流条件	
マッハ数	0.562
全圧	3.792×10^6 Pa
全温	2900 K
流出条件	
静圧	50 Pa

4. 解析結果と考察

推進剤壁面噴出モデルを適用した場合に、推進剤を模擬した壁面から流入条件が適切に付与されていることを確認するため、4 種類の解析モデル (Model 1~4) について壁面近傍の流入状態を比較した。図 3 にモデル上部から 2500 mm の位置における断面分布を示す。この図は、マッハ数のコンター図に速度ベクトルを重ねて示したものである。

速度ベクトルより、五芒星のグレイン全てから垂直方向に流入条件が正しく設定されていることがわかる。また、各グレインから流入したガスは、中心方向に流れており、マッハ数分布からも、中心方向にかけて加速していることがわかる。従って、壁面からの流入条件は、五芒星のグレイン形状において正しく設定されていることが確認でき、本研究で適用している推進剤壁面噴出モデルを用いても問題なく解析できていると判断できる。

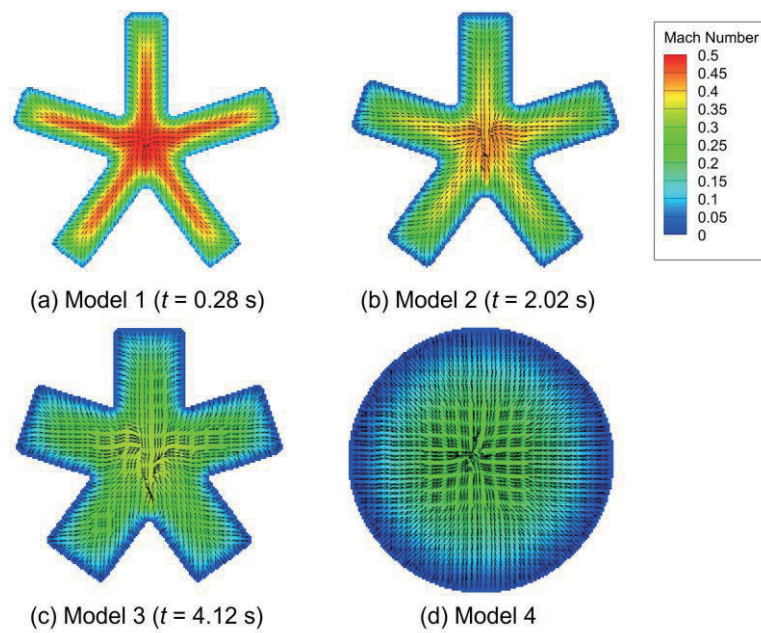


図3 モデル上部から 2500 mm 位置の断面図分布 (マッハ数と速度ベクトル)

次いで、ロールトルクについて検証した。ロールトルクは、ノズル出口部分を対象に、以下の式を用いて算出した。

$$L = \int_S \rho v_a v_r r dS \quad (9)$$

ここで、 ρ は気体の密度、 v_a は軸方向速度、 v_r は接線方向速度、 r は放射座標である。この式を用いて本解析結果からノズル出口部のロールトルクを求め、先行研究の実験結果[13]および解析結果[12]と比較した。比較結果を図4と図5に示す。図4は、燃焼開始からの時間とロールトルクの関係を示したグラフであり、図5は、五芒星型のスロット部の深さ l と幅 d の比 (l/d) とロールトルクの関係を示したグラフである。

図4より、時間経過に伴うロールトルクの変化は、先行研究の解析結果および飛行実験結果と同様の傾向を示すことが確認された。すなわち、ロールトルクは燃焼開始直後の $t = 0.28$ s (Model 1) 付近で最大となり、その後燃焼の進行に伴い減少する傾向を示した。まず、本解析結果と先行研究の解析結果を比較すると、燃焼開始直後の形状に対応する Model 1 において両者はよく一致しており、相対誤差は約 1.9%であった。一方で、燃焼の進行に伴い燃焼領域が広がっている Model 2 および Model 3 においては、先行研究の解析結果との差が大きくなることが確認された。次いで、飛行実験結果との比較を行ったところ、燃焼が進行した形状に対応する Model 3 において最も良い一致が得られた。Model 1, Model 2, Model 3 における飛行実験値との相対誤差はそれぞれ約 26%, 約 42%, 約 4%であり、燃焼が進行した状態ほど飛行実験結果に近づく傾向が確認された。ここで、飛行実験のデータは 21 回の飛行試験結果を平均したものであり、ロールトルクの値は各実験で大きくばらついていたことが報告されている。例えば、先行研究で示されているロールトルクの最大値は平均値として 84 Nm であるが、個々の飛行実験では最大 163 Nm に達した例も報告されている[13]。このことから、飛行実験におけるロールトルクのばらつきは比較的大きかったと考えられるが、具体的な不確かさは報告されていない。そのため、本解析結果と飛行実験結果との間に生じた差の一部は、実験データのばらつきに起因する可能性があると推測される。

図5に示すように、 l/d の値とロールトルクの関係についても、本解析結果は先行研究の解析結

果と飛行実験結果と同様の傾向を示した。Stella et al.[12]は、 l/d とロールトルクの間には指数関数的関係が存在することを報告しているが、本解析結果においても同様に l/d の減少に伴いロールトルクが指数関数的に減少する傾向が確認された。また、燃焼開始直後の形状に対応する Model 1 は l/d が大きい状態であり、一方で燃焼が進行した状態に対応する Model 3 は l/d が小さい状態となっている。従って、燃焼の進行に伴う形状変化がロールトルクの低下に強く影響していると考えられる。

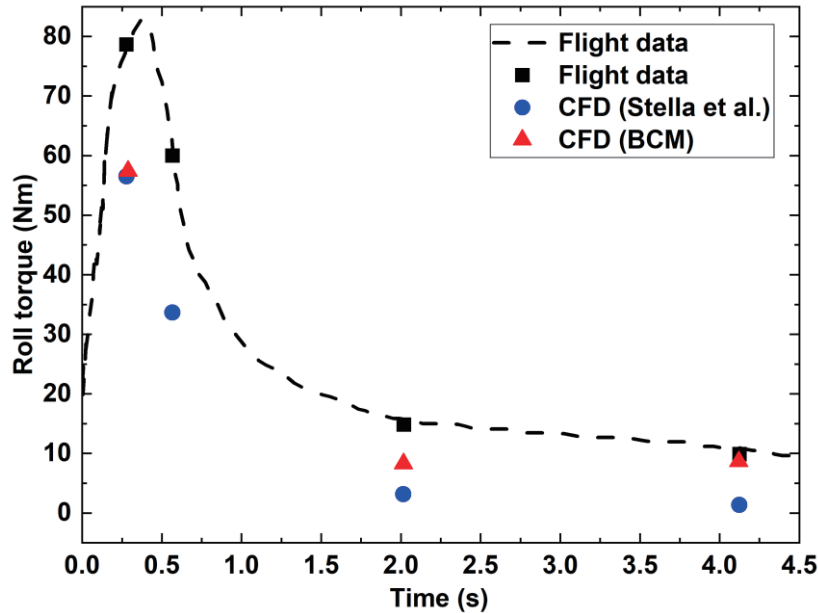


図4 燃焼開始からの時間とロールトルクの関係

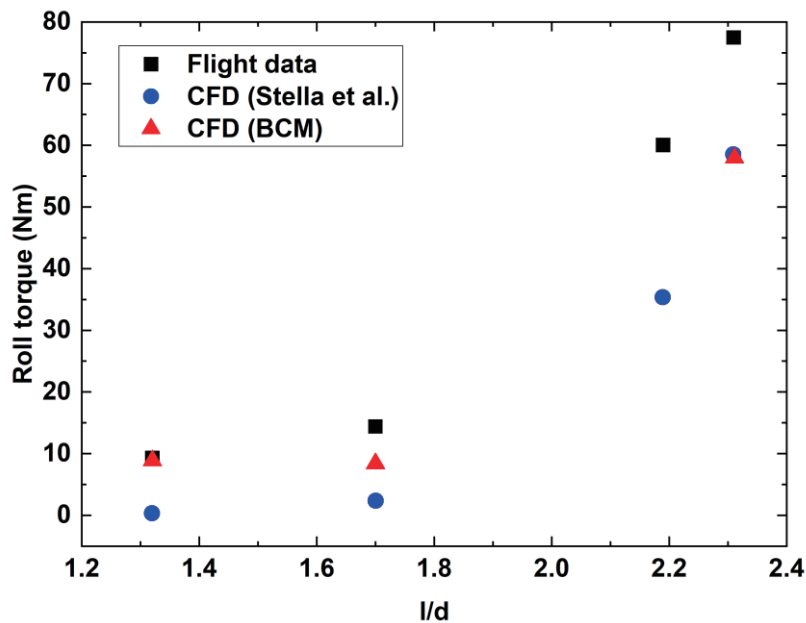


図5 五芒星型のスロット部の深さ l と幅 d の比 (l/d) とロールトルクの関係

5. まとめ

本研究では、3次元 BCM ソルバーの妥当性を検証することを目的として、星型グレイン形状を有する五芒星型固体燃料ロケット内部流れ場を対象とした数値解析を実施した。推進剤の燃焼進行に伴う燃焼室形状を模擬した複数の解析モデルを構築し、各モデルにおける内部流れ場の特徴およびロールトルクの変化について評価した。本研究結果より得られた主な知見を以下に示す。

- (1) 推進剤壁面噴出モデルを適用した解析により、五芒星グレインの各壁面から中心方向に向かう流れが形成されることが確認された。また、マッハ数分布および速度ベクトル分布から、推進剤壁面における流入条件が適切に設定されていることが確認された。
- (2) 燃焼進行に伴うロールトルクの変化は、燃焼開始直後に最大値を示し、その後燃焼室形状の変化とともに減少する傾向を示した。この傾向は、先行研究の数値解析結果および飛行実験結果と同様であった。
- (3) 本解析結果と先行研究の解析結果との比較では、燃焼開始直後の形状に対応する Model 1 において良好な一致が得られ、相対誤差は約 1.9%であった。一方で、燃焼が進行した Model 2 および Model 3 では差が大きくなる傾向が確認された。
- (4) 飛行実験結果との比較では、燃焼が進行した形状に対応する Model 3 において最も良い一致が得られ、相対誤差は約 4%であった。これにより、燃焼が進行した状態の流れ場に対しては、本解析手法が実験結果を良好に再現できることが示された。
- (5) 五芒星グレインのスロット形状を表す l/d とロールトルクの関係について、本解析結果は先行研究と同様に指数関数的な関係を示した。このことから、燃焼進行に伴うグレイン形状の変化がロールトルクの低下に強く影響することが示唆された。

以上の結果より、本研究で用いた 3次元 BCM ソルバーは、星型グレイン形状を有する固体燃料ロケット内部流れ場においてロールトルクの変化を適切に再現できることが確認され、本手法の妥当性が示された。今後は、推進剤の燃焼による形状変化を直接取り扱う移動境界条件の適用や、燃焼反応および粒子挙動を考慮した数値解析を行うことで、固体燃料ロケット燃焼室内部の流動現象のより詳細な解明を進める予定である。

謝辞

本研究は、東北大学サイバーサイエンスセンターのスーパーコンピュータを利用することで実現することができた。また、研究にあたっては同センター関係各位に有益なご指導とご協力をいただいた。

参考文献

- [1] Y. Fabignon, J. Dupays, G. Avalon, F. Vuillot, N. Lupoglazoff, G. Casalis, M. Prévost, Instabilities and pressure oscillations in solid rocket motors, *Aerospace Science and Technology*, Vol. 7, No. 3, pp. 191-200, 2003.
- [2] 森田泰弘, 大塚浩仁, 田中健作, イプシロンロケットの誘導制御, 計測と制御, 57 巻, 4 号, pp. 216-221, 2018.
- [3] J. Anthoine, M. Mettenleiter, O. Repellin, J.-M. Buchlin, S. Candel, Influence of adaptive control on vortex-driven instabilities in a scaled model of solid propellant motors, *Journal of Sound and Vibration*, Vol. 262, No. 5, pp. 1009-1046, 2003.
- [4] M. Salita, Deficiencies and requirements in modeling of slag generation in solid rocket motors, *Journal of Propulsion and Power*, Vol. 11, No. 1, pp. 10-23, 1995.
- [5] W. Tang, R. Yang, T. Zeng, J. Li, J. Hu, X. Zhou, E. Jiang, Y. Zhang, Positive effects of organic fluoride on reduction of slag accumulation in static testing of solid rocket motors of different diameters, *Acta Astronautica*, Vol. 194, pp. 277-285, 2022.

- [6] T. Shimada, M. Sekiguchi, N. Sekino, Flow Inside a Solid Rocket Motor with Relation to Nozzle Inlet Ablation, *AIAA Journal*, Vol. 45, No. 6, pp. 1324-1332, 2007.
- [7] S. Ogawa, D. Sasaki, Two-dimensional computational fluid dynamics analysis of internal flow in a simplified solid rocket motor combustion chamber, *Journal of Fluid Science and Technology*, Vol. 12, pp. JFST0023, 2017.
- [8] 小川泰一郎, 大内健太郎, 佐々木大輔, 直交格子積み上げ法によるノズル付き固体燃料ロケット内部流れの3次元流体解析, 22 卷, pp. 1-10, 2023.
- [9] S. Ogawa, D. Sasaki, Three-dimensional cold flow analysis of the effect of the nozzle cavity volume on the internal flowfield in a solid rocket motor using the building-cube method, *Fluid Dynamics Research*, Vol. 57, No. 4, pp. 045503, 2025.
- [10] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, *Journal of Computational Physics*, Vol. 227, No. 10, pp. 4825-4852, 2008.
- [11] S. Ogawa, D. Sasaki, Development of a Radial Injection Flow Model of a Cylinder Simulating a Solid Rocket Motor Geometry using the Building-Cube Method, *Transactions of the Japan Society for Aeronautical and Space Sciences*, Vol. 67, No. 6, pp. 307-317, 2024.
- [12] F. Stella, M. Giangi, F. Nardecchia, D. Barbagallo, ROLL TORQUE PREDICTION IN SRM: PRACTICAL APPLICATIONS, *Proceedings of 63rd International Astronautical Congress*, IAC-12,C4,2,23,x15674, 2012.
- [13] R. N. Knauber, Roll Torques Produced by Fixed-Nozzle Solid Rocket Motors, *Journal of Spacecraft and Rockets*, Vol. 33, No. 6, pp. 789-793, 1996.

[共同研究成果]

Deep Q-Network と LES を用いた DBD プラズマアクチュエータの 翼周り流れの剥離制御

Tan Kevin : 東京理科大学工学部情報工学科
 浅田 健吾 : 東京理科大学工学部情報工学科
 藤井 孝藏 : 東京理科大学工学部情報工学科
 立川 智章 : 東京理科大学工学部情報工学科

1. はじめに

航空機や風車などの空力機器において、翼面上の流れの剥離は揚力低下や抗力増大を引き起こし、空力性能を大きく低下させる要因となる。そのため、翼周りの流れを制御し剥離を抑制する流体制御技術は重要な研究分野として広く研究が行われている。近年、能動的に翼周り流れの剥離を抑制する手法として、誘電バリア放電 (Dielectric Barrier Discharge, DBD) を利用した DBD プラズマアクチュエータ (以下 PA) [1-3] と呼ばれるマイクロ流体制御デバイスが注目されている。PA は、図 1 に示すように 2 枚の電極とそれに挟まれた誘電体からなる単純かつ軽量の構造であり、放電により生成されるプラズマを利用した空気の流れを制御する空力デバイスである。高い応答性を持つため、Vortex Generator や定常ジェットといった従来デバイスに比べて柔軟な運用が可能なものとして期待されている。

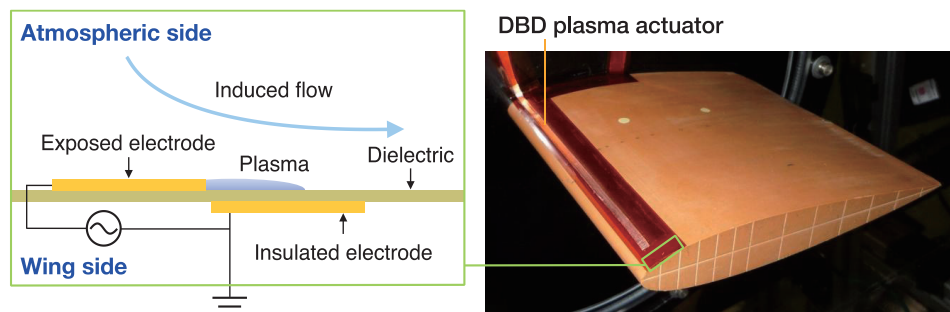


図 1: DBD プラズマアクチュエータの模式図と翼への設置例

先行研究[4-8]より、乱流遷移が生じるレイノルズ数 ($Re \cong 10^4 \sim 10^5$) の条件において、ON と OFF を間欠的に切り替えるバースト波を用いた PA のバースト駆動制御が剥離の抑制に効果的であるということが明らかになっている。図 2 はバースト駆動で用いられる一般的な入力波形であり、バースト時間 T_{on} 、バースト周期 T^+ 、ベース周波数 f_{base} によって表される。さらに、翼周りの剥離制御においては、主流速度 U_∞ と翼コード長 c で無次元化された無次元バースト周波数 $F^+ (= f^+ c / U_\infty)$ とバースト比 (Burst Ratio) $BR = T_{on} / T^+$ が用いられることが多い。特定条件における適切な F^+ に関する理解はこれまでの研究で深まりつつあるが、翼型や流れ条件が変化した場合にも常に有効であるとは限らない。そのため、各条件において適切な F^+ を設定する必要がある。実際には、時々刻々と変化する流れ場において適切な値を決定することが求められる。

流れの状況に応じて適切な駆動パラメータを決定する研究として、小川ら[9]は翼面圧力係数の分散値と閾値に基づく簡易なフィードバック剥離制御アルゴリズムを提案し、流体シミュレーションを用いてその有効性を検証している。近年は特に機械学習を用いた流体制御が大きな注目を浴びている[10]。オープンループ制御では遺伝的アルゴリズムや (Genetic Algorithm, GA) [11] やアントコロニーアルゴリズム (Ant Colony Algorithm, ACA) [12]を用いた最適な制御パラメー

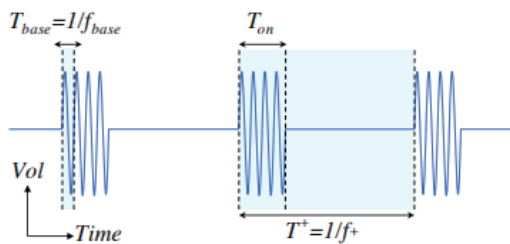


図2 バースト波形

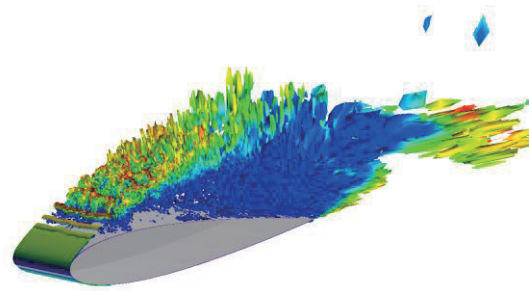


図3 NACA0015 翼周り流れ場

タの組み合わせ探索，クローズドループ制御では，遺伝的プログラミング（Genetic Programming, GP）[13]や深層強化学習（Deep Reinforcement Learning, DRL）[14]などのモデルフリーな機械学習手法の研究が盛んである．下村ら[15]はPAとDeep Q-Network（DQN）を用いた翼周り流れの剥離制御を提案し，風洞実験において固定 F^+ よりも高い制御効果が得られることを示している．数値シミュレーションを用いた研究として，高田ら[16]はRANSとDQNを用いたPAのフィードバック剥離制御手法の提案を行っている．また，小泉ら[17]やRabaultら[18]は，2次元円柱流れにおいて，深層強化学習によるカルマン渦の抑制を行っている．これらの研究は深層強化学習を用いた流れ制御の有効性を示してはいるが，2次元かつ比較的低レイノルズ数条件（ $Re \cong 10^2$ 程度）における Reynolds-Averaged Navier-Stokes（RANS）ベースの数値シミュレーションであり，流れ場の忠実度には限界がある．数値シミュレーションと深層強化学習を組み合わせた研究は，計算コストの制約から2次元計算や低レイノルズ数に留まる場合が多く，乱流遷移を伴うレイノルズ数領域における複雑な流体现象を議論する場合は，Large-Eddy Simulation（LES）のような高忠実なシミュレーションの実施が必要である．しかしながら，翼周りの剥離制御を対象として，深層強化学習と高忠実度数値解析を組み合わせた研究例は限られているのが現状である．

以上の背景から，本研究では翼周り流れを対象に，LESによる高忠実シミュレーションと深層強化学習を組み合わせたPAのフィードバック剥離制御手法を構築し，流れ場に応じたPAの駆動方法を探索するとともに得られた流れ場からその制御メカニズムについて議論することを目的とする．

2. 流れ場の設定と計算手法

2.1 流れ場の設定

本研究では先行研究[8]と同様に， $Re = 63,000$ のNACA0015翼周り流れを対象とする．このレイノルズ数は，UAVやMAVといった小型無人航空機などのスケールに相当する．翼はスパン方向に一様な形状を有する3次元形状である．流れ条件はマッハ数 $M_\infty = 0.2$ ，比熱比 $\gamma = 1.4$ ，プラントル数 $Pr = 0.72$ とする．図3に本条件の非制御時における数値シミュレーション結果を示す．図は，主流方向速度で色付けされた速度勾配テンソルの第二不変量の等値面であり，翼周り流れが前縁から剥離している状態を示している．PAは翼前縁5%の位置に設置する．迎角は失速角直後の12度と15度とする．

2.2 プラズマアクチュエータのモデル化

PA の流体計算への導入は支配方程式に PA を模擬した体積力項を導入することで行う． PA の体積力モデルはいくつか提案されているが[19, 20]，ここでは Suzen と Huang ら[21]の定常体積モデル（以下 Suzen モデル）を非定常に拡張した体積モデルを用いる． 図 4 に Suzen モデルの概形を示す． モデルの数値計算への導入方法の詳細，その信頼性，および数値計算の妥当性については既出論文を参照されたい[22, 23]． PA のパラメータは，先行研究[8,15]にあわせて無次元ベース周波数は $f_{base}U_\infty/c = 60$ ，バースト比 $BR = 0.1$ ，体積力項の大きさを決める運動量係数 D_C は 0.04 とした． ベース周波数は，実験における 6[kHz]に相当し，最大誘起速度は 3[m/s]で電圧は 6[kV_{pp}]，消費電力は 100[W/m]となる．

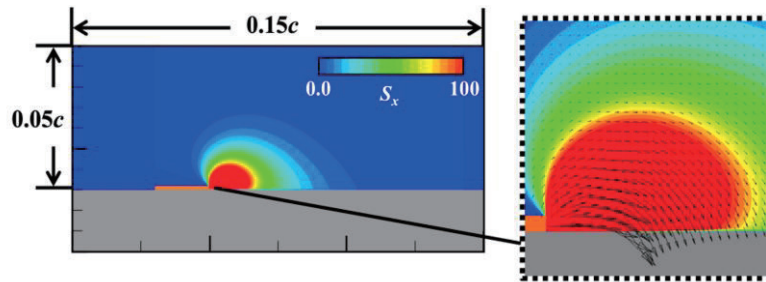


図 4 Suzen モデル

2.3 計算手法

3次元圧縮性 Navier-Stokes 方程式を支配方程式とし，空間の離散化には 6次精度コンパクト差分法[24]を用いた．フィルタリングには 10次精度陰的フィルタを適用し，フィルタ係数は $\alpha_f = 0.45$ とした．時間積分には内部反復を 5 回行う ADI-SGS 陰解法[25]を用いた．また，本研究では陽的なサブグリッドスケール (SGS) モデルを用いない Implicit LES を採用した．時間刻み幅は $dt = 2.0 \times 10^{-4}$ とした．

2.4 計算格子

図 5 に本研究で用いる計算格子を示す．翼周り流れを解く Zone1 および Zone2 と，アクチュエータ近傍の擾乱を解像する高解像度格子 Zone3 から構成される重合格子法を用いる．Zone1 には C 型格子を採用し，外部境界は翼弦長の 25 倍とする．Zone2 は翼後方の下流領域を対象とし，同様に外部境界は 25c とする．スパン方向長さは 0.2c とする．Zone3 には翼弦長 5%位置に設置したアクチュエータの体積力分布を内挿する．格子点数は，Zone1 が約 1.7×10^7 ，Zone2 が約 3.4×10^6 ，Zone3 が約 2.2×10^6 で総格子点数は約 2200 万点である．翼面直交方向の最小格子幅は約 0.00012c とし，翼面には固体壁条件，スパン方向には周期境界条件を適用する．

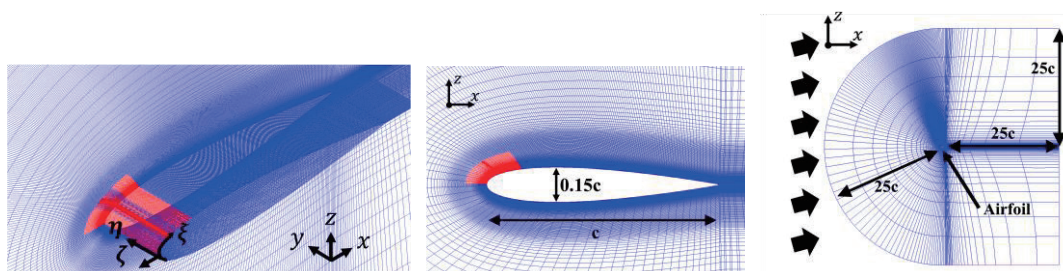


図 5 計算格子

2.5 計算機に関して

計算機として東北大学サイバーサイエンスセンターのスーパーコンピュータ AOBA のサブシステム AOBA-S を利用した。AOBA-S の 1 ノードは 1 ベクトルホスト(VH)と 8 ベクトルエンジン(VE)で構成されており、本計算では格子を 8 分割して 8VE を用いた。具体的には Zone1 を 5 分割, Zone2 を 2 分割した。VE 内はスレッド並列 (自動並列) とし, 各 VE 間は MPI 通信を行うハイブリッド並列を用いた。後述する深層強化学習は VH 側で行う。

3. 深層強化学習フレームワーク

一般的な強化学習は Q テーブルが用いられるが, 本研究では Q テーブルの代わりにニューラルネットワークを用いる DQN を採用する。DQN は行動価値関数をニューラルネットワークで近似し, 状態に対する最適な行動を選択する手法である。DQN アルゴリズムの詳細は[14]などを参照されたい。本研究では, 翼面上の圧力センサから得られる情報を状態として入力し, PA の駆動パラメータ (無次元バースト周波数 F^+) の選択を行動とした。構築した深層強化学習フレームワークを図 6 に示す。

まず, 機械学習を実行する Python プログラム (Agent) と数値シミュレーションを行う Fortran プログラム (Solver) を分離し, 両者が双方向通信を行う構成となっている。Solver は一定周期で環境状態を Agent に送信し, Agent は受信した環境に応じて行動を決定する。各プログラムの通信方式としてはファイルや標準入出力が利用できるようになっている。さらに, 図に示すように処理は大きく制御フェーズと学習フェーズの 2 つに分けられる。制御フェーズでは, Agent が Solver から環境状態を取得し行動選択を一定間隔で繰り返す。その際, DQN アルゴリズムのミニバッチ学習を行うために状態と行動, 行動の結果として得られる状態変化をバッファに蓄積していく。学習フェーズでは, 蓄積されたバッファからミニバッチを生成し, Agent のネットワークパラメータを更新する。制御フェーズと学習フェーズを 1 エピソードとし, 両フェーズを交互に繰り返す。制御フェーズと学習フェーズが終わったら (1 エピソードが終わったら), 環境をリセットする。具体的には初期の剥離状態の流れ場に戻す。Agent はエピソードを重ねていく中で試行錯誤を通じて最適な制御方策を獲得していくことが期待される。

図 6 の Sensor-A, B, C は疑似的な圧力センサであり, それぞれ翼面のスパン方向中央断面におけるコード長 0%, 40%, 100%位置に設置されている。Sensor-A, B の圧力係数を環境状態と

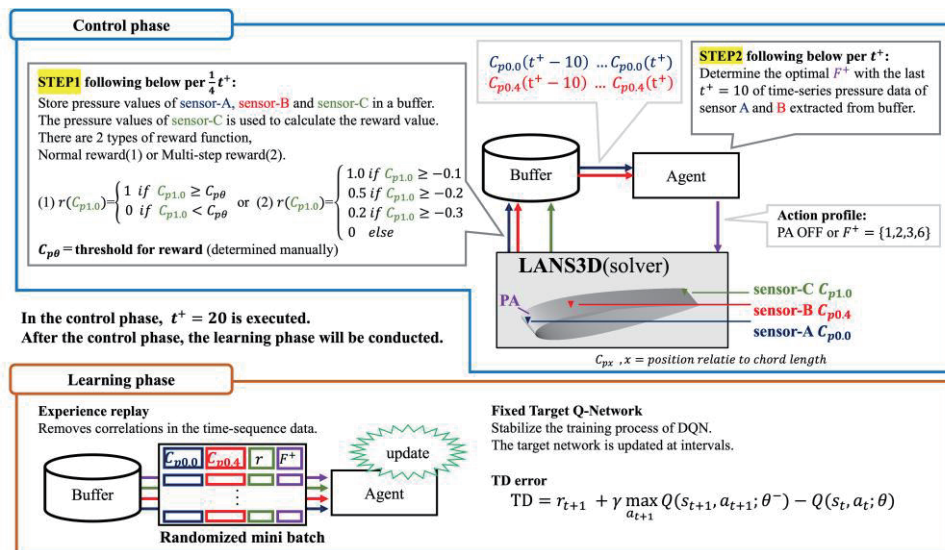


図 6 深層強化学習フレームワーク

して用い、後縁位置の Sensor-C は報酬として活用する。報酬はあらかじめ設定した閾値を越えたかどうかの 2 値で即時報酬を決定することとする。報酬として後縁の圧力係数を用いる理由は、流れの付着状況と後縁圧力には正の相関があるためである。流れが翼に沿って流れる（付着している）とき、翼の後縁に十分な運動量が供給され、翼後縁の圧力係数が上昇することから Sensor-C の圧力係数が閾値を越えた場合は「付着」、下回ったときは「剥離」しているとみなす。

図 7 に Agent のネットワーク構成を示す。ネットワークは 1 次元畳み込み層、バッチ正規化層、最大プーリング層、全結合層からなる。ネットワークへの入力（環境状態）は、Sensor-A, B の直近の 10 無次元時間の圧力履歴からそれぞれ 40 点サンプリングしたベクトル（80 次元）とする。ネットワークの出力（行動）は、 $F^+ = 1, 2, 3, 6$, PA-OFF の 5 種類であり、出力ベクトルは各行動の Q 値となっている。最も Q 値が高い行動（＝報酬期待値が高い行動）が選択され、ソルバに送信される。行動選択後、一定時間は選択された行動が継続される。

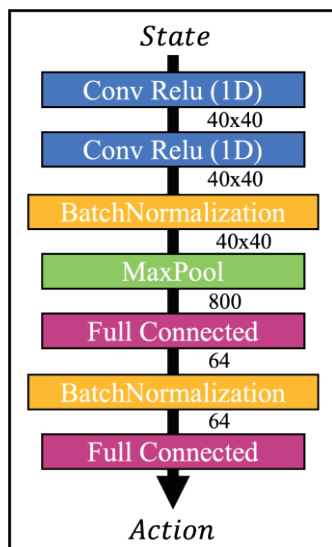


図 7 ネットワーク構成

表 1 DQN の学習パラメータ (Model-A)

Episode number	11
Action number per episode	20
Max memory size	240
Mini-batch size	20
Discount rate	0.99
Optimizer	Adam Optimizer
Target network update interval	3
Epsilon	0.9 to 0.1
Minimum of Epsilon	0.1
Learning rate	0.01
Reward	$C_{p\theta} = -0.09$
$C_{p\theta}$: threshold for reward	$r = \begin{cases} 1 & \text{if } C_{p1.0} \geq C_{p\theta} \\ 0 & \text{if } C_{p1.0} < C_{p\theta} \end{cases}$
$C_{p1.0}$: Pressure of sensor-C	

4. 迎角 12 度環境下でのフィードバック剥離制御

4.1 計算条件

迎角 12 度における DQN の学習パラメータを表 1 に示す。この設定のモデルをここでは Model-A と呼ぶ。1 エピソードあたりの行動変調回数は 20 回とし、各エピソードで一定時間ごとに行動を更新する。1 エピソードあたりの計算時間は AOBA-S で約 32 時間である。経験再生 (Experience Replay) のためのメモリサイズを 240、ミニバッチサイズは 20 とし、学習の安定化のために導入されるターゲットネットワークの更新間隔は 3 とした。割引率は 0.99 とし、将来の報酬も考慮した学習を行う。

行動選択には ϵ -greedy 選択を用い、 ϵ は 0.9 から 0.1 まで線形に減少させることで学習初期は探索を重視し、学習が進むにつれて報酬期待値の高い行動を選択する設定とした。最適化手法には Adam を用いた。学習率は大きすぎると学習が不安定になり、小さすぎると収束までに時間を要する。本研究では 0.01 とした。

報酬設計については、剥離の抑制し翼面上の圧力回復を促す制御方策の獲得を目的として、

Sensor-C の圧力係数 $C_{p1.0}$ が閾値 $C_{p\theta}$ を越えたら報酬 1, 越えない場合は 0 を与えることとした。 $C_{p\theta}$ は予備実験に基づき-0.09 とした。

4.2 制御結果

図 8 に各エピソードにおける翼後縁圧力係数の推移を示す。縦軸は圧力係数(C_p)、横軸は無次元時間 (t^+) である。黒線は閾値であり、閾値を上回っているときは即時報酬が得られている。図より、エピソード初期は閾値を下回る時間が多くみられるが、エピソードが進むにつれて圧力係数が全体的に回復し、閾値を上回る時間が増加していることがわかる。また、 $t^+ = 12$ におけるエピソード 9 とエピソード 11 の瞬間場の図 9 を示す。等値面は速度勾配テンソルの第二不変量で、主流速度成分で色付けしている。図からエピソード 9 ではまだ剥離している流れが、エピソード 11 では大きく改善されていることがわかる。

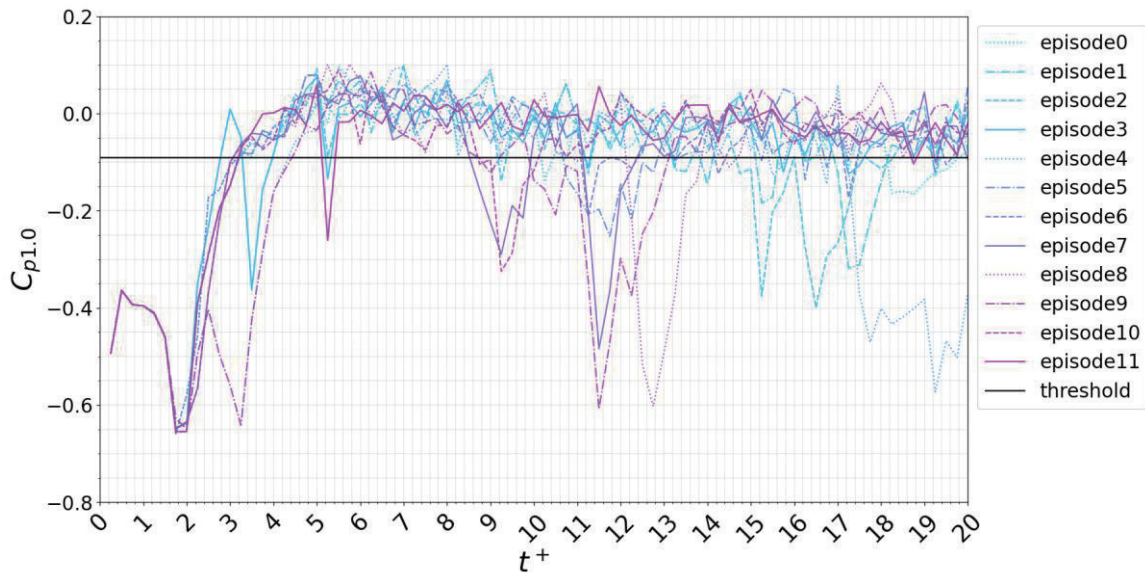


図 8 翼後縁圧力係数の推移

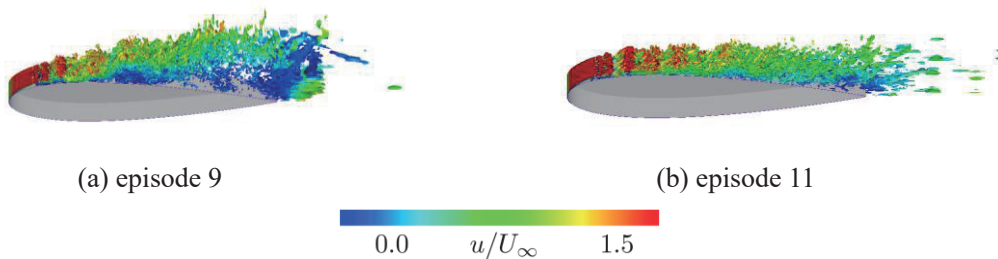


図 9 瞬間場

さらに翼の空力性能を表す揚抗比に注目する。図 10 に代表的なエピソードにおける揚抗比の推移を、図 11 に対応する行動履歴を示す。横軸はいずれも無次元時間である。図より、揚抗比はバースト周波数の選択と強く対応して変化していることが確認できる。特に、 $F^+ = 6$ が選択された後に揚抗比が急激に上昇しており、その後も $F^+ = 6$ が維持されている間は高い揚抗比が持続していることが分かる。これは、適切なバースト駆動によって流れ場が改善され、空力性能が向上していることを示唆している。一方で、 $F^+ = 6$ の選択頻度が低いエピソードでは揚抗比の向上が限定的であり、高い値に到達しない傾向が見られる。このことから、本モデルは学習の進行に伴い、高い揚抗比をもたらすバースト周波数を選択する方策が獲得できていると考えられる。

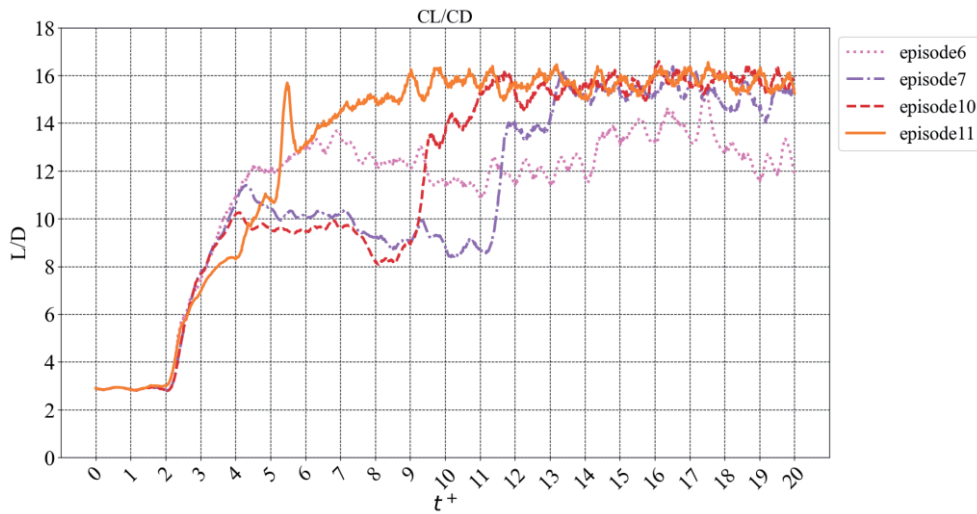


図 10 揚抗比の推移

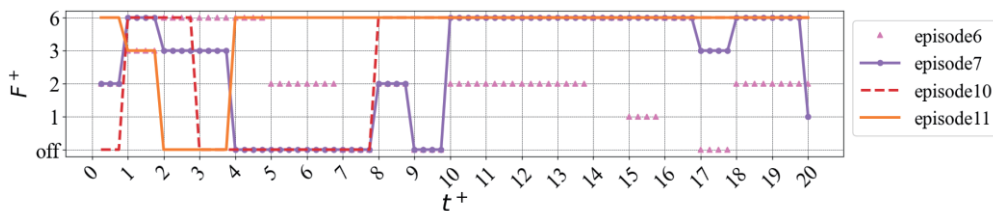


図 11 行動履歴

5. 迎角 15 度環境下でのフィードバック剥離制御

5.1 計算条件

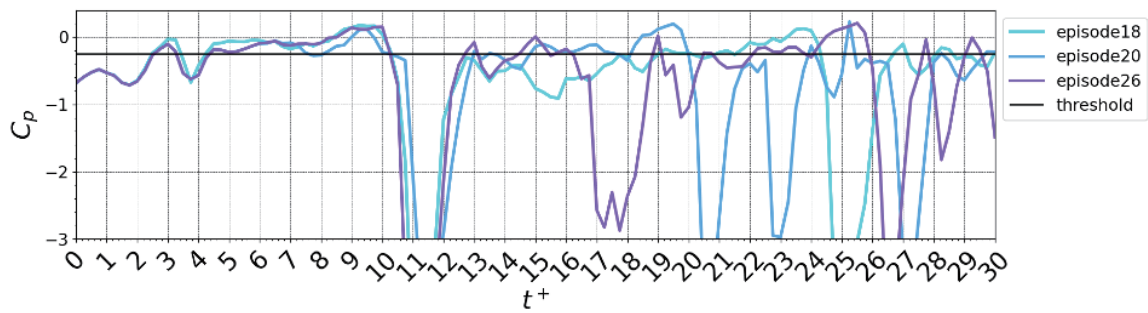
大規模な剥離が生じ剥離の抑制が難しい迎角 15 度における DQN の学習パラメータを表 2 に示す. ここでは, Model-A の設定を基にしつつ, 一部変更を加えたモデル (Model-B と呼ぶ) を検討した. Model-B では, 1 エピソードあたりの行動変調回数は 30 回とし, エピソード数を 26 に増加させより多くの探索を行う. 経験再生のためのメモリサイズは Model-A よりも増加させ 320, ミニバッチサイズも 32 とした.

学習率は 0.01 とした. 報酬設計については, Model-A と同様に剥離の抑制し翼面上の圧力回復を促す制御方策の獲得を目的として, Sensor-C の圧力係数 $C_{p1.0}$ が閾値 $C_{p\theta}$ を越えたら報酬 1, 越えない場合は 0 を与えることとし, $C_{p\theta} = -0.25$ とした.

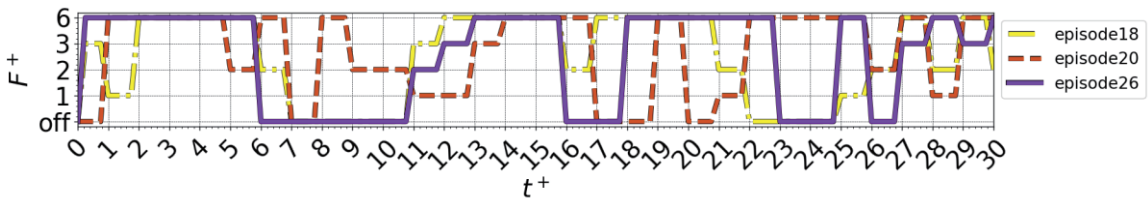
表 2 DQN の学習パラメータ (Model-B)

Episode number	26
Action number per episode	30
Max memory size	320
Mini-batch size	32
Discount rate	0.99
Optimizer	Adam Optimizer
Target network update interval	3
Epsilon	0.9 to 0.1
Minimum of Epsilon	0.1
Learning rate	0.01
Reward	$C_{p\theta} = -0.25$
$C_{p\theta}$: threshold for reward	$r = \begin{cases} 1 & \text{if } C_{p1.0} \geq C_{p\theta} \\ 0 & \text{if } C_{p1.0} < C_{p\theta} \end{cases}$
$C_{p1.0}$: Pressure of sensor-C	

5.2 制御結果



(a) 後縁圧力係数の推移



(b) 行動履歴

図 12 翼後縁圧力係数の推移と行動履歴 (Model-B)

図 12 に代表的なエピソードにおける翼後縁圧力係数の推移と行動履歴を示す. まず, 圧力係数が閾値を大きく下回る箇所が確認でき, このとき翼後縁において大きな剥離が生じていると考えられる. その後, PA が駆動することで圧力係数が一時的に回復し, 閾値付近まで戻る挙動が繰り返し確認できる. ここで注目すべき点として, 圧力係数が回復する前において一時的に PA を OFF にする挙動 (ここでは Nested Burst 駆動と呼ぶ) が複数のエピソードで確認できる. このことは, PA の連続的な駆動ではなく, 適度に間欠的な駆動を行うことで剥離抑制に有効な擾乱を与えている可能性を示唆している.

また, Model-B は 1 エピソード内の行動変調回数を増加させたことでより細かな駆動パラメータの変更が可能となり, 圧力係数が回復する機会が増加していると考えられる. しかし, 圧力係数の急激な低下を抑えることはできておらず, 大規模な剥離が生じる迎角 15 度環境下では継続的な剥離抑制には至っていない.

図 13 に揚抗比の推移を示す. 図より, $t^+ = 5 \sim 7$ にかけて揚抗比が大きく向上した後, 揚抗比が低下していき, その後再び回復に向かっていることが確認できる. 揚抗比は時間的に大きく変動しており, 安定的に剥離を抑制した状態を維持できていないことがわかる. 図 14 に示す瞬間場に着目すると, 揚抗比が高い $t^+ = 6$ では翼面上において剥離が抑制された流れ場となっているの

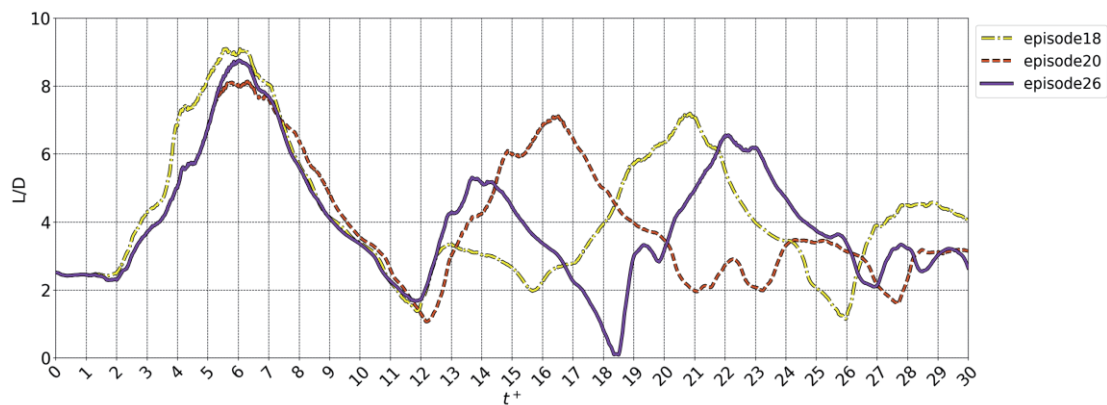


図 13 揚抗比の推移

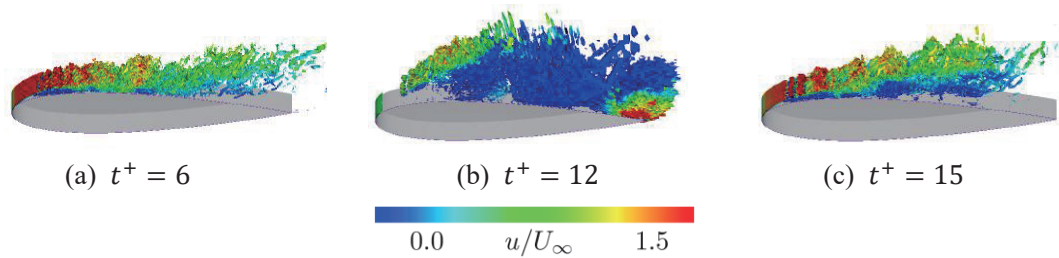


図 14 瞬間場 (episode20)

に対し、揚抗比が低下する $t^+ = 12$ では翼後縁に大規模な剥離構造が発達している状況が確認できる。一方、 $t^+ = 15$ では再び剥離領域が縮小傾向にあり、それに伴い揚抗比が再び増加していることがわかる。図 12 の行動履歴と対応づけると、Nested Burst 駆動により PA を一時的に OFF とすることで、翼面近傍に形成された剥離せん断層が一旦翼面から離脱し、再度 PA を ON とした際に大規模な巻き込み渦が生成されることで、流れ場に強い擾乱が付与されると考えられる。この擾乱が剥離流れの再構成を促進し、結果として揚抗比の回復につながっていると考えられる。

6. まとめ

本研究では、NACA0015 翼周り流れを対象に、LES による高忠実数値シミュレーションと深層強化学習 (DQN) を組み合わせた PA のフィードバック剥離制御フレームワークを構築した。本手法を迎角 12 度および迎角 15 度の条件に適用し、その有効性の検証を行った。

迎角 12 度においては、エピソードの進展に伴い、揚抗比を向上させる最適な無次元バースト周波数 ($F^+ = 6$) を選択する制御方策が獲得された。翼後縁圧力が回復し、揚抗比が継続的な向上が確認され、先行研究と概ね一致する結果となった。

一方、迎角 15 度は大規模な剥離が生じる流れ場であることから、揚抗比は一時的に向上するものの変動が大きく、高い揚抗比を安定的に維持するには至らなかった。流れ場の解析から、剥離の付着と再剥離を繰り返す挙動が確認されたものの、PA の駆動において、連続的なバースト駆動ではなく、バースト駆動の ON と OFF を切り替える間欠的な制御 (Nested Burst 駆動) が見られた。PA を一時的に OFF とすることで、剥離せん断層を翼面から離し、再起動時に大規模な渦構造を誘起することで揚抗比の回復につながっていると考えられる。このことは、単一の周波数制御に加えて ON と OFF を切り替える間欠的な制御戦略の重要性を示している。

しかしながら、現状の試行回数および学習設定では、このような制御を安定して維持する方策の獲得には至っておらず、安定的な剥離抑制の実現にはさらなる学習手法の改良が必要である。

7. 今後の研究計画

本研究では、失速直後の迎角 12 度においては本手法の有効性が確認できたが、大規模な剥離が生じる迎角 15 度ではその効果が限られている。DQN のような価値ベース手法では、状態に対して単一の行動が選択される枠組みのため、柔軟な制御方策を獲得することが難しく、十分な制御効果が得られない可能性がある。今後は、用いる環境情報や行動、学習方法などの改良を進めるとともに、学習コストの増加が伴うものの、価値関数と方策を同時に学習する Actor-Critic や過去の状態や行動履歴の時系列を扱える Transformer などのアーキテクチャを導入し、間欠的な ON/OFF 切り替えを含む制御戦略の性能向上を図る予定である。

謝辞

ここに記載させて頂いた成果は、東北大学サイバーサイエンスセンターのスーパーコンピュータを利用することで実現することができたものである。また、研究にあたっては同センター関係各位に有益なご指導とご協力を頂いた。センターの皆様はこの場を借りて謝意を示したい。

参考文献

- [1] J. R. Roth, D. M. Sherman, and S. P. Wilkinson, “Electrohydrodynamic Flow Control with a Glow-Discharge Surface Plasma,” *AIAA Journal*, Vol. 38, No. 7, pp. 1166–1172, 2000.
- [2] T. C. Corke, M. L. Post, and D. M. Orlov, “Single Dielectric Barrier Discharge Plasma Enhanced Aerodynamics: Physics, Modeling and Applications,” *Experiments in Fluids*, Vol. 46, No. 1, pp. 1–26, 2009.
- [3] T. C. Corke, C. L. Enloe, and S. P. Wilkinson, “Dielectric Barrier Discharge Plasma Actuators for Flow Control,” *Annual Review of Fluid Mechanics*, Vol. 42, pp. 505–529, 2010.
- [4] K. Asada, Y. Ninomiya, A. Oyama, and K. Fujii, “Airfoil Flow Experiment on the Duty Cycle of DBD Plasma Actuator,” *AIAA Aerospace Sciences Meeting*, 2009.
- [5] K. Fujii, “High-Performance Computing Based Exploration of Flow Control with Micro Devices,” *Philos Trans A Math Phys Eng Sci*, Vol. 372, No. 2022, p. 20130326, 2014.
- [6] M. Sato, H. Aono, A. Yakeno, T. Nonomura, K. Fujii, K. Okada, and K. Asada, “Multifactorial Effects of Operating Conditions of Dielectric-Barrier-Discharge Plasma Actuator on Laminar-Separated-Flow Control,” *AIAA Journal*, Vol. 53, No. 9, pp. 2544–2559, 2015.
- [7] H. Aono, S. Sekimoto, M. Sato, A. Yakeno, T. Nonomura, and K. Fujii, “Computational and Experimental Analysis of Flow Structures Induced by a Plasma Actuator with Burst Modulations in Quiescent Air,” *Mechanical Engineering Journal*, Vol. 2, Issue 4, p. 15-00233, 2015.
- [8] Makoto Sato, Koichi Okada, Kengo Asada, Hikaru Aono, Taku Nonomura, and Kozo Fujii. Unified Mechanisms for Separation Control Around Airfoil Using Plasma Actuator with Burst Actuation over Reynolds Number Range of 10^3 - 10^6 ,” *Physics of Fluids*, Vol. 32, No. 2, p. 025102, 2020.
- [9] T. Ogawa, K. Asada, S. Sekimoto, T. Tatsukawa, and K. Fujii, “Dynamic Burst Actuation to Enhance the Flow Control Authority of Plasma Actuators,” *Aerospace*, Vol. 8, No. 12, Dec. 2021.
- [10] S. L. Brunton, B. R. Noack, and P. Koumoutsakos, “Machine Learning for Fluid Mechanics,” *Annual Review of Fluid Mechanics*, Vol. 52, No. 1, pp. 477–508, 2020.
- [11] N. Benard, P. Sujar-Garrido, J.-P. Bonnet, and E. Moreau, “Control of the Coherent Structure Dynamics Downstream of a Backward Facing Step by DBD Plasma Actuator,” *International Journal of Heat and Fluid Flow*, Vol. 61, pp. 158–173, 2016.
- [12] B. F. Zhang, D. W. Fan, and Y. Zhou, “Artificial Intelligence Control of a Low-Drag Ahmed Body Using Distributed Jet Arrays,” *Journal of Fluid Mechanics*, Vol. 963, p. A3, 2023.
- [13] J. Chen, H. Zong, H. Song, Y. Wu, H. Liang, and Z. Su, “Closed-Loop Plasma Flow Control of a Turbulent Cylinder Wake Flow Using Machine Learning at Reynolds Number of 28000,” *Physics of Fluids*, Vol. 36, No. 1, 2024.
- [14] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, “Human-Level Control Through Deep Reinforcement Learning,” *Nature*, Vol. 518, pp. 529-533, 2015.
- [15] S. Shimomura, S. Sekimoto, A. Oyama, K. Fujii, and H. Nishida, “Closed-Loop Flow Separation Control Using the Deep Q Network over Airfoil,” *AIAA Journal*, Vol. 58, No. 10, pp. 4260-4270, 2020.
- [16] N. Takada, A. Watanabe, S. Shimomura, and H. Nishida, “Study on High-frequency Control aiming for Improving Deep Reinforcement Learning-based Flow Separation Control over an Airfoil,” *AIAA Aviation Forum*, 2023.

- [17] H. Koizumi, S. Tsutsumi, and E. Shima, "Feedback Control of Karman Vortex Shedding from a Cylinder using Deep Reinforcement Learning," AIAA Flow Control Conference, 2018.
- [18] J. Rabault, M. Kuchta, A. Jensen, U. Réglade, and N. Cerardi, "Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control," *Journal of Fluid Mechanics*, Vol. 865, pp. 281-302, 2019.
- [19] W. Shyy, B. Jayaraman, and A. Andersson, "Modeling of Glow Discharge-Induced Fluid Dynamics," *Journal of Applied Physics*, Vol. 92, No. 11, pp. 6434-6443, 2002.
- [20] C. L. Enloe, Thomas E. McLaughlin, Robert D. VanDyken, K. D. Kachner, Eric J. Jumper, and Thomas C. Corke, "Mechanisms and responses of a Single Dielectric Barrier Plasma Actuator: Geometric Effect," *AIAA Journal*, Vol. 42, No. 3, pp. 595-604, 2004.
- [21] Y. B. Suzen and P. G. Huang, "Simulations of Flow Separation Control using Plasma Actuators," 44th AIAA Aerospace Sciences Meeting and Exhibit, Aerospace Sciences Meetings, Reno, Nevada, January 2006.
- [22] K. Asada, T. Nonomura, H. Aono, M. Sato, K. Okada, and K. Fujii, "LES of Transient Flows Controlled by DBD Plasma Actuator over a Stalled Airfoil," *International Journal of Computational Fluid Dynamics*, Vol. 29, 2015.
- [23] H. Aono, S. Sekimoto, M. Sato, A. Yakeno, T. Nonomura, and K. Fujii, "Computational and Experimental Analysis of Flow Structures Induced by a Plasma Actuator with Burst Modulations in Quiescent Air," *Bulletin of the JSME Mechanical Engineering Journal* Vol. 2, No. 4, 2015.
- [24] Sanjiva K. Lele, "Compact Finite Difference Schemes with Spectral-like Resolution," *Journal of Computational Physics*, Vol. 103, No. 1, pp. 16-42, 1992.
- [25] K. Fujii, "Simple Ideas for the Accuracy and Efficiency Improvement of the Compressible Flow Simulation Methods," *International CFD Workshop for Super-Sonic Transport Design*, Tokyo, 1998.

[共同研究成果]

Level set 法による相変化解析手法の構築・検証と

円管内気泡への適用

平井克樹、川本裕樹
東海大学

行光実桜、佐々木大輔
大阪公立大学

高橋俊、福家英之
宇宙航空研究開発機構

ヒートパイプ内部の相変化を伴う気液二相流の流動メカニズムの把握と熱輸送性能向上に向け、等間隔直交格子および Level set 法に基づく数値解析手法を構築した。三次元重力下における上昇気泡解析により手法の妥当性を検証した結果、気泡形状、過熱度分布および上昇挙動について先行研究と良好な一致が得られた。また円管内の単一気泡解析から、気泡成長に伴う形状遷移およびテイラーバブル形成過程と、それに伴う熱輸送特性の変化を明らかにした。

1. 序論

ヒートパイプは、密閉容器内に封入した作動流体の蒸発・凝縮による相変化を利用し、小さな温度差でも効率よく熱を輸送するデバイスである。この高い熱輸送性能から、宇宙機器をはじめとする様々な分野で利用されている。一方、ヒートパイプは密閉構造であるため内部流動の直接観察や詳細な物理量の計測が困難であり、内部の流動および熱輸送メカニズムの詳細の把握には数値流体力学 (CFD) が有効であると考えられる。しかし、ヒートパイプ内部では作動流体の蒸発・凝縮を伴う気液二相流が生じるため、界面移動や界面近傍の温度場を高精度に扱う必要があり、解析は容易ではない。これまで、Sharp interface model や Ghost fluid 法などを用いた相変化解析手法が提案され、その有効性が示されている^[1, 2]。その一方で、これらの手法を用いて実際の熱輸送機器内部における流れを詳細に解析した研究は限られている。

本研究の目的は、CFD を用いてヒートパイプ内部における相変化を伴う気液二相流の流動および熱輸送メカニズムの詳細を明らかにし、高効率な熱輸送の実現に資する知見を得ることである。特に円管の直径は、気泡界面形状や壁面液膜厚さを通じて熱伝達特性に影響を及ぼす重要な因子である。本報では、等間隔直交格子および Level set 法を用いた数値解析の検証と、円管内における単一気泡の成長挙動と熱伝達特性の管径依存性を調査した結果について報告する。

2. 数値計算法

本研究では等間隔直交格子上において、気液二相流と熱輸送を取り扱う。支配方程式は相変化を考慮した連続の式、非圧縮性 Navier–Stokes 方程式、エネルギー方程式である。また気液界面の追跡には符号付き距離関数を用いる Level set 法を適用し、密度および粘度は Level set 関数から計算される Heaviside 関数を用いて界面近傍で平滑化する。また、表面張力は CSF (Continuum Surface Force) モデルにより Navier–Stokes 方程式の外力項として与える。ここで連続の式および非圧縮性 Navier–Stokes 方程式をそれぞれ式(1)と式(2)に示す。

$$\nabla \cdot \mathbf{u} = \dot{m} \left(\frac{1}{\rho_v} - \frac{1}{\rho_l} \right) \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{1}{\rho} \nabla p + \frac{1}{\rho} \nabla \cdot \{\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T)\} + \frac{\sigma \kappa \nabla H}{\rho} + \mathbf{g} \quad (2)$$

ここで、 \mathbf{u} は流速、 ρ_v および ρ_l はそれぞれ気相および液相の密度、 \dot{m} はセル内の相変化量、 p は圧力、 μ は粘度、 σ は表面張力、 κ は曲率、 H は Heaviside 関数、 \mathbf{g} は重力加速度である。また温度場の計算には、式(3)に示すエネルギー方程式を用いた。

$$\rho C_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = k \nabla^2 T \quad (3)$$

ここで、 C_p は定圧比熱、 T は温度、 k は熱伝導率である。なお、界面近傍の拡散項は気液界面を飽和温度として扱うため、不等間隔差分に基づく 2 次精度中心差分法により評価した。さらに、気液界面の追跡には Level set 法を用い、Level set 関数 ϕ の移流方程式を式(4)に示す。

$$\frac{\partial \phi}{\partial t} + \left(\mathbf{u} - \frac{\dot{M}}{\rho_v} \mathbf{n} \right) \cdot \nabla \phi = 0 \quad (4)$$

ここで、 $\phi = 0$ は気液界面、 $\phi > 0$ は気相、 $\phi < 0$ は液相、 \mathbf{n} は界面の単位法線ベクトルである。界面を通過する質量流束 \dot{M} は、液相側の熱流束に基づき式(5)より算出した。

$$\dot{M} = -\frac{k_l \nabla T_l \cdot \mathbf{n}}{L} \quad (5)$$

ここで、 k_l は液相の熱伝導率、 T_l は液相温度、 L は潜熱である。なお、連続の式における相変化量 \dot{m} は、界面での質量流束 \dot{M} を界面の面積に基づいてセル内の体積生成項へ換算した値である。

時間発展には Fractional Step 法を用いた。流動計算における移流項には 5 次精度 WENO 法、粘性項には 2 次精度中心差分法、時間積分には 3 次精度 TVD Runge–Kutta 法を適用した。圧力 Poisson 方程式の解法には Bi-CGSTAB 法を用いた。温度場については、移流項に 3 次精度風上差分法、拡散項に 2 次精度中心差分法、時間積分に Crank–Nicolson 法を適用した。また円管形状の定義にも Level set 法を用い、境界条件は Image point を用いる埋め込み境界法^[3]により与えた。

3. 結果および考察

3.1 検証解析

解析手法の妥当性を確認するため、先行研究^[1]に基づき三次元において重力下の過熱液相内を上昇する気泡の解析を実施した^[4]。作動流体はエタノールとし、計算領域は $X=Y=8$ mm、 $Z=15$ mm とした。気泡の初期半径は $R=210$ μm 、初期位置は底面から 1 mm の中心とし、重力加速度は 9.81 m/s^2 とした。また、格子幅の影響を確認するため、Coarse、Medium、Fine の 3 種類の格子を設定した。格子幅 Δx はそれぞれ 40、20、16 μm であり、対応する格子数は $201 \times 201 \times 376$ 、 $401 \times 401 \times 751$ 、 $501 \times 501 \times 939$ である。ここで図 1 に各格子解像度における過熱度分布を示す。気泡の上昇に伴い、気泡下部には尾を引くような過熱度分布が形成され、気泡形状は球形から楕円球状へと変化した。また、速度場には気泡端部を中心とした循環流れが確認された。Medium および Fine では、これらの気泡形状、過熱度分布および流れ場の特徴が先行研究と同様の傾向を示し、定性的に良好な一致が得られた。一方、Coarse では他の条件と比較して気泡径が小さく、過熱度分布にも差異が見られた。これは、本解析では界面厚さを格子幅 4 点で与えているため、格子幅が大きい条件ほど人工的な界面厚さの影響が相対的に大きくなり、界面近傍の物理量勾配に差が

生じたためと考えられる。さらに、気泡上昇速度および正規化気泡半径の時刻歴についても、格子幅を細かくすることで先行研究に近い傾向が得られた。以上より、本解析手法は重力下における上昇気泡の相変化挙動を妥当に再現できていると考えられる。

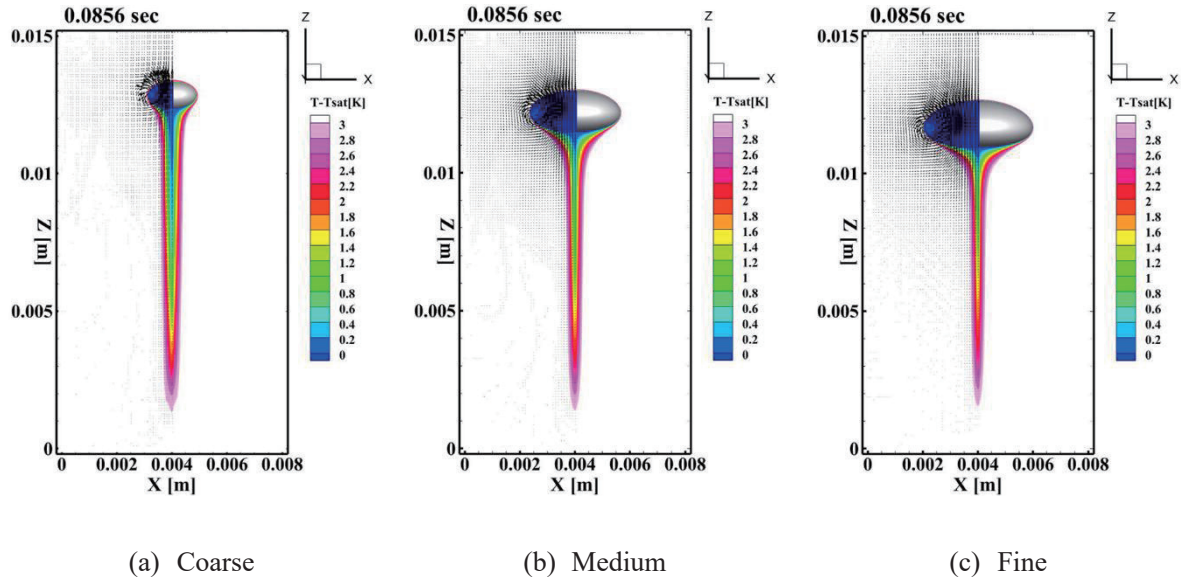


図 1 0.0856 s における各格子解像度での過熱度分布と速度ベクトル

3.2 円管内の単一気泡解析

本節では、3.1 節の検証解析に円管を適用し、ヒートパイプに近い条件を想定した三次元円管内における単一気泡の成長挙動を解析した。解析対象の円管は管径 $D=2$ mm、高さ $H=20$ mm とした。計算格子には完全等間隔の直交格子を用い、作動流体にはエタノールを採用した。初期条件として、気液は静止状態から開始し、気泡は管軸中心上の底面から 1 mm の位置に配置した。気泡の初期直径は約 $0.10D$ とした。また、流入流速は 0.05 m/s に設定し、重力加速度は 9.8 m/s² とした。温度条件として、液相温度は作動流体の大気圧下における飽和温度 T_{sat} に過熱度 3.1 K を加えた値とし、気泡温度は T_{sat} とした。さらに、壁面温度は液相温度よりも高くなるよう設定し、液体との温度差が 10.1 K となる条件とした。

図 2 に加熱度分布の時間変化を示す。気泡は上昇しながら成長しており、気泡直径が管径より小さい段階では、高さ方向および半径方向の両方に成長する様子が確認できる。一方、気泡直径が管径と同程度になると、成長は主に高さ方向に限定され、気泡はテイラーバブルへと変形しながら上昇する挙動を示した。図 3 に 0.080 s における壁面近傍の局所ヌセルト数を示す。気泡領域におけるヌセルト数は液体領域の約 3 倍に達しており、気泡の存在が熱伝達の向上に寄与していることが確認された。

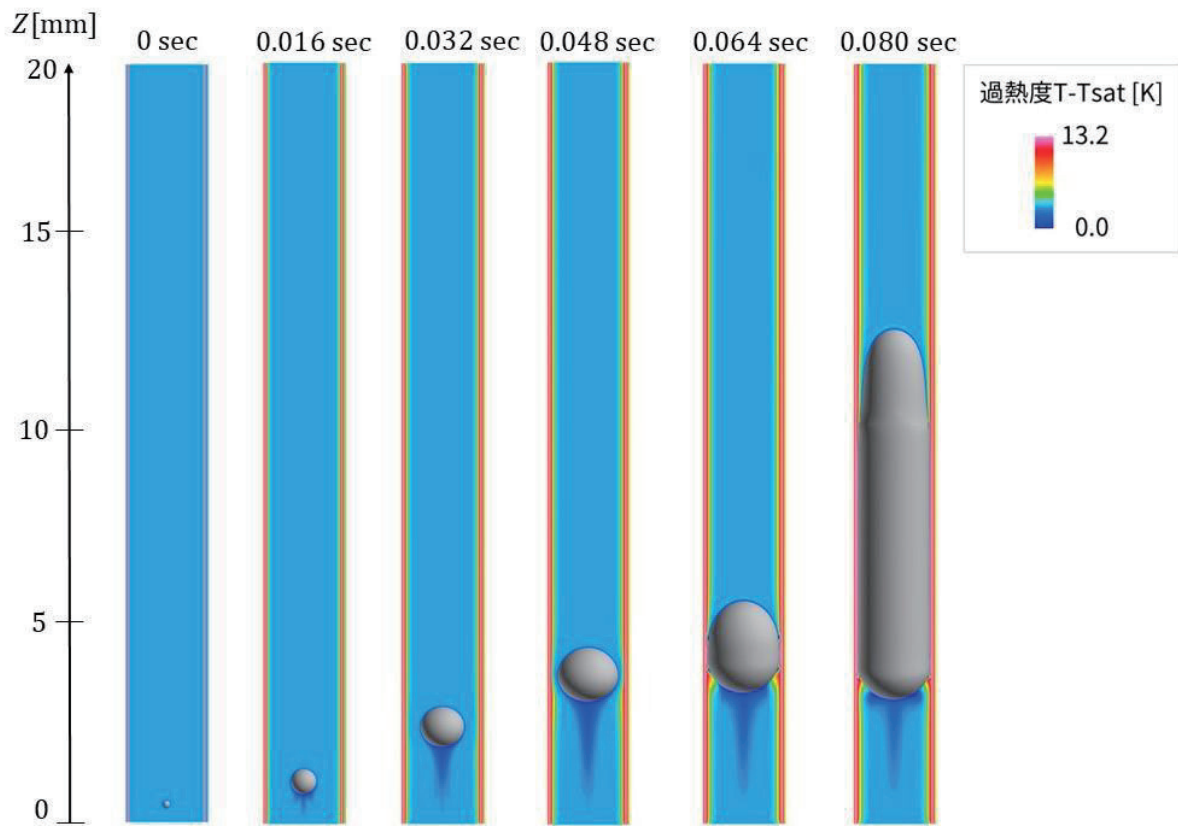


図2 管径 $D=2\text{mm}$ 、高さ $H=20\text{mm}$ における過熱度分布の時刻歴

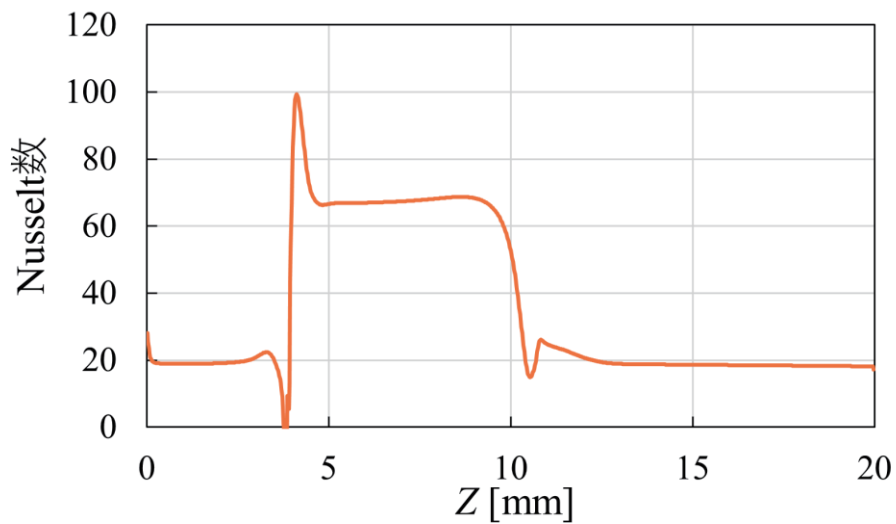


図3 0.080sec における壁面近傍の局所ヌセルト数

4. 結言

ヒートパイプ内部の相変化を伴う気液二相流解析に向けて、Level set 法を用いた数値解析手法を構築した。さらに、重力下における上昇気泡の三次元解析を通じて、提案手法の妥当性を検証した。その結果、気泡形状、過熱度分布および上昇挙動について先行研究と良好な一致が得られ、本手法が相変化を伴う気液二相流の解析に有効であることを確認した。円管内の単一気泡解析においては、気泡径の増大に伴い球状気泡からテイラーバブルへと遷移し、流動様式が変化することを確認した。また、気泡の存在により壁面近傍の熱伝達が促進され、ヌセルト数が増加することを明らかにした。今後は、大規模解析に向けた新たな手法を適用し、より実機条件に近い複雑な流動場の解析や、熱輸送性能の定量評価を通じて、ヒートパイプの高性能化に向けた設計指針の構築に展開する予定である。

謝辞

本研究は、東北大学サイバーサイエンスセンターのスーパーコンピュータを利用することで実現することができた。また、研究にあたっては同センター関係各位に有益なご指導とご協力をいただいた。

参考文献

- [1] Sato, Y., and Ničeno, B., “A sharp-interface phase change model for a mass-conservative interface tracking method,” *Journal of Computational Physics*, 249 (2013), pp. 127–161.
- [2] Tanguy, S., Sagan, M., Lalanne, B., Couderc, F., and Colin, C., “Benchmarks and numerical methods for the simulation of boiling flows,” *Journal of Computational Physics*, 264 (2014), pp. 1–22.
- [3] Takahashi, S., Nonomura, T., and Fukuda, K., “A numerical scheme based on an immersed boundary method for compressible turbulent flows with shocks: Application to two-dimensional flows around cylinders,” *Journal of Applied Mathematics*, 2014 (2014), Article ID 252478.
- [4] Hirai, K., Kawamoto, Y., Nara, S., Takahashi, S., Kawachi, A., Okazaki, S., and Fuke, H., “Numerical Simulation of Gas-Liquid Two-Phase Flow with Phase Change for Engineering Applications”, *Proceedings of the 12th International Conference on Computational Fluid Dynamics*, (2024).

[大規模科学計算システム] 【AOBA-S の利用法】

サブシステム AOBA-S の利用法

情報部デジタルサービス支援課

1 はじめに

本センターはスーパーコンピュータ AOBA のサブシステム AOBA-S の運用を 2023 年 8 月から開始しています。本稿では、サブシステム AOBA-S のプログラミング利用ガイドとして、AOBA-S 用フロントエンドサーバへのログイン、プログラムの作成からコンパイル、およびジョブ実行等の使い方についてご紹介します。

サイバーサイエンスセンターの大規模科学計算システムを利用するためには利用申請が必要です。利用申請について詳しくは「利用申請」(<https://www.ss.cc.tohoku.ac.jp/apply-for-use/>) ご参照ください。

2 システムの構成

本センターでは、スーパーコンピュータ AOBA として、サブシステム AOBA-S (SX-Aurora TSUBASA Type 30A) と、サブシステム AOBA-A (SX-Aurora TSUBASA Type 20B)、およびサブシステム AOBA-B (LX 406Rz-2) の 3 つの計算機システムをサービスしています (図 1)。また、AOBA-S 用のストレージとして 5.6PB、AOBA-A および AOBA-B 用のストレージとして 2PB のストレージシステムをサービスしています。利用者は SINET6 を利用したりリモートアクセス接続により、全国から大規模科学計算システムを利用することが可能です。

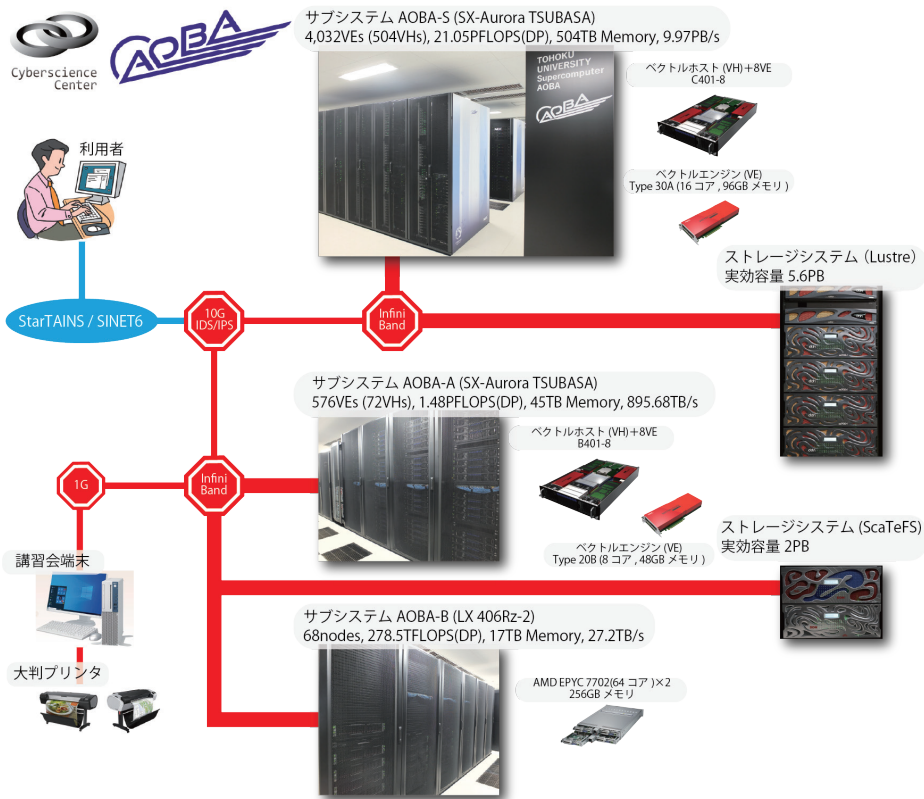


図 1 スーパーコンピュータ AOBA の構成

2.1 サブシステム AOBA-S の特徴

2.1.1 SX-Aurora TSUBASA アーキテクチャ

サブシステム AOBA-S はサブシステム AOBA-A と同じくベクトルアーキテクチャを継承しています。アプリケーション演算処理を行うベクトルエンジン（以下、VE）部と、主に OS 処理を行うベクトルホスト（以下、VH）部により構成されます。PCIe カードに搭載される VE 部はベクトルプロセッサ、および高速メモリから構成され、x86/Linux である VH と PCIe 経由で接続されます。

1VE は理論最大演算性能 4.91TFLOPS(DP) となる 16 コアベクトルプロセッサを 1 基、主記憶は 96GB を搭載し、2.45TB/s という高いメモリバンド幅でプロセッサと接続されることで、高い演算性能とメモリ性能の最適化を実現しています。

今回導入した AOBA-S システムは、1VH と 8VE が構成単位となる C401-8 モデルを採用し、システム全体では 504 個の VH と 4,032 個の VE で構成されます。VE と VH を合わせたシステム全体の理論演算性能は、21.05PFLOPS(DP)、主記憶は 504TB、総メモリバンド幅は 9.97PB/s となります。



図2 サブシステム AOBA-S (32 ラック)

- マルチコアベクトルエンジン（16 コア）あたり、4.91TFLOPS(DP) の理論演算性能
- 1VE あたり 96GB の共有メモリを搭載し、メモリバンド幅（データ転送性能）は 2.45TB/s

2.1.2 ベクトルプロセッサ

ベクトルプロセッサとは、ベクトル演算を行う専用のハードウェアを持つコンピュータです。ベクトル演算は、ループ中で繰り返し処理されるような配列データの演算に対して一括して演算を実行するため、高速に演算することができます。

ベクトル計算機は科学技術用の数値計算に適しており、大量のデータを繰り返し処理するような大規模計算に向いていると言われています。本センターでは、流体解析や気象解析、電磁界解析をはじめとする大規模シミュレーションに利用されます。

2.1.3 ソフトウェア

Linux OS 環境上で、ベクトルエンジンの開発環境を利用できます。ベクトル処理、並列処理に対応した大規模プログラムの作成と実行が可能です。

■**プログラミング言語** GNU 互換環境を装備し、アプリケーションの実効性能を向上させる高度な自動ベクトル化・自動並列化機能を備えた Fortran/C/C++ コンパイラが利用できます。それぞれのコンパイラは自動ベクトル化機能、自動並列化機能を有していますので、既存のプログラムを修正することなくベクトル化、並列化することができます。自動並列化機能と OpenMP による共有メモリ並列実行と、システム構成に最適化された MPI ライブラリにより、分散メモリ並列実行も可能です。

■**科学技術計算ライブラリ** NEC Numeric Library Collection (NLC) は業界標準の BLAS、FFTW、LAPACK、ScaLAPACK を含む、最適化された科学技術計算ライブラリです。NLC は広範な分野の数値シミュレーションプログラムの作成を強力に支援する数学ライブラリのコレクションで、VE での実行に最適化されています。NLC を用いることにより、難解な数値計算アルゴリズムの詳細に煩わされることなく高度な科学技術計算プログラムを作成することができ、数値シミュレーションプログラム開発の生産性を大幅に改善することができます。NLC は、Fortran または C 言語プログラムから利用できます。

2.2 プログラムの実行方法

プログラムの実行の方法として表 1 の 4 種類が利用できます。並列実行には 3 つの手法があります。

■**逐次実行** 単一のコアで動作するプログラムです。VE 内には 16 コアがありますが、逐次実行では 1 コアのみが利用されます。メモリは 96GB まで利用できます。

■**自動並列実行** 逐次処理プログラムを、コンパイラが並列化可能な箇所を自動的に判断して並列化します。プログラムを改めて書き直す必要はなく、既存のプログラムをそのまま利用することができます。VE 内の 16 コアまでの並列実行が可能で、メモリは 96GB まで利用できます。

■**OpenMP 並列実行** 自動並列化と同じく逐次処理プログラムを並列化します。並列化の判断は自動ではなくユーザが明示的に行います。ソース中の並列化したい箇所に並列化指示行を追加します。VE 内の 16 コアまでの並列実行が可能で、メモリは 96GB まで利用できます。

■**MPI 並列実行** MPI ライブラリによりプロセッサ間通信を行います。データの分割、処理方法等の並列処理手順を明示的に記述した MPI プログラムを作成する必要があります。分散メモリ並列実行が可能なので、複数の VE を用いた実行が可能です。

センターでは通常のサービスとして最大 2,048VE (32,768 コア)、メモリは 192TB まで利用できます。MPI 並列と自動並列/OpenMP 並列を同時に利用した並列実行も可能です。

表 1 実行の種類、特徴、最大並列数、最大メモリ量

実行の種類	並列化の方法	コードの改変量	最大並列数	最大メモリ量
逐次実行	-	-	1 コア	96GB
自動並列化	コンパイラによる自動並列化	なし	16 コア	96GB
OpenMP 並列	ユーザによる指示行挿入	少ない	16 コア	96GB
MPI 並列	MPI ライブラリを用いたプログラミング	多い	32,768 コア	192TB

3 利用者向けサーバへのログイン方法

AOBA-S 用のフロントエンドサーバ、およびデータ転送サーバへのログインは、AOBA-A,B 用のログインサーバと同じく、公開鍵認証方式による SSH 接続を採用しています。

公開鍵認証方式で使用する鍵ペアの作成方法と、各サーバへのログイン方法についてご紹介します。解説では SSH 接続に以下のターミナル（端末）ソフトを使用する例をご紹介します。

- （Windows の場合） Windows PowerShell
- （macOS / Linux の場合） ターミナル

本センターのシステムをはじめて利用する方は以下の手続きが必要です。

- (1) 利用者番号の取得（利用申請：<https://www.ss.cc.tohoku.ac.jp/apply-for-use/>）
- (2) 鍵ペアの作成（3.3 節）

AOBA-A および AOBA-B を利用していた方は、(1) (2) の手続きは不要です。以前使用していた利用者番号および鍵ペアをそのままご利用いただけます。3.4 節からお読みください。

3.1 利用者向けサーバ名と用途

表 2 に AOBA-S 用の利用者向けサーバ名とホスト名、および用途を示します。

表 2 AOBA-S 用の利用者向けサーバ

サーバ名	ホスト名	用途
フロントエンドサーバ	sfront.cc.tohoku.ac.jp	コンパイル作業、AOBA-S へのジョブ投入 ローカル PC との小規模なデータ転送
データ転送サーバ	sfile.cc.tohoku.ac.jp	ローカル PC との大規模なデータ転送 AOBA-A,B 用ストレージとのデータ転送
HPCI 用フロントエンドサーバ	shpcif.cc.tohoku.ac.jp	HPCI、HPCI-JHPCN 課題用フロントエンドサーバ

3.2 SSH 認証鍵ペアの作成とログインまでの概要

図 3 に、鍵ペアの作成からログインまでの流れを示します。

1. 鍵ペアの作成は 3.3 節で
2. ターミナルソフトの設定および 3. ログイン については 3.4 節でそれぞれ解説します。

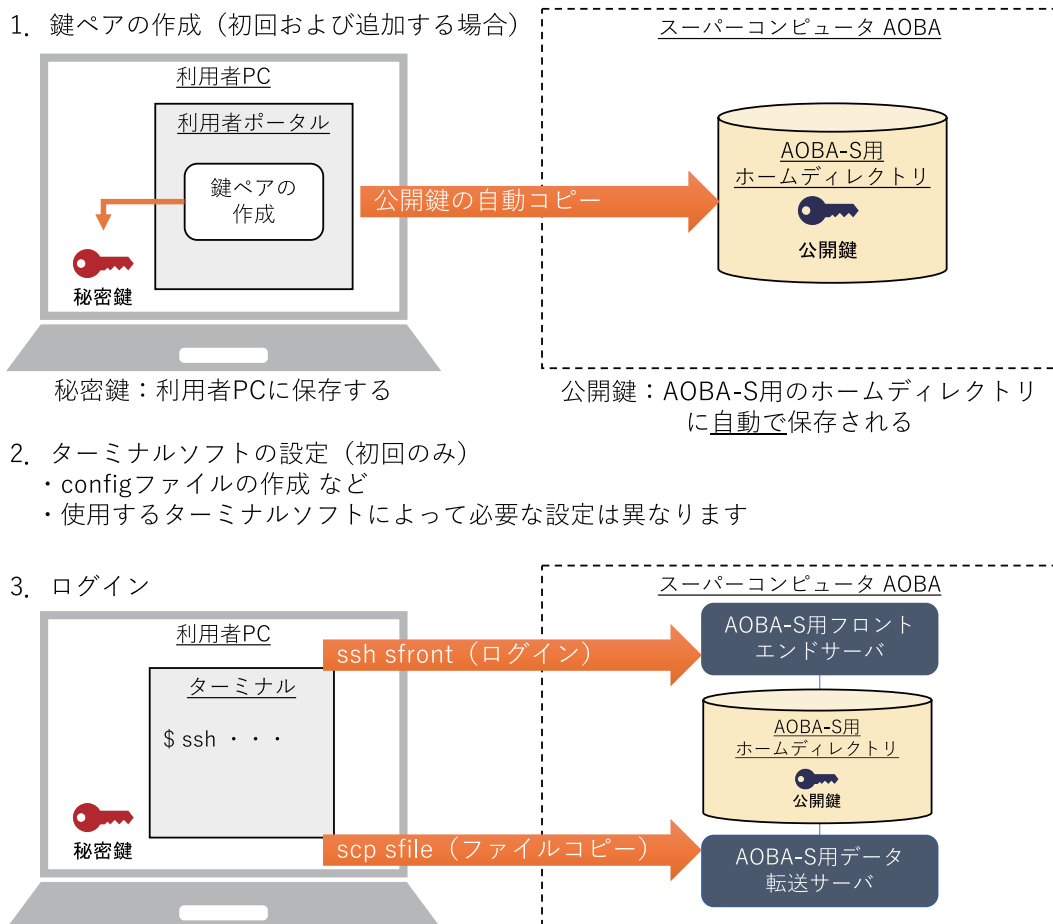


図 3 鍵ペアの作成からログインまで

1. 鍵ペアの作成 (初回ログイン時、およびログイン端末を追加する場合)
利用者ポータルで鍵ペアを作成します。作成された秘密鍵は利用者のローカル PC に保存します。公開鍵は、スーパーコンピュータ AOBA のホームディレクトリ上に自動で保存されます。
【利用者ポータル】 <https://www.ss.cc.tohoku.ac.jp/portal/>
2. ターミナルソフトの設定 (初回ログイン時)
各利用者用サーバにログインするための設定を行います。使用するターミナルソフトによって必要な設定が異なります。
3. ログイン
利用者のローカル PC に保存した秘密鍵を使って利用者用サーバにログインします。

3.3 公開鍵認証方式で使用する鍵ペアの作成

3.3.1 公開鍵認証方式を使用する上での注意事項

以下の注意事項を必ず守ってください。

守られない場合、不正アクセス（不正ログイン、クライアントのなりすまし、暗号化された通信の暴露、他サーバへの攻撃など）のリスクが非常に高まるので**大変危険**です。

- パスフレーズなしの秘密鍵を使用しないこと
- 秘密鍵、パスフレーズを使いまわさないこと
- 秘密鍵を持ち出さないこと（メールに添付しない、USB メモリ等に保存しない）
- 秘密鍵をスーパーコンピュータ AOBA のホームディレクトリに保存しないこと
- 公開鍵と秘密鍵の鍵ペアを同一ホスト上に保存しないこと

3.3.2 鍵ペアの作成（初回ログイン時、および接続する PC を追加する場合）

■初回ログイン時 鍵ペアの作成は利用者ポータルで行います。

- (1) 以下の URL から利用者ポータルに接続します。利用者ポータルには利用者番号と LDAP パスワード（※）でログインします。

【利用者ポータル】 <https://www.ss.cc.tohoku.ac.jp/portal/>

- (2) 「SSH 公開鍵登録」 ボタンをクリックします。
- (3) 利用者ポータルの画面の説明に従い、鍵ペアを作成します。
(パスフレーズを設定し、鍵生成・登録ボタンをクリック)
- (4) 作成された秘密鍵を利用者のローカル PC に保存します。保存先は以下を推奨します。フォルダがない場合は新規作成します。
 - (Windows の場合) C:¥Users¥ (ユーザ名) ¥.ssh
 - (macOS / Linux の場合) \$HOME/.ssh

ポータルサイトで作成された公開鍵は、AOBA-S と AOBA-A,B のホームディレクトリの公開鍵ファイル (\$HOME/.ssh/authorized_keys) に自動で保存されます。

※利用者ポータルで使用する LDAP パスワードの変更方法は、3.6 節を参照してください。

■別の PC からログインする場合 接続する PC からポータルサイトにアクセスし、新しい鍵ペアを作成します。既存の秘密鍵を使いまわすのではなく、端末ごとに鍵ペアを作成してください。

3.4 利用者向けサーバへのログイン方法

3.4.1 ターミナルソフトの設定（各 PC での設定）

利用者の PC 上で、ターミナルソフトの設定を行います。以降の解説では、次のフォルダを「.ssh フォルダ」と呼び、秘密鍵を「id_rsa.cc」というファイル名で.ssh フォルダに保存した場合とします。

- (Windows の場合) C:¥Users¥ (ユーザ名) ¥.ssh
- (macOS / Linux の場合) \$HOME/.ssh

各利用者向けサーバのホスト名は、次の文字列で設定するものとして解説します。ホスト名には任意の文字列を設定することができますが、他の文字列を設定した場合は、以降の解説におけるホスト名を読み替えてください。

- (AOBA-S 用フロントエンドサーバ) sfront
- (AOBA-S 用データ転送サーバ) sfile
- (AOBA-S HPCI 用フロントエンドサーバ) shpcif

(1) macOS / Linux の場合は、秘密鍵のパーミッションを 600 に変更が必要です。ターミナルソフトを起動し以下のコマンドを実行します (リスト 1)。

リスト 1 秘密鍵のパーミッション変更

```
(localhost)$ chmod 600 ${HOME}/.ssh/id_rsa_cc
```

以降は Windows、macOS / Linux 共通です。

(2) .ssh フォルダの「config」というファイルをテキストエディタで開きます。ファイルがない場合は新規作成します。拡張子は付けません。(フォルダの設定を「拡張子を表示しない」にしている場合、意識せずに拡張子付きのファイルを作成している可能性があります。config ファイルに拡張子がついていると正しく設定が読み込まれません。)

(3) config ファイルにリスト 2 に示した設定を記述します。ホスト名を例とは別に設定した場合は、以降のコマンドではご自身の環境に合わせて読み替えてください。

リスト 2 config ファイルの設定方法

```
# AOBA-S 用フロントエンドサーバに接続するための設定
Host sfront # AOBA-S 用フロントエンドサーバ名を sfront で指定
HostName sfront.cc.tohoku.ac.jp # ホスト名を FQDN で指定
User 利用者番号 # 利用者番号を指定
IdentityFile $HOME/.ssh/id_rsa_cc # 秘密鍵の保存場所とファイル名を指定

# AOBA-S 用データ転送サーバに接続するための設定
Host sfile # AOBA-S 用データ転送サーバ名を sfile で指定
HostName sfile.cc.tohoku.ac.jp # ホスト名を FQDN で指定
User 利用者番号 # 利用者番号を指定
IdentityFile $HOME/.ssh/id_rsa_cc # 秘密鍵の保存場所とファイル名を指定

# HPCI 用フロントエンドサーバに接続するための設定
Host shpcif # HPCI 用フロントエンドサーバ名を shpcif で指定
HostName shpcif.cc.tohoku.ac.jp # ホスト名を FQDN で指定
User 利用者番号 # 利用者番号を指定
IdentityFile $HOME/.ssh/id_rsa_cc # 秘密鍵の保存場所とファイル名を指定
```

3.4.2 AOBA-S 用フロントエンドサーバへのログイン

ターミナルソフトを起動しリスト 3 に示したコマンドを実行すると config ファイルに記述した設定が読み込まれ、AOBA-S 用のフロントエンドサーバにログインします。

リスト 3 AOBA-S 用フロントエンドサーバへのログイン

```
(localhost)$ ssh sfront
Last login: Tue Aug 1 12:34:56 2023 from x.x.x.x
(sfront)$ # 接続するとプロンプトのホスト名が変わります
```

フロントエンドサーバは冗長構成になっており、自動的に sfront1 または sfront2 が選択されます。どちらにログインしても環境は変わりません。

なお、フロントエンドサーバでは一定時間以上のプロセスは実行できません。また、大容量のデータ転送はシステムに高い負荷がかかります。大容量のデータ転送を行う場合はデータ転送サーバをご利用ください。

3.4.3 AOBA-S 用データ転送サーバの利用方法

AOBA-S 用のデータ転送サーバは、利用者のローカル PC と AOBA-S 用のストレージ間のデータ転送、および AOBA-S 用ストレージと AOBA-A,B 用ストレージ間のデータコピーに用います。

ローカル PC とのデータ転送は、scp コマンドや sftp コマンド、ファイル転送のアプリケーションを用いて行います。リスト 4 に scp コマンドを使用したデータ転送の例を示します。ターミナルソフトを起動し、scp コマンドでローカル PC と AOBA-S 用ストレージ間のデータ転送を行います。

リスト 4 AOBA-S 用データ転送サーバ

```
# ローカルPCのデータをAOBA-S用ストレージに転送する場合
(localhost)$ scp -r ローカルPCのデータ sfile:/uhome/利用者番号/コピー先

# AOBA-S用ストレージのデータをローカルPCに転送する場合
(localhost)$ scp -r sfile:/uhome/利用者番号/データ ローカルPCのコピー先
```

また AOBA-S 用ストレージと AOBA-A,B 用ストレージ間のデータコピーは、AOBA-S 用のデータ転送サーバにログインした後に cp コマンドで行います。リスト 5 に AOBA-S 用のデータ転送サーバへのログイン方法、およびリスト 6 に cp コマンドを使用したデータコピーの例を示します。

ターミナルソフトを起動しリスト 5 示したコマンドを実行すると config ファイルに記述した設定が読み込まれ、AOBA-S 用のデータ転送サーバにログインします。

リスト 5 AOBA-S 用データ転送サーバ

```
(localhost)$ ssh sfile
Last login: Tue Aug 1 12:34:56 2023 from x.x.x.x
(sfile)$ # 接続するとプロンプトのホスト名が変わります
```

AOBA-S 用データ転送サーバ上でのマウントポイントは以下の通りです。

- (AOBA-S 用ストレージ) /uhome/利用者番号/
- (AOBA-A,B 用ストレージ) /mnt/stfs/uhome/利用者番号/

sfile にログインした後リスト 6 に示したコマンドで、AOBA-S 用ストレージと AOBA-A,B 用ストレージ間のデータコピーを行います。

リスト 6 AOBA-S 用データ転送サーバ

```
# AOBA-A,B用ストレージのデータをAOBA-S用ストレージにコピーする場合
(sfile)$ cp /mnt/stfs/uhome/利用者番号/AOBA-A,Bのデータ /uhome/利用者番号/AOBA-Sのコピー先

# AOBA-S用ストレージのデータをAOBA-A,Bストレージにコピーする場合
(sfile)$ cp /uhome/利用者番号/AOBA-Sのデータ /mnt/stfs/uhome/利用者番号/AOBA-A,Bのコピー先
```

なお、AOBA-S 用フロントエンドサーバと HPCI 用フロントエンドサーバからは、AOBA-A,B 用ストレージ用のマウントポイントにはアクセスできません。

3.4.4 HPCI 用フロントエンドサーバへのログイン

ターミナルソフトを起動しリスト 7 に示したコマンドを実行すると config ファイルに記述した設定が読み込まれ、HPCI 用のフロントエンドサーバにログインします。HPCI および HPCI-JHPCN 課題利用者のみログイン可能です。

リスト 7 HPCI 用フロントエンドサーバ

```
(localhost)$ ssh shpcif
Last login: Tue Aug 1 12:34:56 2023 from x.x.x.x
(shpcif)$ # 接続するとプロンプトのホスト名が変わります
```

3.5 ログインシェルの確認と変更

ログインシェルのデフォルトは `bash` が設定されています。設定の確認および変更はリスト 8 に示した手順で行います。ログインシェルの変更がシステム全体に反映されるまで、15 分程度かかります。

ログインシェルの変更は、AOBA システム（AOBA-S 用、AOBA-A,B 用の全てのホスト、プリンタサーバ）の利用者向けサーバに対して設定されます。

リスト 8 ログインシェルの確認と変更

<code>(sfront)\$ fchsh</code>	(現在のログインシェルの確認)
<code>Enter Password:</code>	(LDAPパスワードを入力)
<code>loginShell: /bin/bash</code>	(現在のログインシェルが表示される)
<code>(sfront)\$ fchsh /bin/tcsh</code>	(ログインシェルを/bin/tcshに変更)
<code>Enter Password:</code>	(LDAPパスワードを入力)
<code>Changed loginShell to /bin/tcsh</code>	(ログインシェルが変更された)

3.6 LDAP パスワードの変更

利用者ポータルやログインシェルの変更などで使用する LDAP パスワードの変更は、利用者ポータルで行います。

- (1) 以下の URL から利用者ポータルにログインします。利用者ポータルには、利用者番号と現在の LDAP パスワードでログインします。

【利用者ポータル】 <https://www.ss.cc.tohoku.ac.jp/portal/>

- (2) 「パスワード変更」ボタンをクリックします。
- (3) 利用者ポータルの画面の説明に従い、新しい LDAP パスワードを設定します。
- (4) 以下で使用するパスワードが変更されます。
 - 利用者ポータルへのログイン
 - 大判カラープリンタのプリンタサーバへのログイン
 - ログインシェルの変更時

4 プログラムのコンパイルと実行

本章ではプログラムのコンパイルと、サブシステム AOBA-S で実行する手順を説明します。

4.1 AOBA-S 用フロントエンドサーバへのログイン

サブシステム AOBA-S 用プログラムのコンパイルとジョブ投入は AOBA-S 用フロントエンドサーバで行います。AOBA-S 用フロントエンドサーバへの詳しい接続方法は 3.4 節をご参照ください。設定が完了している場合はリスト 9 に示したコマンドで、AOBA-S 用フロントエンドサーバにログインします。

なお AOBA-S 用フロントエンドサーバから、AOBA-A および AOBA-B へのジョブ投入などの操作はできません。

リスト 9 AOBA-S 用フロントエンドサーバへのログイン

```
(localhost)$ ssh sfront
Last login: Tue Aug 1 12:34:56 2023 from x.x.x.x
(sfront)$ # 接続するとプロンプトのホスト名が変わります
```

4.2 ソースコードの作成

ソースコードを作成する代表的な方法は以下の通りです。

- フロントエンドサーバ上でテキストエディタで作成する
- ローカル PC 上のテキストエディタで作成し、AOBA-S 用ストレージにファイル転送する
- Visual Studio Code などの開発環境で作成する

フロントエンドサーバのコンソール上でソースファイルを作成するためのテキストエディタは、emacs や vim、nano が利用できます。

ローカル PC で作成したソースコードファイルを転送する場合は、ファイル転送ソフトや scp コマンドで利用者のホームディレクトリに転送してください。その際、ソース（テキスト）ファイルは ASCII モードで転送します。

Windows 環境で作成したソースコードの改行コードと文字コードを Linux 用に変換するためには、AOBA-S 用ストレージに転送したソースコードに nkf コマンドを利用します（リスト 10）。

リスト 10 nkf コマンドの使用例

```
# 改行コードを LF、文字コードを UTF-8 に変換し、別のファイルに書き出す方法
(sfront)$ nkf -Lu ソースコードファイル名 > 変換後ファイル名

# 改行コードを LF、文字コードを UTF-8 に変換して上書き保存する方法
(sfront)$ nkf --overwrite -Lu ソースコードファイル名
```

開発環境の利用方法については各アプリケーションのマニュアルをご参照ください。

4.3 コンパイル

4.3.1 コンパイラの仕様

AOBA-S では SX-Aurora TSUBASA 用に最適化された Fortran コンパイラ、および C/C++ コンパイラを利用できます。コンパイラの仕様は以下の通りです。

Fortran コンパイラ

- nfort、mpinfort コマンドでコンパイルとリンク
- 自動ベクトル化・自動並列化機能を実装
- ISO/IEC 1539-1:2010 Programming languages - Fortran に準拠
- OpenMP Application Program Interface Version 4.5 に準拠
- ISO/IEC 1539-1:2018 Programming languages - Fortran の一部機能にも対応
- OpenMP Application Program Interface Version 5.0 の一部機能にも対応

C/C++ コンパイラ

- ncc、mpincc、nc++、mpinc++ コマンドでコンパイルとリンク
- 自動ベクトル化・自動並列化機能を実装
- ISO/IEC 9899:2011 Programming languages - C に準拠
- ISO/IEC 14882:2014 Programming languages - C++ に準拠
- ISO/IEC 14882:2017 Programming languages - C++ に準拠
- OpenMP Application Program Interface Version 4.5 に準拠
- ISO/IEC 14882:2020 Programming languages - C++ の一部機能にも対応
- OpenMP Application Program Interface Version 5.0 の一部機能にも対応

4.3.2 コンパイルの手順

ソースコードファイルはそれぞれの言語用コマンドでコンパイルを行います。以下では単一コアで実行する逐次実行、自動並列化または OpenMP 並列化による共有メモリ並列実行、および MPI ライブラリによる分散メモリ並列実行のコンパイル手順について解説します。

■Fortran プログラムのコンパイル ソースコードファイルは nfort コマンドまたは mpinfort コマンドでコンパイルを行います。必要に応じて、表 3 に示したコンパイルオプションをコンパイル時に指定します。その他のコンパイルオプションについては、6 章のコンパイラユーザーズガイドをご参照ください。

ソースファイルの拡張子は、自由形式（フリーフォーマット）なら.f90 .f95 か.F90 .F95 を、固定形式（7カラム目から記述）なら.f か.F を、2003 形式であれば.f03 か.F03 を付けます。（拡張子が大文字で始まるものは、コンパイルの前に fpp によるプリプロセス処理が行われます。）

コンパイルが成功すると出力ファイル名を指定しない場合は、カレントディレクトリに実行モジュールの a.out が作成されます。

表3 Fortran,C/C++ コンパイラの主なオプション

コンパイルオプション	機能
-On n に指定できる値	最適化レベルを指定する
4	言語仕様を逸脱した副作用を伴う最大限の最適化・自動ベクトル化を適用する
3	副作用を伴う最適化・自動ベクトル化、および、多重ループの最適化を適用する
2	(既定値) 副作用を伴う最適化・自動ベクトル化を適用する
1	副作用を伴わない最適化・自動ベクトル化を適用する
0	最適化、自動ベクトル化、並列化、インライン展開を適用しない (最適化レベルを高くすると、最適化の副作用により計算結果が変わることがありますのでご注意ください)
-finline-functions	自動インライン展開機能を適用する
-mparallel	自動並列化機能を利用する
-fopenmp	OpenMP 並列化を利用する
-mno-parallel-omp-routine	OpenMP ディレクティブを含むルーチンを自動並列化しない (-mparallel と -fopenmp が同時に指定されたとき、ループが OpenMP の並列区間の外側にある場合は外側のループが自動並列化の対象となります)
-ftrace	性能解析 ftrace 機能用の実行ファイルを作成する
-report-diagnostics	ベクトル診断リストを出力する
-fdiag-vector=2	詳細なベクトル化診断メッセージを出力する (既定値は 1)
-report-format	ベクトル化、並列化などの最適化情報がソース行とともに出力された編集リストを出力する
-report-all	コード生成リスト、診断メッセージリスト、編集リスト、インラインリスト、オプションリスト、ベクトルリストを出力する
-fcheck=bounds	バウンズチェック (配列の上下限のチェック) を行う
-fcheck=all	仮引数のエリアスへの代入、ビット intrinsic な引数、配列の上下限、不正なポインタ、DO ループのステップ値が 0 かどうか、整数オーバーフロー、ポインタ参照、省略可能な引数の参照、不正な再帰呼出しのチェックを行う

リスト 11 nfort コマンドの使用例

(逐次実行)
(sfront)\$ nfort コンパイルオプション Fortranソースファイル名
(自動並列化実行)
(sfront)\$ nfort -mparallel コンパイルオプション Fortranソースファイル名
(OpenMP並列実行)
(sfront)\$ nfort -fopenmp コンパイルオプション Fortranソースファイル名

リスト 12 mpinfort コマンドの使用例

(MPI並列実行)
(sfront)\$ mpinfort コンパイルオプション Fortranソースファイル名
(MPIと自動並列化の同時並列実行)
(sfront)\$ mpinfort -mparallel コンパイルオプション Fortranソースファイル名
(MPIとOpenMPの同時並列実行)
(sfront)\$ mpinfort -fopenmp コンパイルオプション Fortranソースファイル名

■**C/C++ プログラム** ソースコードは `ncc` または `mpincc` コマンドで C プログラムを、`nc++` または `mpinc++` コマンドで C++ プログラムをコンパイルします。必要に応じて Fortran コンパイラと同じく、表 3 に示したコンパイルオプションをコンパイル時に指定します。その他のコンパイルオプションについては、6 章のコンパイラユーザーズガイドをご参照ください。

ソースファイルの拡張子は、C 言語であれば `.c` を、C++ であれば `.C` `.cc` `.cpp` `.cp` `.cxx` `.c++` のいずれかを付けます。

コンパイルが成功すると出力ファイル名を指定しない場合は、カレントディレクトリに実行モジュールの `a.out` が作成されます。

リスト 13 `ncc/nc++` コマンドの使用例

```
(逐次実行)
(sf) $ ncc コンパイルオプション Cソースファイル名
(sf) $ nc++ コンパイルオプション C++ソースファイル名

(自動並列化実行)
(sf) $ ncc -mparallel コンパイルオプション Cソースファイル名
(sf) $ nc++ -mparallel コンパイルオプション C++ソースファイル名

(OpenMP並列実行)
(sf) $ ncc -fopenmp コンパイルオプション Cソースファイル名
(sf) $ nc++ -fopenmp コンパイルオプション C++ソースファイル名
```

リスト 14 `mpincc/mpinc++` コマンドの使用例

```
(MPI並列実行)
(sf) $ mpincc コンパイルオプション Cソースファイル名
(sf) $ mpinc++ コンパイルオプション C++ソースファイル名

(MPIと自動並列化の同時並列実行)
(sf) $ mpincc -mparallel コンパイルオプション Cソースファイル名
(sf) $ mpinc++ -mparallel コンパイルオプション C++ソースファイル名

(MPIとOpenMPの同時並列実行)
(sf) $ mpincc -fopenmp コンパイルオプション Cソースファイル名
(sf) $ mpinc++ -fopenmp コンパイルオプション C++ソースファイル名
```

■**実行時性能解析情報 (FTRACE) の出力** コンパイル時に `-ftrace` オプションを指定すると、実行後にディレクトリに性能解析情報の結果ファイル (`ftrace.out`) が書き出されます。MPI 並列実行時の性能情報も取得可能です。

性能解析情報の確認は、フロントエンドサーバ上で `ftrace` コマンドを実行してテキスト形式で確認するか、`ftraceviewer` コマンドで GUI で確認することができます。図 4 は Ftrace Viewer の表示例です。

FTRACE と Ftrace Viewer についての詳細は、6 章の PROGINF/FTRACE ユーザーズガイド、および NEC Ftrace Viewer ユーザーズガイドをご参照ください。



図4 Ftrace Viewer の表示例

4.4 バッチリクエストによるジョブの実行

フロントエンドサーバでコンパイル作業を行って作成したプログラムの実行は、バッチ処理と呼ばれる方法で計算機に実行を依頼します。本センターではバッチ処理に NEC Network Queuing System V (以下、NQSV) を採用しています。

NQSV についてのコマンドやオプションについての詳細は、「NQSV 利用の手引 操作編」をご参照ください。なお、当センター独自の運用方法のためマニュアルの記載の通りに動作しない場合もあります。

4.4.1 バッチリクエストの概要

バッチリクエストは、バッチ処理で計算機にジョブの実行を依頼するリクエストのことです。ジョブとは、実行可能ファイル、プログラム、実行モジュールまたはコマンドのことで、リクエストは1つまたは複数のジョブから構成されます。

図5にバッチリクエスト実行の概念図を示します。

1. ジョブスクリプトの作成は4.4.3 から4.4.5 節で
2. バッチリクエストの投入は4.4.6 節で
3. 実行待ち→実行は4.4.7 から4.4.9 節で
4. 実行終了は4.4.10 でそれぞれ解説します。

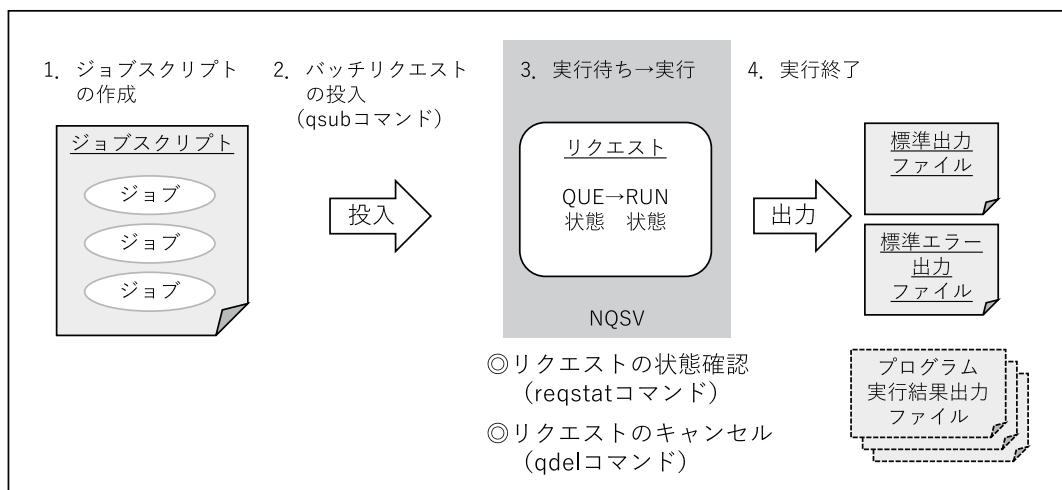


図5 バッチリクエストの概念図

1. ジョブスクリプトの作成

プログラムの実行手続きを書いたテキストファイルを作成します。利用形態、利用 VE 数、最大経過時間も記述します。ノードの空き状況によっては最大経過時間をデフォルトの時間よりも短く指定することで、バックフィルスケージュERINGにより実行開始予定時刻が早くなる場合があります。

2. バッチリクエストの投入

3. 実行待ち→実行

以下のコマンドでバッチリクエストを操作します。

- qsub コマンド NQSV にバッチリクエストの投入
- reqstat コマンド 投入したリクエストの状態を表示（状態確認）
- qdel コマンド 投入したリクエストのキャンセル

4. 実行終了

リクエストの実行が終了すると、標準出力と標準エラー出力がファイルとして書き出されます。標準出力/標準エラー出力のファイルサイズ上限は 1GB です。ファイルサイズが上限値を超えた場合はプログラムが強制終了しますので、実行結果はファイル名を指定して書き出しを行ってください。

終了したリクエストは reqstat コマンドの表示から削除されます。

4.4.2 キュー構成

リクエストの投入キューについて説明します。

表 4 に AOBA-S のキュー構成を示します。ジョブスクリプトの冒頭で、表に示したキュー名、利用 VE 数、最大経過時間を指定します。

表 4 サブシステム AOBA-S のキュー構成

利用形態	キュー名	VE 数	最大経過時間 規定値/最大値	メモリサイズ
無料	sxsf	1	1 時間/1 時間	96GB
共有	sxs	1~2,048	72 時間/720 時間	96GB × VE 数
占有		個別設定		96GB × VE 数

経過時間は、バッチリクエストが開始してから終了するまでの時間です。指定した最大経過時間を超えた場合、プログラムの実行中でもバッチリクエストは強制的に終了します。I/O が高負荷となる場合、経過時間が想定よりも遅くなる場合がありますので、必要十分な最大経過時間を指定してください。

なお、最大経過時間は最大値 720 時間（720:00:00）を超えて指定することは出来ません。（sxsf では経過時間の最大値は 1:00:00 です。）

4.4.3 バッチリクエストの作成

バッチリクエスト用のシェルスクリプトファイル（ジョブスクリプト）を作成します。通常のシェルスクリプトと同様に、テキストファイル形式で任意のコマンドを組み合わせるプログラムの実行手続きを記述します。ジョブスクリプトを実行するシェルは、sh、csh ともに使用できますが、解説では sh スクリプト形式で記述し、ファイル名を run.sh とします。

ジョブスクリプトに記述する基本項目は以下の通りです。その他に、環境変数の指定やファイル操作コマンドが必要な場合は、適切な箇所に記述します。

- 投入キュー名
- 利用する VE 数
- 最大経過時間
- 作業ディレクトリへの移動コマンド
- プログラムの実行コマンド

4.4.4 ジョブスクリプトの例

リスト 15～リスト 20 に AOBA-S 向けのジョブのスクリプトファイル例を示します。

■**逐次実行の場合** リスト 15 およびリスト 16 に、逐次実行の場合のジョブスクリプトの例を示します。逐次実行の場合、プログラムは 1VE 内の 1 コアで実行されます。

リスト 15 ジョブスクリプトの例 (逐次実行)

```
#!/bin/sh
#PBS -q sxs                # AOBA-Sを使用する
#PBS --venode 1           # VEを1個使う
#PBS -l elapstim_req=2:00:00 # 最大経過時間を2時間に指定

cd $PBS_0_WORKDIR        # qsubを実行したディレクトリに移動
./a.out                  # カレントディレクトリの a.out を実行
```

リスト 16 ジョブスクリプトの例 (逐次実行、無料キュー)

```
#!/bin/sh
#PBS -q sxs                # AOBA-Sの無料キューを使用する
#PBS --venode 1           # VEを1個使う
#PBS -l elapstim_req=0:30:00 # 最大経過時間を30分に指定

cd $PBS_0_WORKDIR        # qsubを実行したディレクトリに移動
./a.out                  # カレントディレクトリの a.out を実行
```

■**自動並列化/OpenMP 並列実行の場合** リスト 17 に自動並列化/OpenMP 並列実行の場合のジョブスクリプトの例を示します。自動並列化/OpenMP 並列実行の場合、プログラムは 1VE 中の 2～16 コアで実行されます。実行コア数の指定は環境変数 VE_OMP_NUM_THREADS で行います。逐次実行と同様に、sxsf も指定できます。

リスト 17 ジョブスクリプトの例 (自動並列化/OpenMP 並列実行)

```
#!/bin/sh
#PBS -q sxs                # AOBA-Sを使用する
#PBS --venode 1           # VEを1個使う
#PBS -l elapstim_req=2:00:00 # 最大経過時間を2時間に指定

export VE_OMP_NUM_THREADS=16 # VE内は16コア並列で実行

cd $PBS_0_WORKDIR        # qsubを実行したディレクトリに移動
./a.out                  # カレントディレクトリの a.out を実行
```

■**MPI 並列実行の場合** リスト 18 に MPI 並列実行の場合のジョブスクリプトの例を示します。MPI 並列実行の場合は、mpirun コマンドでプログラムの実行を行います。

MPI プロセス数が、確保した VE の物理コア数 (VE 数× 16) を超えると演算性能が著しく低下しますのでご注意ください。

逐次実行や自動並列化/OpenMP 並列実行用でコンパイルされたプログラムは、mpirun コマンドで実行を行っても複数 VE での実行はされません。

リスト 18 ジョブスクリプトの例 (MPI 並列実行)

```
#!/bin/sh
#PBS -q sxs          # AOBA-Sを使用する
#PBS --venode 8     # VEを8個使う
#PBS -l elapstim_req=2:00:00 # 最大経過時間を2時間に指定

cd $PBS_0_WORKDIR   # qsubを実行したディレクトリに移動
mpirun -np 128 ./a.out # カレントディレクトリの a.out を128プロセス並列で実行
```

■MPI と自動並列化/OpenMP 同時並列実行の場合 リスト 19 に MPI と自動並列化/OpenMP の同時並列実行の場合のジョブスクリプトの例を示します。

(MPI プロセス数) × (VE 内並列数) が、確保した VE の物理コア数 (VE 数 × 16) を超えると演算性能が著しく低下しますのでご注意ください。

リスト 19 ジョブスクリプトの例 (MPI と自動並列化/OpenMP 同時並列実行)

```
#!/bin/sh
#PBS -q sxs          # AOBA-Sを使用する
#PBS --venode 8     # VEを8個使う
#PBS -l elapstim_req=2:00:00 # 最大経過時間を2時間に指定

export VE_OMP_NUM_THREADS=16 # VE内は16コア並列で実行

cd $PBS_0_WORKDIR   # qsubを実行したディレクトリに移動
mpirun -np 8 ./a.out # カレントディレクトリの a.out を8プロセス×16コア並列で実行
```

■標準出力、標準エラーファイルをプロセス毎に分割出力する 標準出力および標準エラー出力は、1つのファイルに各プロセスからの出力が混在して書き出されます。このとき、システムで用意されたシェルスクリプト mpisep.sh を利用すると、同一ファイルにプロセス毎の出力が入り混じって出力されないように、MPI プロセスごとにファイルを分けて出力することができます。リスト 20 に示したように、シェルスクリプト/opt/nec/ve/bin/mpisep.sh を実行形式ファイル a.out の前に記述し、環境変数 NMPI_SEPSELECT に 1 から 4 の値を指定することで、表 5 に示した動作を選択できます。

リスト 20 出力をプロセス毎に分割する方法

```
#!/bin/sh
#PBS -q sxs          # AOBA-Sを使用する
#PBS --venode 8     # VEを8個使う
#PBS -l elapstim_req=2:00:00 # 最大経過時間を2時間に指定
#PBS -v NMPI_SEPSELECT=3 # 出力形式3を指定

cd $PBS_0_WORKDIR   # qsubを実行したディレクトリに移動
mpirun -np 128 /opt/nec/ve/bin/mpisep.sh ./a.out # カレントディレクトリの a.out を128プロセス並列で実行
```

表 5 NMPI_SEPSELECT の引数と出力形式

NMPI_SEPSELECT の引数	出力形式
1	MPI プロセスの標準出力をプロセス別のファイルに保存
2	(既定値) MPI プロセスの標準エラー出力を プロセス別のファイルに保存
3	MPI プロセスの標準出力および標準エラー出力を それぞれプロセス別のファイルに保存
4	MPI プロセスの標準出力および標準エラー出力を プロセス別の 1 つのファイルに保存

実行時に stdout.*や stderr.*の同名ファイルが存在する場合は、上書きではなくファイルに追記します。

4.4.5 qsub コマンドの主なオプション

表 6 に、qsub コマンドの主なオプションと機能を示します。ジョブスクリプトの冒頭から、#PBS の後に各オプションを記述します。

表 6 qsub コマンドの主なオプションと機能

オプション	機能	指定
-q 投入キュー名	投入キュー名を指定	必須
-venode 利用 VE 数	AOBA-S で利用する VE 数を指定 (このオプションは先頭のハイフンが 2 つ)	必須
-l elapstim_req=hh:mm:ss	最大経過時間を指定	必須 (注 1 参照)
-A 課金先番号 (PJ)	課金先の番号を指定	任意 (注 2 参照) 省略時: デフォルトの PJ
-N リクエスト名	リクエスト名を指定	任意 省略時: ジョブスクリプトファイル名
-o ファイル名	標準出力のファイル名を指定	任意 省略時: リクエスト名.o リクエスト ID
-e ファイル名	標準エラー出力のファイル名を指定	任意 省略時: リクエスト名.e リクエスト ID
-jo	標準出力および標準エラー出力を 同一ファイルに出力	任意
-m b	リクエスト実行開始時にメールを送信	任意
-m e	リクエスト実行終了時にメールを送信	任意
-m be	リクエスト実行開始時と終了時にメールを送信	任意
-M メールアドレス	メールの送信先を指定	任意
-r y または n	リクエストのリラン可否を指定 (注 3 参照)	任意 省略時: y

(注 1) 経過時間はバッチリクエストが実行を開始してから、終了するまでの時間です。プログラムの実行に、最大でどれくらいの経過時間の確保が必要かを指定します。指定した最大経過時間が短いリクエストほど計算資源が確保されやすく、実行待ちの時間を短縮することができます。

ただし、指定した最大経過時間を超えると、実行は打ちきりとなり強制終了します。I/O が高負荷となる場合、経過時間が想定よりも長くなることがありますので、必要十分な時間を指定してください。

(注 2) デフォルトの課金先番号や、利用可能な課金先番号の確認は、フロントエンドサーバ上で project コマンドを実行します。デフォルトの課金先番号の変更も可能です。

(注 3) リランとは、リクエストを始めから実行し直すことで、計算機のメンテナンスや障害発生時に管理者側でリクエストをリランする場合があります。

リランによりプログラムの実行に不都合が生じる場合 (実時間で時間計測を行っている場合など) は、-r n (リランしない) を指定してください。

4.4.6 バッチリクエストの投入

バッチリクエストの投入は qsub コマンドで行います。リスト 21 に qsub コマンドの実行例を示します。ジョブスクリプトファイル名が run.sh の場合です。

リスト 21 qsub コマンドによるバッチリクエスト投入

```
(sfront)$ qsub run.sh
```

バッチリクエストが正常に投入されるとリスト 22 のようなメッセージが表示されます。この例では、リクエスト ID には 1234.sjob が割り当てられ、投入先は sxs キュー、課金先番号は un0000 が指定されたことを示しています。リクエスト ID はバッチリクエストの状況確認やキャンセルの際に用います。

バッチリクエストの投入が失敗した場合は、エラーメッセージが表示されますので、ジョブスクリプトの記述などに誤りがないかを確認してください。

リスト 22 qsub コマンドによるバッチリクエスト投入

```
(sfront)$ qsub run.sh
プロジェクトコード:un0000 にリクエストを投入します
Request 1234.sjob submitted to queue : sxs.
```

4.4.7 バッチリクエストの実行

投入されたリクエストは、実行待ちの列に並びます。リクエストの実行順序はバックフィルスケージュERINGで制御されます。

バックフィルスケージュERINGでは、計算資源が確保できたリクエストから実行を開始します。実行に必要な計算資源量（利用 VE 数×最大経過時間）が少ないリクエストは、投入順序によらず早く実行開始する場合があります。

4.4.8 バッチリクエストの状態確認

バッチリクエストの状態確認は reqstat コマンドで行います。reqstat コマンドを実行すると、実行待ちおよび実行中のリクエストの状態が表示されます。該当するリクエストがない場合は何も表示されません。

表 7 に reqstat コマンドの表示項目を、表 8 にリクエストの状態の説明を示します。

表 7 reqstat コマンドの表示項目

表示項目	内容
RequestID	リクエスト ID
RequestName	リクエスト名
User	利用者番号
PJCode	課金先番号（プロジェクトコード）
Que	実行キュー名
Node	実行時に指定した VE 数
ElapseLimit	投入時に指定した最大経過時間
STT	リクエストの状態
StartTime	実行開始予定時刻 (実行開始後は実際の実行開始時刻)
Memory1	現在の VH 使用メモリ量
Memory2	現在の VE 使用メモリ量
ElapseTime	現在までの経過時間
NodeTime	現在までのノード時間 (ElapseTime × Node) (課金対象時間)

表 8 reqstat コマンドの表示項目

表記	リクエストの状況
RUN	実行中
QUE	実行待ち
EXT	標準出力/標準エラー出力ファイルの出力中 (※)

リクエストの実行開始予定時刻は、StartTime の欄に表示されます。実行開始予定時刻は他のリクエストの状況により随時更新されます。通常は更新前の時刻よりも遅くなることはありませんが、システムの障害発生時にはその限りではありません。

※ 以下のような場合、リクエストが EXT 状態で長く停滞することがあります。他の利用者のリクエストの実行に影響が出ますのでご注意ください。

- ファイル容量がクォータ値を超えたため、標準出力/標準エラー出力ファイルを出力できない
ファイルを整理して容量を空けてください。空き容量が確保されるまで EXT 状態で停滞したままになります。容量が確保され次第、ファイルが出力されリクエストが終了します。
- システム上で問題が発生している
管理者で対応しますのでそのままお待ちください。状況をメールでご連絡いたします。

ファイル容量の追加方法は以下のページをご参照ください。

【データ転送 (ストレージ)】 <https://www.ss.cc.tohoku.ac.jp/storage/>

4.4.9 バッチリクエストのキャンセル

投入したリクエストをキャンセルする場合は、qdel コマンドを使用します。qdel コマンドを実行すると実行中のプログラムは強制終了し、リクエストは削除されます。リクエストのキャンセルは投入した利用者番号から行うことができ、他利用者のリクエストはキャンセルできません。

リスト 23 に qdel コマンドの実行例を示します。1234.sjob というジョブ ID のリクエストをキャンセルする例で、qdel に続けてキャンセルするリクエスト ID を指定します。リクエスト投入時、または reqstat コマンドで表示されるリクエスト ID を指定してください。

リクエストがキャンセルされるとメッセージが表示されます。

リスト 23 qdel コマンド

```
(sfront)$ qdel 1234.sjob
Request 1234.sjob was deleted.
```

4.4.10 バッチリクエストの終了

リクエストが終了すると、reqstat コマンドで表示されなくなります。終了したリクエストの実行結果ファイルとして、リクエスト実行中に標準出力された内容を保存した「標準出力ファイル」と、標準エラー出力の内容を保存した「標準エラー出力ファイル」が作成されます。

ジョブのスクリプトでオプション -o と -e を指定した場合はそのファイル名で作成されます。指定しない場合は以下のファイル名で作成されます。

- 標準出力ファイル： リクエスト名.o リクエスト ID
- 標準エラーファイル： リクエスト名.e リクエスト ID

なお、標準出力/標準エラー出力のファイルサイズの上限値は 1GB です。ファイルサイズが上限値を超えた場合はプログラムが強制終了しますので、実行結果はプログラム内でファイル名を指定して書き

出すようにしてください。

4.5 会話リクエストの投入と利用

AOBA-S では、qlogin コマンドを利用したセッション接続タイプの会話リクエストの投入が可能です。会話リクエストでは 1VH+8VE を利用して、ソースコードのコンパイル、会話型形式での実行、および GDB (ve-gdb) の利用が可能です。

会話リクエストはバッチリクエストと同様に演算負担額が発生します。会話リクエストのセッションが開始されてからセッションが切断されるまで、が演算負担額の対象となります。ジョブの実行を行っていない場合でも、セッションの接続時間が課金対象時間となりますのでご注意ください。

会話リクエストの最大経過時間は 1 時間です。会話リクエスト用ノードの利用状況により、セッションの開始まで実行待ちが発生することがあります。

4.5.1 qlogin による会話リクエストの投入方法

会話リクエストの投入は、AOBA-S 用フロントエンドサーバ上で qlogin コマンドで行います。リスト 24 に qlogin コマンドの例を示します。

リスト 24 qlogin コマンド

```
(sfront)$ qlogin -q inter -l elapstim_req=1800
# 会話キュー inter の指定 (必須) と最大経過時間の指定 (任意)
プロジェクトコード : un0000 にリクエストを投入します
Request 1234.sjob submitted to que: inter. # 会話リクエストが受け付けられた
Waiting for 1234.sjob to start. # 実行ホストとのセッション接続待ち
(sxat3001)$ # 実行ホストのシェルプロンプトに表示が変わる
(sxat3001)$ exit # セッションの切断
(sfront)$ # シェルプロンプト表示に戻る
```

リスト 25 に ve-gdb (VE 用 GDB) の起動方法を示します。

リスト 25 ve-gdb コマンド

```
(sxat3001)$ ve-gdb a.out
No symbol table is loaded. Use the "file" command.
GNU gdb (GDB) 7.12.1-8.e18
Modified by NEC Corporation for the VE port, 2017-2019
Modified by Arm. Copyright (C) 2002-2019 Arm Limited (or its affiliates).
All rights reserved.
(省略)
For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from a.out...(no debugging symbols found)...done.
(gdb)
```

VE 用の GDB と一般的な GDB の相違点については、6 章の GDB の相違点をご参照ください。

5 SX-Aurora TSUBASA 用数値演算ライブラリ

5.1 NLC (NEC Numeric Library Collection)

NEC Numeric Library Collection は、広範な分野の数値シミュレーションプログラムの作成を強力に支援する数学ライブラリのコレクションであり、Vector Engine に対応しています。NEC Numeric Library Collection を用いることにより、難解な数値計算アルゴリズムの詳細に煩わされることなく高度な科学技術計算プログラムを作成することができ、数値シミュレーションプログラム開発の生産性を大幅に改善することができます。

NEC Numeric Library Collection は、Fortran または C 言語プログラムから利用できます。

5.1.1 Fortran プログラムから利用する場合

Fortran プログラムから利用する場合、表 9 に示すライブラリが使用できます。

表 9 Fortran 用 NLC の機能概要

ライブラリ名		機能概要
ASL	ネイティブインタフェース	数値計算・統計計算のための各種アルゴリズムを備えた科学技術計算ライブラリ
	統合インタフェース	フーリエ変換、乱数、ソート
	FFTW3 インタフェース	FFTW (version 3.x) の API で ASL のフーリエ変換を利用するためのインタフェースライブラリ
BLAS		ベクトル、行列の基本演算
LAPACK		連立 1 次方程式、固有値方程式、特異値分解
ScaLAPACK		連立 1 次方程式、固有値方程式、特異値分解 (分散メモリ並列用)
BLACS		ベクトル、行列の基本演算のためのメッセージパッシングライブラリ (分散メモリ並列用)
SBLAS		スパース行列の基本演算
HeteroSolver		連立 1 次方程式 (スパース行列用の直接法ソルバ)
Stencil Code Accelerator		ステンシル計算の加速

5.1.2 C プログラムから利用する場合

C プログラムから利用する場合、表 10 に示すライブラリが使用できます。

表 10 C 言語用 NLC の機能概要

ライブラリ名		機能概要
ASL	ネイティブインタフェース	数値計算・統計計算のための各種アルゴリズムを備えた科学技術計算ライブラリ
	統合インタフェース	フーリエ変換、乱数、ソート
	FFTW3 インタフェース	FFTW (version 3.x) の API で ASL のフーリエ変換を利用するためのインタフェースライブラリ
CBLAS		BLAS C 言語インタフェース
SBLAS		スパース行列の基本演算
HeteroSolver		連立 1 次方程式 (スパース行列用の直接法ソルバ)
Stencil Code Accelerator		ステンシル計算の加速

NLC の詳細、およびコンパイルとリンク方法については 6 章の NLC (NEC Numeric Library Collection) ユーザーズガイドをご参照ください。

6 マニュアル

サブシステム AOBA-S についてのマニュアルはオンライン上で公開されています。以下リンク先の NEC Aurora Forum の NEC SX-Aurora TSUBASA Documentation をご参照ください。英語版、日本語版ともに提供されています。

【NEC Aurora Forum Documentation】 <https://www.hpc.nec/documentation>

当センター独自の運用方法により、マニュアルに記載の事項が動作しない場合もありますのでご注意ください。

6.1 コンパイラマニュアル

コンパイラについては以下のマニュアルをご参照ください。

■SDK

- C/C++ Compiler ユーザーズガイド、C/C++ Compiler User's Guide
- Fortran Compiler ユーザーズガイド、Fortran Compiler User's Guide
- NEC Parallel Debugger ユーザーズガイド、NEC Parallel Debugger User's Guide

■NEC MPI

- NEC MPI ユーザーズガイド、NEC MPI User's Guide

6.2 性能情報取得についてのマニュアル

実行時の性能情報取得については以下のマニュアルをご参照ください。

■SDK

- PROGINF/FTRACE ユーザーズガイド、PROGINF/FTRACE User's Guide
- NEC Ftrace Viewer ユーザーズガイド、NEC Ftrace Viewer User's Guide

6.3 NQSV についてのマニュアル

NQSV の利用方法については以下のマニュアルをご参照ください。

■NQSV

- NQSV 利用の手引 操作編、NQSV User's Guide Operation

6.4 GDB についてのマニュアル

VE 用の GDB と一般的な Linux の GDB の使用方法に関する相違点については以下の資料をご参照ください。

■VEOS

- GDB の相違点

6.5 数値計算ライブラリ (NLC) マニュアル

数値計算ライブラリ (NLC) については以下のマニュアルをご参照ください。

■SDK

- NLC (NEC Numeric Library Collection) ユーザーズガイド、
NLC (NEC Numeric Library Collection) User's Guide

7 おわりに

本稿では、スーパーコンピュータ AOBA のサブシステム AOBA-S のプログラミング利用ガイドとして基本的な手順を紹介しました。研究室のサーバでは実現できなかったプログラムやアイデアを、ぜひ最新鋭のスーパーコンピュータでお試してください。研究の強力なツールとしてご活用いただければ幸いです。

ご不明な点、ご質問等がありましたら、お気軽にセンターまでお問い合わせください。お問い合わせについては利用相談フォームをご利用ください。

【利用相談フォーム】 <https://www.ss.cc.tohoku.ac.jp/consultation/>

センターから運用に関するお知らせは以下をご参照ください。

【センターからのお知らせ】 <https://www.ss.cc.tohoku.ac.jp/information/>

[大規模科学計算システム] 【A0BA-A および A0BA-B の利用法】

鍵ペアの作成とログイン方法

情報部デジタルサービス支援課

1. はじめに

本センターのシステムは、セキュリティ対策として、公開鍵認証方式による SSH 接続を採用しています。また、フロントエンドサーバは、ログインサーバを経由しなければログインできない構成としています。

本稿では、公開鍵認証方式で使用する鍵ペアの作成と各サーバのログイン方法についてご紹介します。解説では以下のターミナルソフトを使用する例をご紹介します。

(Windows の場合) Windows PowerShell
(macOS/Linux の場合) ターミナル

本センターのシステムをはじめて利用する方は、以下の手続きが必要です。

- (1) 利用者番号の取得 (利用申請 : <https://www.ss.cc.tohoku.ac.jp/apply-for-use/>)
- (2) 鍵ペアの作成 (4 章)

以前のシステムを利用していた方は、(1)(2)の手続きは不要です。以前使用していた利用者番号および鍵ペアをそのままご利用いただけます。5 章からお読みください。

2. ログイン認証方式

表 1 に、各サーバのログイン認証方式を示します。

表 1 各サーバのログイン認証方式

サーバ名	用途	ログインホスト名	認証方式
ログインサーバ	フロントエンドサーバの入口 (踏み台サーバ)	login.cc.tohoku.ac.jp	公開鍵
フロントエンドサーバ	計算機の利用 (コンパイル、ジョブ実行、等)	(※1)	公開鍵またはパスワード
データ転送サーバ	ストレージシステムとの大容量のデータ転送	file.cc.tohoku.ac.jp	公開鍵
HPCI 用ログインノード	HPCI、HPCI-JHPCN ユーザ専用ログインノード	hpcif.cc.tohoku.ac.jp	公開鍵
-	センター内施設の利用(※2)	-	パスワード

(※1) フロントエンドサーバは、ログインサーバからしかログインできません。本稿では多段 SSH による接続方法を解説します。

(※2) 本センター内の施設（大判カラープリンタ、利用者端末、講習会端末）はパスワード認証でご利用いただけます。利用にあたり、秘密鍵を持参する必要はありません。

3. 鍵ペアの作成からログインまでの流れ

図 1 に、鍵ペア作成からログインまでの流れを示します。①は 4 章、②③は 5 章で詳しく解説します。

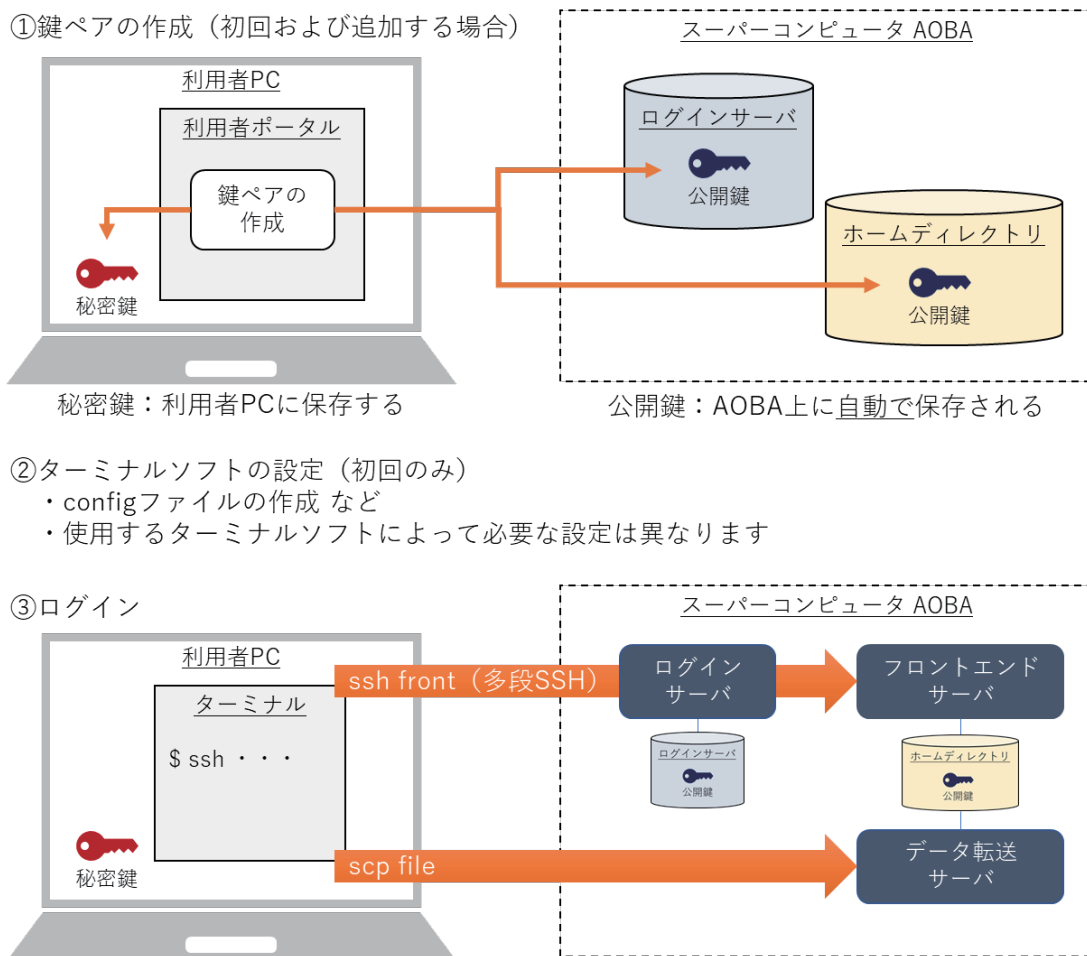


図 1 鍵ペア作成からログインまでの流れ

① 鍵ペアの作成（初回ログイン時、および、ログイン端末を追加する場合）

利用者ポータルで鍵ペアを作成します。作成された秘密鍵は、利用者のローカル PC に保存します。公開鍵は、スーパーコンピュータ AOA のホームディレクトリ上に自動で保存されます。

② ターミナルソフトの設定（初回ログイン時）

各サーバにログインするための設定を行います。使用するターミナルソフトによって必要な設定は異なります。

③ ログイン

利用者のローカル PC に保存した秘密鍵を使ってログインします。フロントエンドサーバは、ログインサーバを経由して多段 SSH でログインします。

4. 公開鍵認証方式で使用する鍵ペアの作成

4.1. 公開鍵認証方式を使用する上での注意事項

以下の注意事項を必ず守ってください。守らない場合、不正アクセス（不正ログイン、クライアントのなりすまし、暗号化された通信の暴露、他サーバへの攻撃、等）のリスクが非常に高まり、大変危険です。ご注意ください。

- ・ パスフレーズなしの秘密鍵を使用しないこと
- ・ 秘密鍵、パスフレーズを使いまわさないこと
- ・ 秘密鍵を持ち出さないこと（メールに添付しない、USB メモリ等に保存しない）
- ・ 秘密鍵をスーパーコンピュータ AOBA のホームディレクトリに保存しないこと
- ・ 公開鍵と秘密鍵の鍵ペアを同一ノード上に保存しないこと

4.2. 鍵ペアの作成（初回ログイン時、および、ログイン端末を追加する場合）

○初回ログイン時

鍵ペアの作成は、利用者ポータルで行います。

(1) 以下の URL 先から利用者ポータルを開きます。

利用者ポータルには、利用者番号とパスワード（※）でログインします。

利用者ポータル：<https://www.ss.cc.tohoku.ac.jp/portal/>

(2) 「SSH 公開鍵登録」ボタンをクリックします。

(3) 利用者ポータルの画面の説明に従い、鍵ペアを作成します。

（パスフレーズを設定し、鍵生成・登録ボタンをクリック）

(4) 作成された秘密鍵を利用者のローカル PC に保存します。保存先は以下を推奨します。

フォルダがない場合は新規作成します。

（Windows の場合） C:¥Users¥ユーザ¥.ssh

（macOS/Linux の場合） ~/.ssh

公開鍵は、ホームディレクトリ（~/.ssh/authorized_keys）に自動で保存されます。

※利用者ポータルで使用するパスワードの変更方法は、6 章を参照してください。

○別の PC からログインする場合（ログイン端末を追加する場合）

既存の秘密鍵を使いまわすのではなく、ログイン端末ごとに鍵ペアを作成してください。初回ログイン時と同じ手順で、新しい鍵ペアを追加します。

5. 各サーバのログイン方法

5.1. ターミナルソフトの設定（初回ログイン時）

利用者のローカル PC 上で、ターミナルソフトの設定を行います。

以降の解説は、次のフォルダを「.ssh フォルダ」と呼び、秘密鍵を「id_rsa_cc」というファイル名で.ssh フォルダに保存した場合とします。

（Windows の場合） C:¥Users¥ユーザ¥.ssh

（macOS/Linux の場合） ~/.ssh

各ログインホストのホスト名は、次の文字列で設定するものとして解説します。ホスト名には任意の文字列を設定することができます（他の設定との重複は不可）。他の文字列を設定した場合は、以降の解説におけるホスト名を読み替えてください。

（ログインサーバ） login

（フロントエンドサーバ） front

（データ転送サーバ） file

（HPCI 用ログインノード） hpcif

(1) macOS/Linux の場合は、秘密鍵のパーミッションの変更（600 に設定）が必要です。ターミナルソフトを起動し、以下のコマンドを実行します。

```
$ chmod 600 ~/.ssh/id_rsa_cc
```

以降は Windows、macOS/Linux 共通です。

(2) .ssh フォルダの「config」というファイルをテキストエディタで開きます。ファイルがない場合は新規作成します。拡張子はありません。

（フォルダの設定を「拡張子を表示しない」にしている場合、意識せずに拡張子付きのファイルを作成している可能性があります。config ファイルに拡張子がついていると、ログインできません。ご注意ください）

(3) config ファイルに以下の設定を記述します。太字下線の部分は、ご自身の環境に合わせて読み替えてください。

○フロントエンドサーバを利用するための設定 (※)

```
# ログインサーバの設定 (ホスト名を”login”とする場合)
Host login # ホスト名を指定
HostName login.cc.tohoku.ac.jp # ログインホスト名を指定
User 利用者番号 # 利用者番号を指定
IdentityFile ~/.ssh/id_rsa_cc # 秘密鍵の保存場所とファイル名を指定

# フロントエンドサーバの設定 (ホスト名を”front”とする場合)
Host front
HostName front.cc.tohoku.ac.jp
User 利用者番号
ProxyCommand ssh -CW %h:%p login # login 経由で多段 SSH する設定
IdentityFile ~/.ssh/id_rsa_cc
```

○データ転送サーバを利用するための設定

```
# データ転送サーバの設定 (ホスト名を”file”とする場合)
Host file
HostName file.cc.tohoku.ac.jp
User 利用者番号
IdentityFile ~/.ssh/id_rsa_cc
```

○HPCI 用ログインノードを利用するための設定

```
# HPCI 用ログインノードの設定 (ホスト名を”hpcif”とする場合)
Host hpcif
HostName hpcif.cc.tohoku.ac.jp
User 利用者番号
IdentityFile ~/.ssh/id_rsa_cc
```

(※) Windows の場合、フロントエンドサーバへのログイン時に以下のようなエラーが出る場合があります。

```
$ ssh front
CreateProcessW failed error:2
posix_spawn: No such file or directory
```

エラーが出た場合は次の要領で config ファイルを書き換えてください。

[1] ターミナルソフトを起動し、以下のコマンドで ssh の絶対パスを調べる。

```
$ gcm ssh
CommandType Name Version Source
-----
Application ssh.exe x.x.x C:¥WINDOWS¥System32¥OpenSSH¥ssh.exe
```

[2] config ファイルの「ProxyCommand ssh …」の行の「ssh」の部分、絶対パス ([1] で「Source」に表示された文字列) に書き換える。

```
# 修正前
ProxyCommand ssh -CW %h:%p login
# 修正後
ProxyCommand C:¥WINDOWS¥System32¥OpenSSH¥ssh.exe -CW %h:%p login
```

5.2. フロントエンドサーバのログイン方法

ターミナルソフトを起動し、以下のコマンドを実行するとログインします。ホスト名を別の文字列で設定している場合は「front」の部分を読み替えてください。

```
$ ssh front
```

フロントエンドサーバは冗長構成になっており、自動的に front1 または front2 が選択されます。どちらにログインしても、動作は変わりません。

なお、フロントエンドサーバでは一定時間以上のプロセスは実行できません。また、大容量のデータ転送はシステムに高い負荷がかかります。大容量のデータ転送を行う場合は、データ転送サーバをご利用ください。

5.3. データ転送サーバの利用方法

データ転送サーバは、ログインして利用するのではなく、利用者のローカル PC 上から scp コマンドや sftp コマンドで利用します。詳しくは以下をご参照ください。

データ転送 (ストレージ) : <https://www.ss.cc.tohoku.ac.jp/storage/>

5.4. HPCI 用ログインノードのログイン方法

ターミナルソフトを起動し、以下のコマンドを実行するとログインします。ホスト名を別の文字列で設定した場合は「hpcif」の部分を読み替えてください。

```
$ ssh hpcif
```

5.5. ログインシェルの確認と変更

ログインシェルは、デフォルトでは `bash` が設定されています。設定の確認および変更は以下の手順で行います。ログインシェルの変更がシステム全体に反映されるまで、15 分程度かかります。

- (1) フロントエンドサーバにログインする。
- (2) 以下のコマンドを実行する。

○ログインシェルの確認

```
front1 $ fchsh (ログインシェルの確認)
Enter Password: (パスワードを入力)
loginShell: /bin/bash (現在のログインシェルが表示される)
```

○ログインシェルの変更

```
front1 $ fchsh /bin/tcsh (ログインシェルを/bin/tcshに変更)
Enter Password: (パスワードを入力)
Changed loginShell to /bin/tcsh (ログインシェルが変更された)
```

6. パスワードの変更

利用者ポータルなどで使用するパスワードの変更は、以下の手順で行います。

- (1) 以下の URL 先から利用者ポータルを開きます。
利用者ポータルには、利用者番号とパスワードでログインします。
利用者ポータル : <https://www.ss.cc.tohoku.ac.jp/portal/>
- (2) 「パスワード変更」 ボタンをクリックします。
- (3) 利用者ポータルの画面の説明に従い、新しいパスワードを設定します。
- (4) 以下で使用するパスワードが変更されます。
 - ・利用者ポータルへのログインパスワード
 - ・大判カラープリンタのプリンタサーバへのログイン
 - ・ログインシェルの変更時のパスワード

7. おわりに

本稿では、鍵ペアの作成とログイン方法についてご紹介しました。センターのシステムを安全にご利用いただければ幸いです。ご不明な点、ご質問等ございましたら、お気軽にセンター（利用相談）までお問い合わせください。

利用相談 : <https://www.ss.cc.tohoku.ac.jp/consultation/>

また、センターからのお知らせは、ウェブサイトにてご確認ください。

センターウェブサイト : <https://www.ss.cc.tohoku.ac.jp/>

[大規模科学計算システム] 【AOBA-A および AOBA-B の利用法】

サブシステム AOBA-A の利用法

情報部デジタルサービス支援課

1 はじめに

本センターはスーパーコンピュータ AOBA のサブシステム AOBA-A の運用を 2020 年 10 月から開始しています。本稿では、サブシステム AOBA-A でのプログラミング利用ガイドとして、プログラムの作成からコンパイル、実行等の使い方についてご紹介します。

サイバーサイエンスセンターの大規模科学計算システムを利用するためには利用申請が必要です。利用申請について詳しくは「利用申請」(<https://www.ss.cc.tohoku.ac.jp/apply-for-use/>) 参照ください。

2 システムの構成

本センターでは、スーパーコンピュータ AOBA として、サブシステム AOBA-S (SX-Aurora TSUBASA Type 30A) と、サブシステム AOBA-A (SX-Aurora TSUBASA Type 20B)、およびサブシステム AOBA-B (LX 406Rz-2) の 3 つの計算機システムをサービスしています (図 1)。また、AOBA-S 用のストレージとして 5.6PB、AOBA-A および AOBA-B 用のストレージとして 2PB のストレージシステムをサービスしています。

利用者は SINET6 を利用したりリモートアクセス接続により、全国から大規模科学計算システムを利用することが可能です。

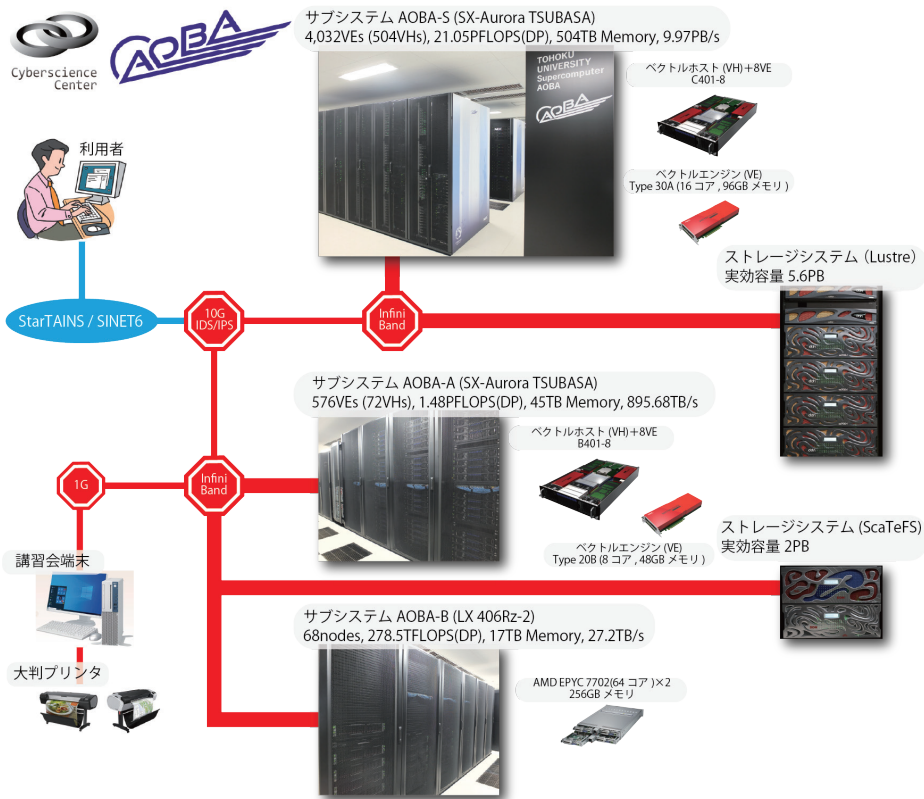


図 1 スーパーコンピュータ AOBA の構成

2.1 サブシステム AOBA-A の特徴

2.1.1 SX-Aurora TSUBASA アーキテクチャ

SX-Aurora TSUBASA（日本電気株式会社製）は、前システム SX-ACE と同じくベクトルアーキテクチャを継承しています。アプリケーション演算処理を行うベクトルエンジン（以下、VE）部と、主に OS 処理を行うベクトルホスト（以下、VH）部により構成されます。PCIe カードに搭載される VE 部はベクトルプロセッサ、及び高速メモリから構成され、x86/Linux である VH と PCIe 経由で接続されます。

1VE は理論最大演算性能 2,456GFLOPS(DP) となるマルチコア (8 コア) ベクトルプロセッサを 1 基、主記憶は 48GB を搭載し、1.53TB/s という高いメモリバンド幅でプロセッサと接続されることで、高い演算性能とメモリ性能の最適化を実現しています。

今回導入した SX-Aurora TSUBASA のシステムは、1VH と 8VE が構成単位となる B401-8 モデルを採用し、システム全体では 72 個の VH と 576 個の VE で構成されます。VE と VH を合わせたシステム全体の理論演算性能は、1.48PFLOPS(DP)、主記憶は 45TB、総メモリバンド幅は 895.68TB/s となります。



図 2 サブシステム AOBA-A (4 ラック)

- マルチコアベクトルエンジン (8 コア) あたり、2,456GFLOPS(DP) の理論演算性能
- 1VE あたり 48GB の共有メモリを搭載し、メモリバンド幅 (データ転送性能) は 1.53TB/s

2.1.2 ベクトルプロセッサ

ベクトルプロセッサとは、ベクトル演算を行う専用のハードウェアを持つコンピュータです。ベクトル演算は、ループ中で繰り返し処理されるような配列データの演算に対して一括して演算を実行するため、高速に演算することができます。

ベクトル計算機は科学技術用の数値計算に適しており、大量のデータを繰り返し処理するような大規模計算に向いていると言われています。本センターでは、流体解析や気象解析、電磁界解析をはじめとする大規模シミュレーションに利用されます。

2.1.3 ソフトウェア

Linux OS 環境上で、ベクトルエンジンの開発環境を利用できます。ベクトル処理、並列処理に対応した大規模プログラムの作成と実行が可能です。

サブシステム AOBA-A 向けにインストールされているアプリケーションについては、「アプリケーションサービス」(<https://www.ss.cc.tohoku.ac.jp/software-service/>)をご参照ください。

■**プログラミング言語** GNU 互換環境を装備し、アプリケーションの実効性能を向上させる高度な自動ベクトル化・自動並列化機能を備えた Fortran/C/C++ コンパイラが利用出来ます。それぞれのコンパイラは自動ベクトル化機能、自動並列化機能を有していますので、既存のプログラムを修正することなくベクトル化、並列化することができます。自動並列化機能と OpenMP による共有メモリ並列実行と、システム構成に最適化された MPI ライブラリにより、分散メモリ並列実行も可能です。

■**科学技術計算ライブラリ** 業界標準の BLAS, FFTW, LAPACK, ScaLAPACK を含む、最適化された科学技術計算ライブラリを利用出来ます。NEC Numeric Library Collection (NLC) は、広範な分野の数値シミュレーションプログラムの作成を強力に支援する数学ライブラリのコレクションで、VE に対応しています。NLC を用いることにより、難解な数値計算アルゴリズムの詳細に煩わされることなく高度な科学技術計算プログラムを作成することができ、数値シミュレーションプログラム開発の生産性を大幅に改善することができます。NLC は、Fortran または C 言語プログラムから利用できます。

2.2 プログラムの実行方法

プログラムの実行の方法として表 1 の 4 種類が利用できます。並列実行には 3 種類があります。

■**逐次実行** 単一のコアで動作するプログラムです。VE 内には 8 コアがありますが、逐次実行では 1 コアのみが利用されます。メモリは 48GB まで利用出来ます。

■**自動並列化** 逐次処理プログラムを、コンパイラが並列化可能な箇所を自動的に判断して並列化します。プログラムを改めて書き直す必要はなく、プログラムをそのまま利用することができます。

共有メモリ並列実行が可能なので、VE 内の 8 コアまで、メモリは 48GB まで利用出来ます。

■**OpenMP 並列** 自動並列化と同じく逐次処理プログラムを並列化します。並列化の判断は自動ではなくユーザが明示的に行います。ソース中の並列化したい箇所に並列化指示行を追加します。

自動並列化と同じく、共有メモリ並列実行が可能なので、VE 内の 8 コアまで、メモリは 48GB まで利用出来ます。

■**MPI 並列** MPI ライブラリによりプロセッサ間通信を行います。データの分割、処理方法等の並列処理手順を明示的に記述した MPI プログラムを作成する必要があります。

分散メモリ並列実行が可能なので、複数の VE を用いた実行が可能です。センターでは最大 256VE (2,048 コア)、メモリは 12TB まで利用出来ます。

MPI 並列と自動並列化／OpenMP 並列を同時に利用した並列実行も可能です。

表 1 実行の種類, 特徴, 最大並列数, 最大メモリ量

実行の種類	並列化の方法	コードの改変量	最大並列数	最大メモリ量
逐次実行	-	-	1 コア	48GB
自動並列化	コンパイラによる自動並列化	なし	8 コア	48GB
OpenMP 並列	ユーザによる指示行挿入	少ない	8 コア	48GB
MPI 並列	MPI ライブラリを用いたプログラミング	多い	2,048 コア	12,288GB

3 プログラムのコンパイルと実行手順

本章ではプログラムのコンパイルと、サブシステム AOBA-A で実行する手順を説明します。

3.1 フロントエンドサーバへのログイン

サブシステム AOBA-A 用のコンパイルはフロントエンドサーバで行います。セキュリティ対策のため、フロントエンドサーバへはログインサーバを経由します。ストレージシステムとの大容量のデータ転送は、データ転送サーバを利用します。いずれのサーバにも公開鍵暗号方式による SSH 接続でログインを行います。

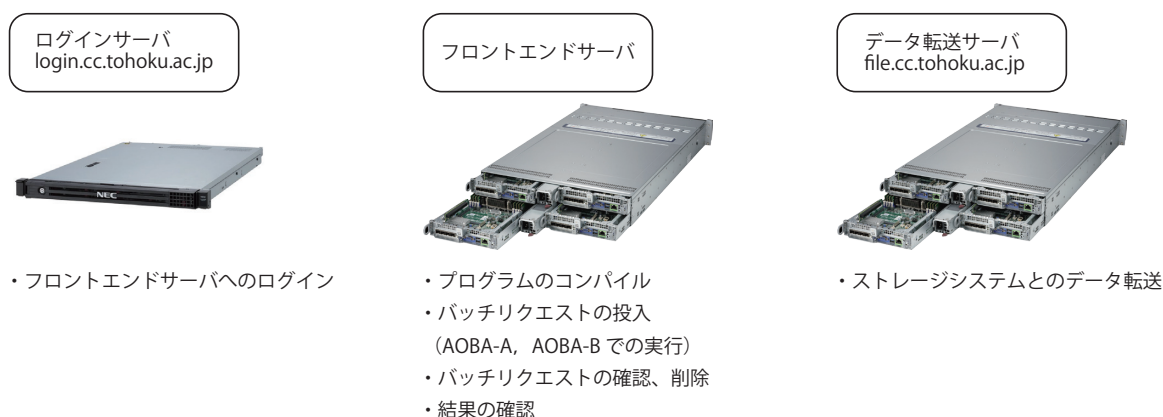


図 3 ログインサーバ, フロントエンドサーバ, データ転送サーバ

鍵の作成からログインサーバを経由してフロントエンドサーバへログインする方法については、利用申請からログインまで (<https://www.ss.cc.tohoku.ac.jp/first-use/>) をご参照ください。

データ転送サーバの利用方法や、ストレージ容量の追加方法などについては、「データ転送 (ストレージ)」 (<https://www.ss.cc.tohoku.ac.jp/storage/>) をご参照ください。

3.2 プログラムの作成とコンパイル

3.2.1 ソースファイルの作成

フロントエンドサーバ上でソースファイルを作成するためのエディタは、emacs または vim が一般的です。研究室等のサーバやパソコンからソースファイルを転送する場合は、SSH 対応のファイル転送ソフトや scp コマンドで利用者のホームディレクトリに転送してください。その際、ソース (テキスト

ト) ファイルは ASCII モードで転送します。Windows 環境で作成したソースコードの改行コードと文字コードを Linux 用にするためには、転送した後のソースコードに `nkf` コマンドを利用します (リスト 1)。

リスト 1 nkf コマンドの使用例

```
改行コードを LF, 文字コードを UTF-8 に変換し, 別のファイルに書き出す方法
front$ nkf -Lu 変換前ソースファイル名 > 変換後ソースファイル名

改行コードを LF, 文字コードを UTF-8 に変換してソースファイルを上書き保存する方法
front$ nkf --overwrite -Lu ソースファイル名
```

3.2.2 コンパイラの仕様

SX-Aurora TSUBASA 用に最適化された Fortran および C/C++ コンパイラを利用できます。コンパイラの仕様は以下の通りです。

Fortran コンパイラ

- `nfort`, `mpinfort` コマンドでコンパイル
- 自動ベクトル化・自動並列化機能機能対応
- ISO/IEC 1539-1:2004 Programming languages - Fortran に準拠
- OpenMP Application Program Interface Version 4.5 に準拠
- ISO/IEC 1539-1:2010 Programming languages - Fortran の一部機能にも対応

C/C++ コンパイラ

- `ncc`, `mpincc`, `nc++`, `mpinc++` コマンドでコンパイル
- 自動ベクトル化・自動並列化機能機能対応
- ISO/IEC 9899:2011 Programming languages - C に準拠
- ISO/IEC 14882:2014 Programming languages - C++ に準拠
- OpenMP Application Program Interface Version 4.5 に準拠

3.2.3 コンパイルを行う

ソースファイルはそれぞれの言語用コマンドでコンパイルします。単一コアで実行する逐次実行、自動並列化または OpenMP による共有メモリ並列実行および MPI ライブラリによる分散メモリ並列実行のコンパイル手順について解説します。

■**Fortran プログラム** ソースコードは `nfort` コマンドまたは `mpinfort` コマンドでコンパイルします。利用したい機能があれば、表 2 に示したようなコンパイルオプションを同時に指定します。その他のオプションについてはマニュアル (5 章) をご参照ください。

ソースファイルの拡張子は、自由形式 (フリーフォーマット) なら `.f90` `.f95` か `.F90` `.F95` を、固定形式 (7 カラム目から記述) なら `.f` か `.F` を、2003 形式であれば `.f03` か `.F03` を付けます。(拡張子が大文字で始まるものは、コンパイルの前に `fpp` によるプリプロセス処理が行われます。)

コンパイルが成功すると、カレントディレクトリに実行オブジェクト `a.out` が作成されます。

表2 Fortran,C,C++ コンパイラの主なオプション

コンパイルオプション	機能
-On n に指定できる値 4 3 2 1 0	最適化レベルを指定する 言語仕様を逸脱した副作用を伴う最大限の最適化・自動ベクトル化を適用する 副作用を伴う最適化・自動ベクトル化, および, 多重ループの最適化を適用する (既定値) 副作用を伴う最適化・自動ベクトル化を適用する 副作用を伴わない最適化・自動ベクトル化を適用する 最適化, 自動ベクトル化, 並列化, インライン展開を適用しない (最適化レベルを高くすると, 最適化の副作用により計算結果が変わることがありますのでご注意ください)
-finline-functions	自動インライン展開機能を適用する
-mparallel	自動並列化機能を利用する
-fopenmp	OpenMP 並列化を利用する
-mno-parallel-omp-routine	OpenMP ディレクティブを含むルーチンを自動並列化しない (-mparallel と -fopenmp が同時に指定されたとき, ループが OpenMP の並列区間の外側にある場合は外側のループが自動並列化の対象となります)
-ftrace	性能解析 ftrace 機能用の実行ファイルを作成する
-report-diagnostics	ベクトル診断リストを出力する
-fdiag-vector=2	詳細なベクトル化診断メッセージを出力する (既定値は 1)
-report-format	ベクトル化, 並列化などの最適化情報がソース行とともに出力された編集リストを出力する
-fbounds-check	バウンズチェック (配列の上下限のチェック) を行う

リスト 2 nfort コマンドの使用例

(逐次実行) front\$ nfort コンパイルオプション Fortranソースファイル名
(自動並列化実行) front\$ nfort -mparallel コンパイルオプション Fortranソースファイル名
(OpenMP並列実行) front\$ nfort -fopenmp コンパイルオプション Fortranソースファイル名

リスト 3 mpinfort コマンドの使用例

(MPI並列実行) front\$ mpinfort コンパイルオプション Fortranソースファイル名
(MPIと自動並列化の同時並列実行) front\$ mpinfort -mparallel コンパイルオプション Fortranソースファイル名
(MPIとOpenMPの同時並列実行) front\$ mpinfort -fopenmp コンパイルオプション Fortranソースファイル名

■C/C++ プログラム ソースコードは ncc または mpincc コマンドで C プログラムを, nc++ または mpinc++ コマンドで C++ プログラムをコンパイルします。利用したい機能があれば Fortran コンパイラと同じく, 表 2 に示したようなコンパイルオプションを同時に指定します。その他のオプションについてはマニュアル (5 章) をご参照ください。

ソースファイルの拡張子は, C 言語であれば.c を, C++ であれば.C .cc .cpp .cp .cxx .c++ のいずれかを付けます。

コンパイルが成功すると、カレントディレクトリに実行オブジェクト a.out が作成されます。

リスト 4 ncc/nc++ コマンドの使用例

```

(逐次実行)
front$ ncc コンパイルオプション Cソースファイル名
front$ nc++ コンパイルオプション C++ソースファイル名

(自動並列化実行)
front$ ncc -mparallel コンパイルオプション Cソースファイル名
front$ nc++ -mparallel コンパイルオプション C++ソースファイル名

(OpenMP並列実行)
front$ ncc -fopenmp コンパイルオプション Cソースファイル名
front$ nc++ -fopenmp コンパイルオプション C++ソースファイル名
    
```

リスト 5 mpincc/mpinc++ コマンドの使用例

```

(MPI並列実行)
front$ mpincc コンパイルオプション Cソースファイル名
front$ mpinc++ コンパイルオプション C++ソースファイル名

(MPIと自動並列化の同時並列実行)
front$ mpincc -mparallel コンパイルオプション Cソースファイル名
front$ mpinc++ -mparallel コンパイルオプション C++ソースファイル名

(MPIとOpenMPの同時並列実行)
front$ mpincc -fopenmp コンパイルオプション Cソースファイル名
front$ mpinc++ -fopenmp コンパイルオプション C++ソースファイル名
    
```

■性能解析情報 (FTRACE) を確認する コンパイル時に-ftrace オプションを指定すると、実行後にディレクトリに性能解析情報の結果ファイルが書き出されます。MPI 並列実行時の性能情報も取得可能です。

結果の確認は、ftrace コマンドでテキスト形式で確認するか、ftraceviewer コマンドで GUI で確認することが出来ます。図 4 は Ftrace Viewer の表示例です。

FTRACE と Ftrace Viewer についての詳細は、5 章に記載のマニュアルをご参照ください。

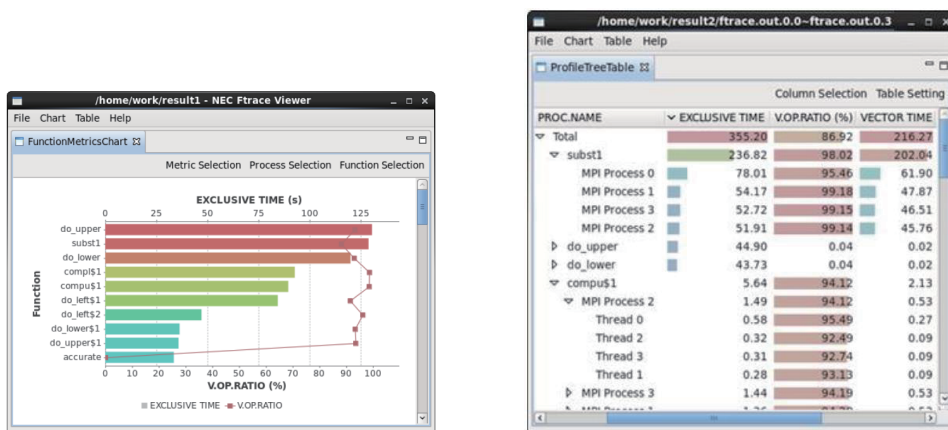


図 4 Ftrace Viewer の表示例

3.3 プログラムの実行

フロントエンドサーバでコンパイル作業を行って作成したプログラム (ジョブ) の実行は、バッチリクエストと呼ばれる方法で計算機に実行を依頼します。本センターではバッチリクエストの処理に NEC Network Queuing System V (以下, NQSV) を採用しています。図 5 にバッチリクエストの概念図を示します。

詳しくは「ジョブの実行方法」(<https://www.ss.cc.tohoku.ac.jp/nqs/>) をご参照ください。

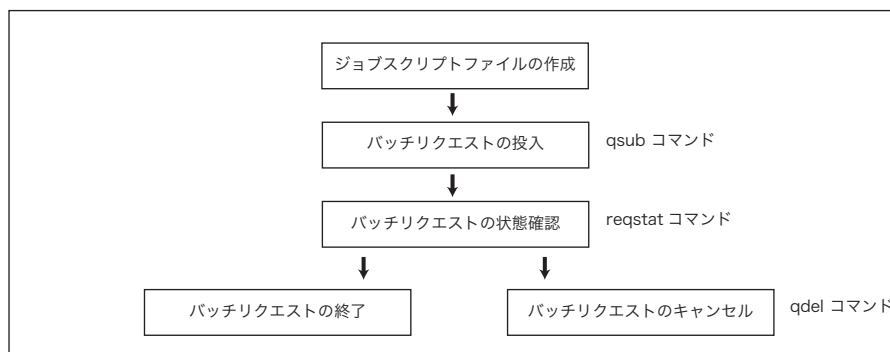


図5 バッチリクエストの概念図

3.3.1 ジョブスクリプト

ジョブスクリプトに指定が必要となるのは、

- 投入キュー名
- 利用する VE 数
- 最大経過時間
- 作業ディレクトリへの移動コマンド
- プログラムの実行コマンド

です。他に環境変数の指定、ファイルの操作コマンド等があれば適切な箇所に手続きを記述します。オプション、環境変数はジョブスクリプト中では#PBS の後に記述します。

サブシステム AOBA-A で実行する場合の、必須オプションを表3に、利用形態を表4に示します。

表3 サブシステム AOBA-A で実行する場合の必須オプション

オプション	指定するもの	機能
-q	sx sxf のどちらか 1 つ	AOBA-A の利用形態を指定
--venode	1~256 の整数値	利用する VE 数を指定
-l elapstim_req	最大経過時間を hh:mm:ss 形式で	計算機を利用する時間を指定

表4 サブシステム AOBA-A で実行する場合の利用形態

投入キュー名	利用可能 VE 数	リクエストの実行形態
sxf	1	無料の 1VE リクエスト (VH を共用する)
sx	1	1VE リクエスト (VH を共用する)
sx	2~256	8VE 単位で確保 (VH を共用しない)

VH を共用しない： 投入したリクエストが他のリクエストと VH を共用しないで実行されます。2~7 個の VE を使うと指定をしたリクエストも、8 個の VE と 1 個の VH を確保します。他のリクエストと VH を共用しないため、演算時間のバラツキが少なくなります。

VH を共用する： 投入したリクエストは他のリクエストと VH を共用して実行されます。2 個の VE を使うと指定したリクエストは、他 6 個の VE で別のリクエストが実行されることがあります。指定し

た数の VE (2~8 個) を確保しやすく、混雑時にも待ち時間が短くなります。

3.3.2 ジョブスクリプトの例

■**逐次実行の場合** リスト 6 とリスト 7 に逐次実行の場合のジョブスクリプトの例を示します。逐次実行の場合、プログラムは 1VE 中の 1 コアで実行されます。

リスト 6 ジョブスクリプトの例 (逐次実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q sx #AOBA-A を使用する
#PBS --venode 1 #VE を1個使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を2時間に指定

cd $PBS_O_WORKDIR #qsub を実行したディレクトリに移動
./a.out #カレントディレクトリの a.out を実行
```

リスト 7 ジョブスクリプトの例 (逐次実行, 無料キュー)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q sxf #AOBA-A の無料キューを使用する
#PBS --venode 1 #VE を1個使う
#PBS -l elapstim_req=0:30:00 #最大経過時間を30分に指定

cd $PBS_O_WORKDIR #qsub を実行したディレクトリに移動
./a.out #カレントディレクトリの a.out を実行
```

■**自動並列化/OpenMP 並列実行の場合** リスト 8 に自動並列化/OpenMP 並列実行の場合のジョブスクリプトの例を示します。自動並列化/OpenMP 並列実行の場合、プログラムは 1VE 中の 2~8 コアで実行されます。実行コア数の指定は、環境変数 `VE_OMP_NUM_THREADS` で行います。逐次実行と同様に、`sxf` も指定できます。

リスト 8 ジョブスクリプトの例 (自動並列化/OpenMP 並列実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q sx #AOBA-A を使用する
#PBS --venode 1 #VE を1個使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を2時間に指定

export VE_OMP_NUM_THREADS=8 #VE内は8コア並列で実行

cd $PBS_O_WORKDIR #qsub を実行したディレクトリに移動
./a.out #カレントディレクトリの a.out を実行
```

■**MPI 並列実行の場合** リスト 9 に MPI 並列実行の場合のジョブスクリプトの例を示します。MPI 並列実行の場合は、`mpirun` コマンドでプログラムの実行を行います。(逐次実行や自動並列化/OpenMP 並列実行用にコンパイルされたプログラムは、MPI 並列実行を行っても 1VE 中の 8 コア並列でしか実行されません。)

MPI プロセス数が確保した VE の物理コア数 (VE 数× 8) を超えると、演算性能が著しく低下しますのでご注意ください。

リスト 9 ジョブスクリプトの例 (MPI 並列実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q sx #AOBA-A を使用する
#PBS --venode 8 #VE を8個使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を2時間に指定

cd $PBS_O_WORKDIR #qsub を実行したディレクトリに移動
mpirun -np 64 ./a.out #カレントディレクトリの a.out を64プロセス並列で実行
```

■MPI と自動並列化／ OpenMP 同時並列実行の場合 リスト 10 に MPI と自動並列化／ OpenMP 同時並列実行の場合のジョブスクリプトの例を示します。

(MPI プロセス数) × (VE 内並列数) が確保した VE の物理コア数 (VE 数 × 8) を超えると、演算性能が著しく低下しますのでご注意ください。

リスト 10 ジョブスクリプトの例 (MPI と自動並列化／ OpenMP 同時並列実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q sx #AOBA-A を使用する
#PBS --venode 8 #VE を 8 個使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を 2 時間に指定

export VE_OMP_NUM_THREADS=8 #VE 内は 8 コア 並列で実行

cd $PBS_O_WORKDIR #qsub を実行したディレクトリに移動
mpirun -np 8 ./a.out #カレントディレクトリの a.out を 8 プロセス × 8 コア 並列で実行
```

標準出力、標準エラーファイルをプロセス毎に分割： 標準出力および標準エラー出力は、1 つのファイルに各プロセスからの出力が混在して書き出されます。このとき、システムで用意されたシェルスクリプト mpisep.sh を利用すると、同一ファイルにプロセス毎の出力が入り混じって出力されないように、MPI プロセスごとにファイルを分けて出力することができます。リスト 11 のようにシェルスクリプト /opt/nec/ve/bin/mpisep.sh を実行形式ファイル a.out の前に記述し、環境変数 NMPI_SEPSELECT に 1 から 4 の値を指定することで、表 5 に示した動作を選択します。

リスト 11 出力をプロセス毎に分割する方法

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q sx #AOBA-A を使用する
#PBS --venode 8 #VE を 8 個使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を 2 時間に指定
#PBS -v NMPI_SEPSELECT=3 #出力形式に 3 を指定

cd $PBS_O_WORKDIR #qsub を実行したディレクトリに移動
mpirun -np 64 /opt/nec/ve/bin/mpisep.sh ./a.out #カレントディレクトリの a.out を 64 プロセス 並列で実行
```

表 5 NMPI_SEPSELECT の引数と動作

NMPI_SEPSELECT の引数	動作
1	各 MPI プロセスの標準出力のみを stdout.uuu:rrr へ格納
2	各 MPI プロセスの標準エラーのみを stderr.uuu:rrr へ格納 (既定値)
3	各 MPI プロセスの標準出力と標準エラーをそれぞれ stdout.uuu:rrr および stderr.uuu:rrr へ格納
4	各 MPI プロセスの標準出力と標準エラーを同一ファイル stdout.uuu:rrr へ格納

実行時、stdout.*や stderr.*の同名ファイルが存在する場合は、上書きでなく追記します。

4 スーパーコンピュータ用数値演算ライブラリ

4.1 NLC (NEC Numeric Library Collection)

NEC Numeric Library Collection は、広範な分野の数値シミュレーションプログラムの作成を強力に支援する数学ライブラリのコレクションであり、Vector Engine に対応しています。NEC Numeric Library Collection を用いることにより、難解な数値計算アルゴリズムの詳細に煩わされることなく高度な科学技術計算プログラムを作成することができ、数値シミュレーションプログラム開発の生産性を大幅に改善することができます。

NEC Numeric Library Collection は、Fortran または C 言語プログラムから利用できます。

4.1.1 Fortran プログラムから利用する場合

Fortran プログラムから利用する場合、表 6 に示すライブラリが使用できます。

表 6 Fortran 用 NLC ライブラリと機能概要

ライブラリ名	機能概要
ASL ネイティブインタフェース	数値計算・統計計算のための各種アルゴリズムを備えた科学技術計算ライブラリ
ASL 統合インタフェース	フーリエ変換, 乱数, ソート
ASL FFTW3 インタフェース	FFTW (version 3.x) の API で ASL のフーリエ変換を利用するためのインタフェースライブラリ
BLAS	ベクトル, 行列の基本演算
LAPACK	連立 1 次方程式, 固有値方程式, 特異値分解
ScaLAPACK	連立 1 次方程式, 固有値方程式, 特異値分解 (分散メモリ並列用)
BLACS	ベクトル, 行列の基本演算のためのメッセージパッシングライブラリ (分散メモリ並列用)
SBLAS	スパース行列の基本演算
HeteroSolver	連立 1 次方程式 (スパース行列用の直接法ソルバ)
Stencil Code Accelerator	ステンシル計算の加速

4.1.2 C プログラムから利用する場合

C プログラムから利用する場合、表 7 に示すライブラリが使用できます。

表 7 C 言語用 NLC ライブラリと機能概要

ライブラリ名	機能概要
ASL ネイティブインタフェース	数値計算・統計計算のための各種アルゴリズムを備えた科学技術計算ライブラリ
ASL 統合インタフェース	フーリエ変換, 乱数, ソート
ASL FFTW3 インタフェース	FFTW (version 3.x) の API で ASL のフーリエ変換を利用するためのインタフェースライブラリ
CBLAS	BLAS C 言語インタフェース
SBLAS	スパース行列の基本演算
HeteroSolver	連立 1 次方程式 (スパース行列用の直接法ソルバ)
Stencil Code Accelerator	ステンシル計算の加速

NLC ライブラリの詳細およびコンパイルとリンク方法については 5 章の NLC (NEC Numeric Library Collection) ユーザーズガイドをご参照ください。

5 マニュアル

サブシステム AOBA-A についてのマニュアルは、NEC Aurora Forum の NEC SX-Aurora TSUBASA Documentation をご参照下さい。英語版, 日本語版ともに提供されています。

<https://www.hpc.nec/documentation>

当センター独自の運用方法により、マニュアルに記載の事項が動作しない場合もあります。

5.1 コンパイラマニュアル

コンパイラについては以下のマニュアルをご参照下さい。

- C/C++ Compiler ユーザーズガイド, C/C++ Compiler User's Guide
- Fortran Compiler ユーザーズガイド, Fortran Compiler User's Guide
- NEC MPI ユーザーズガイド, NEC MPI User's Guide

5.2 性能情報取得についてのマニュアル

実行時の性能情報取得については以下のマニュアルをご参照下さい。

- PROGINF/FTRACE ユーザーズガイド, PROGINF/FTRACE User's Guide
- NEC Ftrace Viewer ユーザーズガイド, NEC Ftrace Viewer User's Guide

5.3 数値計算ライブラリ (NLC) マニュアル

数値計算ライブラリ (NLC) については以下のマニュアルをご参照下さい。

- NLC (NEC Numeric Library Collection) ユーザーズガイド,
NLC (NEC Numeric Library Collection) User's Guide

6 おわりに

本稿では、スーパーコンピュータ AOBA のサブシステム AOBA-A のプログラミング利用ガイドとして基本的な手順を紹介しました。研究室のサーバでは実現できなかったプログラムやアイデアを、ぜひ最新鋭のスーパーコンピュータでお試してください。研究の強力なツールとしてご活用いただければ幸いです。

ご不明な点、ご質問等ございましたら、お気軽にセンターまでお問い合わせください。問い合わせについては「利用相談」(<https://www.ss.cc.tohoku.ac.jp/consultation/>)をご参照下さい。

[大規模科学計算システム] 【AOBA-A および AOBA-B の利用法】

サブシステム AOBA-B の利用法

情報部デジタルサービス支援課

1 はじめに

本センターはスーパーコンピュータ AOBA のサブシステム AOBA-B の運用を 2020 年 10 月から開始しています。本稿では、サブシステム AOBA-B でのプログラミング利用ガイドとして、プログラムの作成からコンパイル、実行等の使い方についてご紹介します。

サイバーサイエンスセンターの大規模科学計算システムを利用するためには利用申請が必要です。利用申請について詳しくは「利用申請」(<https://www.ss.cc.tohoku.ac.jp/apply-for-use/>) ご参照ください。

2 システムの構成

本センターでは、スーパーコンピュータ AOBA として、サブシステム AOBA-S (SX-Aurora TSUBASA Type 30A) と、サブシステム AOBA-A (SX-Aurora TSUBASA Type 20B)、およびサブシステム AOBA-B (LX 406Rz-2) の 3 つの計算機システムをサービスしています (図 1)。また、AOBA-S 用のストレージとして 5.6PB、AOBA-A および AOBA-B 用のストレージとして 2PB のストレージシステムをサービスしています。

利用者は SINET6 を利用したりリモートアクセス接続により、全国から大規模科学計算システムを利用することが可能です。

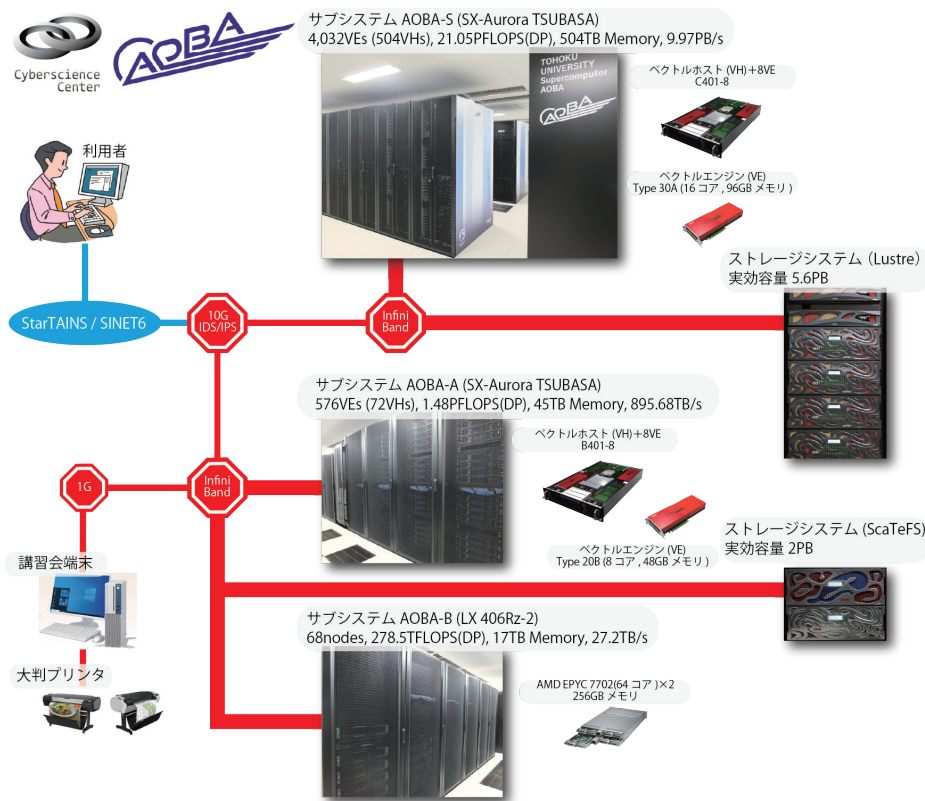


図 1 スーパーコンピュータ AOBA の構成

2.1 サブシステム AOBA-B の特徴

LX 406Rz-2 は、1 ノードに AMD EPYC プロセッサ 7702 (64 コア) を 2 基と 256GB の主記憶装置を搭載し、合計 68 ノードで構成されます。OpenMP・MPI を利用したノード内の並列処理は 128 並列まで可能で、ノードあたりの最大演算性能は 4.096TFLOPS (倍精度)、システム全体では 278.5 TFLOPS (倍精度) となります。複数のノードを使用した並列処理は、MPI の利用により最大 2,048 コア並列、4TB の主記憶を利用可能です。ベクトル演算に不向きなプログラムや、商用アプリケーションの高速な実行が可能です。



図2 サブシステム AOBA-B (4 ラック)

- 1 ノード (64 コア×2) あたり、4.096TFLOPS(DP) の理論演算性能
- 1 ノードあたり 256GB の共有メモリを搭載し、メモリバンド幅 (データ転送性能) は 409.6GB/s
- 一時領域として、アクセスが高速な SSD ストレージを搭載

2.1.1 ソフトウェア

フロントエンドサーバの Linux OS 環境上で、EPYC プロセッサ向けのプログラムを作成できます。一般的な Linux OS ですので、商用アプリケーションやオープンソフトウェアも実行が可能です。

サブシステム AOBA-B 向けにインストールされているアプリケーションについては、「アプリケーションサービス」(<https://www.ss.cc.tohoku.ac.jp/software-service/>) をご参照ください。

■**プログラミング言語** Fortran/C/C++ コンパイラとして、AMD Optimizing C/C++ Compiler と GNU Compiler Collection が利用出来ます。OpenMP による共有メモリ並列実行と MPI ライブラリによる分散メモリ並列実行が可能です。また、前システムの並列コンピュータで実行していたソースコードの移行用として、Intel Compiler をライセンス数限定で利用できます。

■**科学技術計算ライブラリ** EPYC プロセッサに最適化された、AMD Optimizing CPU Libraries (AOCL) を利用出来ます。

- AOCL-Sparse
- BLIS
- FFTW (Fastest Fourier Transform in the West)
- AMD Math Library (LibM)
- libFLAME
- ScaLAPACK
- AMD Random Number Generator Library

AOCL の詳細については <https://developer.amd.com/tools-and-sdks/> をご参照ください。

また、Intel Compiler からは MKL ライブラリが利用出来ます。MKL についての詳細は <https://www.xlsoft.com/jp/products/intel/perflib/mkl/index.html> をご参照ください。

■アプリケーション性能解析ツール AMD μ Prof が利用できます。実行したプログラムの性能およびシステム性能分析機能が利用出来ます。

AMD μ Prof の詳細については <https://developer.amd.com/tools-and-sdks/> をご参照ください。

2.2 プログラムの実行方法

プログラムの実行の方法として表 1 の 3 種類が利用できます。並列実行には 3 種類があります。

■逐次実行 単一のコアで動作するプログラムです。1 ノード内には 64 コアの CPU が 2 つありますが、逐次実行では 1 コアのみが利用されます。メモリは 256GB まで利用出来ます。

■OpenMP 並列 逐次処理プログラムをユーザが明示的に並列化します。ソース中の並列化したい箇所に並列化指示行を追加します。

共有メモリ並列実行が可能なので、1 ノード内の 128 コア (2CPU) を利用出来ます。メモリは 256GB まで利用出来ます。

■MPI 並列 MPI ライブラリによりプロセッサ間通信を行います。データの分割、処理方法等の並列処理手順を明示的に記述した MPI プログラムを作成する必要があります。

分散メモリ並列実行が可能なので、複数のノードを用いた実行が可能です。サブシステム AOBA-B では最大 16 ノード (2,048 コア)、メモリは 4,096GB まで利用出来ます。

MPI 並列と OpenMP 並列を同時に利用した並列実行も可能です。

表 1 実行の種類, 特徴, 最大並列数, 最大メモリ量

実行の種類	並列化の方法	コードの改変量	最大並列数	最大メモリ量
逐次実行	-	-	1 コア	256GB
OpenMP 並列	ユーザによる指示行挿入	少ない	128 コア	256GB
MPI 並列	MPI ライブラリを用いたプログラミング	多い	2,048 コア	4,096GB

3 プログラムのコンパイルと実行手順

本章ではプログラムのコンパイルと、サブシステム AOBA-B で実行する手順を説明します。コンパイラについては AOCC を使用した場合について説明します。

3.1 フロントエンドサーバへのログイン

サブシステム AOBA-B 用のコンパイルはフロントエンドサーバで行います。セキュリティ対策のため、フロントエンドサーバへはログインサーバを経由します。ストレージシステムとの大容量のデータ転送は、データ転送サーバを利用します。いずれのサーバにも公開鍵暗号方式による SSH 接続でログインを行います。

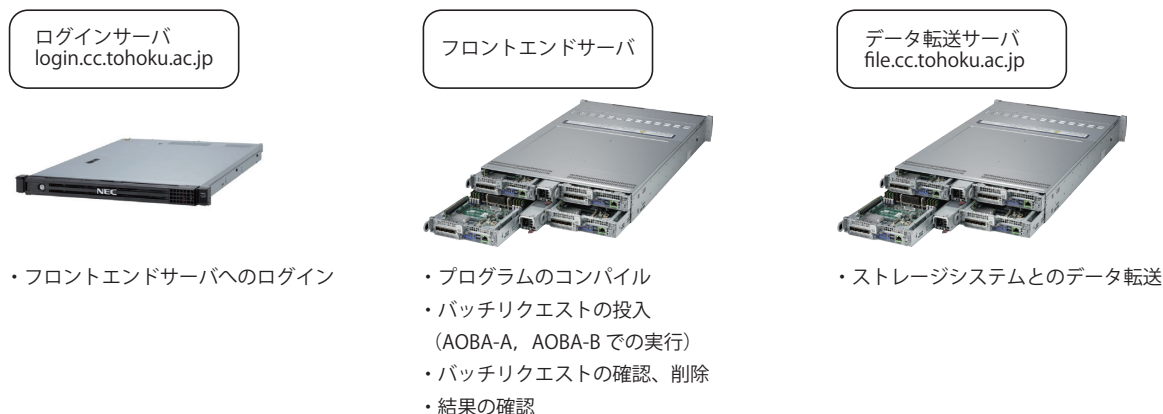


図3 ログインサーバ、フロントエンドサーバ、データ転送サーバ

鍵の作成からログインサーバを経由してフロントエンドサーバへログインする方法については、利用申請からログインまで (<https://www.ss.cc.tohoku.ac.jp/first-use/>) をご参照ください。

データ転送サーバの利用方法や、ストレージ容量の追加方法などについては、「データ転送 (ストレージ)」 (<https://www.ss.cc.tohoku.ac.jp/storage/>) をご参照ください。

3.2 プログラムの作成とコンパイル

3.2.1 ソースファイルの作成

フロントエンドサーバ上でソースファイルを作成するためのエディタは、emacs または vim が一般的です。研究室等のサーバやパソコンからソースファイルを転送する場合は、SSH 対応のファイル転送ソフトや scp コマンドで利用者のホームディレクトリに転送してください。Windows 環境で作成したソースコードの改行コードと文字コードを Linux 用にするためには、転送した後のソースコードに nkf コマンドを利用します (リスト 1)。

リスト 1 nkf コマンドの使用例

```
改行コードを LF, 文字コードを UTF-8 に変換し, 別のファイルに書き出す方法
front$ nkf -Lu 変換前ソースファイル名 > 変換後ソースファイル名

改行コードを LF, 文字コードを UTF-8 に変換してソースファイルを上書き保存する方法
front$ nkf --overwrite -Lu ソースファイル名
```

3.2.2 利用可能なコンパイラの種類

AOCC コンパイラとして、以下の Fortran および C/C++ コンパイラを利用できます。

AOCC

- flang : Fortran コンパイラ
- mpifort : MPI プログラム用 Fortran コンパイラ
- clang : C コンパイラ
- mpicc : MPI プログラム用 C コンパイラ
- clang++ : C++ コンパイラ
- mpic++ : MPI プログラム用 C++ コンパイラ

3.2.3 コンパイルを行う

ソースファイルはそれぞれの言語用コマンドでコンパイルします。単一コアで実行する逐次実行、OpenMP による共有メモリ並列実行および MPI ライブラリによる分散メモリ並列実行のコンパイル手順について解説します。

■Fortran プログラム ソースコードは flang コマンドまたは mpifort コマンドでコンパイルします。利用したい機能があれば、表 2 に示したようなコンパイルオプションを同時に指定します。その他のオプションについてはマニュアル (4 章) をご参照ください。

ソースファイルの拡張子は、自由形式 (フリーフォーマット) なら.f90 .f95 か.F90 .F95 を、固定形式 (7カラム目から記述) なら.f か.F を、2003 形式であれば.f03 か.F03 を付けます。(拡張子が大文字で始まるものは、コンパイルの前に cpp によるプリプロセス処理が行われます。)

コンパイルが成功すると、標準ではカレントディレクトリに実行オブジェクト a.out が作成されます。

表 2 Fortran,C,C++ コンパイラ的主要なオプション

コンパイルオプション	機能
-march=znver2	EPYC CPU (Rome) 用にコンパイルすることを指定する
-On	最適化レベルを指定する
n に指定できる値	
fast	最大限の最適化を適用する
3	積極的に最適化を適用する
2	(既定値) 標準的な最適化を適用する
1	最低限の最適化を適用する
0	すべての最適化を無効化する
-fopenmp	OpenMP 並列化を利用する

最適化レベルを高くすると、最適化の副作用により計算結果が変わることがありますのでご注意ください。

リスト 2 flang コマンドの使用例

```
(逐次実行)
front$ flang コンパイルオプション Fortranソースファイル名

(OpenMP並列実行)
front$ flang -fopenmp コンパイルオプション Fortranソースファイル名
```

リスト 3 mpifort コマンドの使用例

```
(MPI並列実行)
front$ mpifort コンパイルオプション Fortranソースファイル名
```

```
(MPIとOpenMPの同時並列実行)
front$ mpifort -fopenmp コンパイルオプション Fortranソースファイル名
```

■C/C++ プログラム ソースコードは clang または mpicc コマンドで C プログラムを, clang++ または mpic++ コマンドで C++ プログラムをコンパイルします。利用したい機能があれば Fortran コンパイラと同じく, 表 2 に示したようなコンパイルオプションを同時に指定します。その他のオプションについてはマニュアル (4 章) をご参照ください。

ソースファイルの拡張子は, C 言語であれば.c を, C++ であれば.C .cc .cpp .cp .cxx .c++ のいずれかを付けます。コンパイルが成功すると, カレントディレクトリに実行オブジェクト a.out が作成されます。

リスト 4 clang/clang++ コマンドの使用例

```
(逐次実行)
front$ clang コンパイルオプション Cソースファイル名
front$ clang++ コンパイルオプション C++ソースファイル名

(OpenMP並列実行)
front$ clang -fopenmp コンパイルオプション Cソースファイル名
front$ clang++ -fopenmp コンパイルオプション C++ソースファイル名
```

リスト 5 mpicc/mpic++ コマンドの使用例

```
(MPI並列実行)
front$ mpicc コンパイルオプション Cソースファイル名
front$ mpic++ コンパイルオプション C++ソースファイル名

(MPIとOpenMPの同時並列実行)
front$ mpicc -fopenmp コンパイルオプション Cソースファイル名
front$ mpic++ -fopenmp コンパイルオプション C++ソースファイル名
```

3.3 プログラムの実行

フロントエンドサーバでコンパイル作業を行って作成したプログラム (ジョブ) の実行は, バッチリクエストと呼ばれる方法で計算機に実行を依頼します。本センターではバッチリクエストの処理に NEC Network Queuing System V (以下, NQSV) を採用しています。図 4 にバッチリクエストの概念図を示します。

詳しくは「ジョブの実行方法」(<https://www.ss.cc.tohoku.ac.jp/nqs/>) をご参照ください。

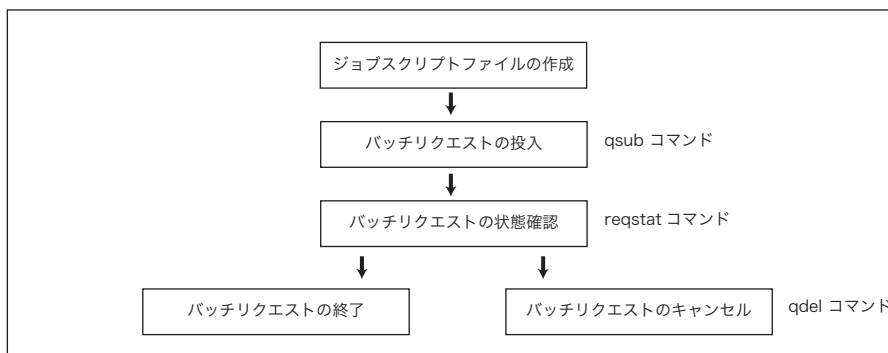


図 4 バッチリクエストの概念図

3.3.1 ジョブスクリプト

ジョブスクリプトに指定が必要となるのは,

- 投入キュー名
- 利用するノード数
- 最大経過時間
- 作業ディレクトリへの移動コマンド
- プログラムの実行コマンド

です。他に環境変数の指定、ファイルの操作コマンド等があれば適切な箇所に手続きを記述します。オプション、環境変数はジョブスクリプト中では#PBS の後に記述します。

サブシステム AOBA-B で実行する場合の、必須オプションを表 3 に示します。

表 3 サブシステム AOBA-B で実行する場合の必須オプション

オプション	指定するもの	機能
-q	lx	AOBA-B を利用することを指定
-b	1~16 の整数値	利用するノード数を指定
-l elapstim_req	最大経過時間を hh:mm:ss 形式で	計算機を利用する時間を指定

3.3.2 ジョブスクリプトの例

■**逐次実行の場合** リスト 6 に逐次実行の場合のジョブスクリプトの例を示します。逐次実行の場合、プログラムは 1CPU 中の 1 コアで実行されます。

リスト 6 ジョブスクリプトの例 (逐次実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q lx #AOBA-B を使用する
#PBS -b 1 #1ノードを使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を2時間に指定

cd $PBS_O_WORKDIR #qsubを実行したディレクトリに移動
./a.out #カレントディレクトリの a.out を実行
```

■**OpenMP 並列実行の場合** リスト 7 に OpenMP 並列実行の場合のジョブスクリプトの例を示します。OpenMP 並列実行の場合、プログラムは 1 ノード内の 2~128 コアで実行されます。実行コア数の指定は、環境変数 OMP_NUM_THREADS で行います。

リスト 7 ジョブスクリプトの例 (OpenMP 並列実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -q lx #AOBA-B を使用する
#PBS -b 1 #1ノードを使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を2時間に指定

export OMP_NUM_THREADS=128 #128コア並列で実行

cd $PBS_O_WORKDIR #qsubを実行したディレクトリに移動
./a.out #カレントディレクトリの a.out を実行
```

■**MPI 並列実行の場合** リスト 8 に MPI 並列実行の場合のジョブスクリプトの例を示します。MPI 並列実行の場合、mpirun コマンドでプログラムの実行を行います。mpirun コマンドの後に、-b オプションで指定した使用ノード数に応じて、バッチリクエストのノード割当てを自動的に設定するための環境変数\$NQSVMPIOPT を指定します。(逐次実行や OpenMP 並列実行用にコンパイルされたプログラムは MPI 並列実行を行っても、1 ノード内でしか実行されません。)

MPI プロセス数が確保したノードの物理コア数 (ノード数× 128) を超えると、演算性能が著しく低下しますのでご注意ください。

リスト 8 ジョブスクリプトの例 (MPI 並列実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -T openmpi # MPI実行環境を指定, AOCCではopenmpiを使用する
#PBS -q lx #AOBA-Bを使用する
#PBS -b 1 #1ノードを使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を2時間に指定

cd $PBS_O_WORKDIR #qsubを実行したディレクトリに移動
mpirun -np 128 $NQSVMPIOPTS ./a.out #カレントディレクトリの a.out を128プロセス並列で実行
```

■MPI, OpenMP 並列実行同時利用の場合 リスト 9 に MPI, OpenMP 並列実行同時利用の場合のジョブスクリプトの例を示します。

(MPI プロセス数) × (ノード内並列数) が確保したノードの物理コア数 (ノード数 × 128) を超えると、演算性能が著しく低下しますのでご注意ください。

リスト 9 ジョブスクリプトの例 (MPI, OpenMP 同時並列実行)

```
(run.sh の記述内容)
#!/bin/sh
#PBS -T openmpi # MPI実行環境を指定, AOCCではopenmpiを使用する
#PBS -q lx #AOBA-Bを使用する
#PBS -b 2 #2ノードを使う
#PBS -l elapstim_req=2:00:00 #最大経過時間を2時間に指定

export OMP_NUM_THREADS=64 #ノード内は64コア並列で実行

cd $PBS_O_WORKDIR #qsubを実行したディレクトリに移動
mpirun -np 2 $NQSVMPIOPTS ./a.out #カレントディレクトリの a.out を2プロセス×64コア並列で実行
```

4 マニュアル

4.1 コンパイラマニュアル

コンパイラについては以下のマニュアルをご参照下さい。

- AMD Optimizing C/C++ Compiler : <https://developer.amd.com/amd-aocc/>
- Intel Compiler : https://www.xlsoft.com/jp/products/intel/studio_xe/

4.2 数値計算ライブラリマニュアル

並列コンピュータ用数値計算ライブラリについては以下のマニュアルをご参照下さい。

- AMD Optimizing CPU Libraries (AOCL) : <https://developer.amd.com/amd-aocl/>
- Intel MKL : <https://www.xlsoft.com/jp/products/intel/perflib/mkl/>

5 おわりに

本稿では、スーパーコンピュータ AOBA のサブシステム AOBA-B のプログラミング利用ガイドとして基本的な手順を紹介しました。研究室のサーバでは実現できなかったプログラムやアイデアを、ぜひ最新鋭のスーパーコンピュータでお試ください。研究の強力なツールとしてご活用いただければ幸いです。

ご不明な点、ご質問等ございましたら、お気軽にセンターまでお問い合わせください。問い合わせについては「利用相談」(<https://www.ss.cc.tohoku.ac.jp/consultation/>)をご参照下さい。

[大規模科学計算システム] 【AOBA-A および AOBA-B の利用法】

ストレージシステムの利用法

情報部デジタルサービス支援課

1 はじめに

本稿では、スーパーコンピュータ AOBA のストレージシステムの利用法について紹介します。

1 つ目にストレージ環境にあるホームディレクトリと課題領域の容量確認方法について説明します。

2 つ目に実行した結果ファイル等のデータ (ストレージ環境) をローカル PC へ転送する方法およびローカル PC からストレージ環境へ転送する方法について説明します。

2 ストレージ環境

2.1 ホームディレクトリ (uhome)

プログラムファイル等を置く自分専用のホームディレクトリになり、ScaTeFS マウントし、AOBA-A と AOBA-B の両方に共有しています。

ディレクトリ名： /uhome/利用者番号

クォータ (容量) 制限： 5TB

クォータ制限を超過した場合、新規の書き込みができなくなりますのでご注意ください。クォータ制限を下回るように容量を削減すれば再度書き込みが可能になります。

- ホームディレクトリの容量確認コマンド

```
front$ uquota
```

表示例

```
Disk quotas for user 利用者番号
```

Filesystem	used(KB)	quota(TB)
/uhome/利用者番号	4	5

ホームディレクトリの容量追加申請については、2.3 章をご参照ください。

2.2 課題領域 (/short/プロジェクトコード)

課題領域は事前申請となり、同一プロジェクトコードの利用者間で大規模なデータ容量を利用される領域になります。申請を希望される際は、ストレージ資源の兼ね合いもありますので、「利用相談」(<https://www.ss.cc.tohoku.ac.jp/consultation/>) から事前にご連絡をお願いします。

容量については、申請されたディスク容量 (quota 値) になります。ホームディレクトリ同様に AOBA-A と AOBA-B の両方に共有しています。

課題領域の容量については、以下のコマンドの quota(TB) 部分をご確認をお願いします。

- 課題領域の容量確認コマンド

```
front$ uquota -A プロジェクトコード
```

表示例

```
Disk quotas for project プロジェクトコード
Filesystem          used(KB)    quota(TB)
/short/プロジェクトコード    10         20
```

- 利用方法

- 同一プロジェクトコードの利用者間でデータを共有する。
- 対象ディレクトリ内で利用者がそれぞれの専用サブディレクトリ (パーミッション：700) を作成し使用する。

利用方法の一例

プロジェクトコード：xx200001 の場合

コマンド例①) プロジェクトコードの利用者間で share を作成しデータを共有

```
front$ cp ホームディレクトリデータ /short/xx200001/share/
```

コマンド例②) 利用者の専用サブディレクトリを作成し使用

```
front$ mkdir /short/xx200001/利用者番号
```

```
front$ chmod 700 /short/xx200001/利用者番号
```

```
front$ cp ホームディレクトリデータ /short/xx200001/利用者番号/
```

【留意事項】

- ファイル同期コマンド (rsync コマンド)、コピーコマンド (cp コマンド) を使用する際は、グループ権限を保持するオプションは設定せずにご利用ください。
オプションを付けた場合、同一プロジェクトコード間のグループによる容量制限で正しく管理できなくなる恐れがあります。
移動コマンド (mv コマンド) によるファイルの移動を行った場合、元のファイルのグループ権限が保持されてしまいますので、rsync コマンド、cp コマンドを利用するようにしてください。
- 課題領域を当年度までのご利用の際、翌年度はデータ保管を行っていません。猶予期間後、対象課題領域を削除しますので、ローカル PC のディスクへ移行を速やかに進めてください。

2.3 ストレージ申請

ファイル容量の追加は 1TB 単位から申請可能です。ホームディレクトリ、課題領域ともに利用負担金が発生しますので、詳しくは「利用負担金」(<https://www.ss.cc.tohoku.ac.jp/charge/>) をご参照ください。

申請用紙は、「ストレージ容量申請書」(<https://www.ss.cc.tohoku.ac.jp/application-form/>) を使い申請をお願いします。

3 データ転送方法

データ転送サーバへログインし、SSH による暗号化を行う scp(Secure CoPy), SFTP(Ssh File Transfer Protocol) を利用します。接続方法については「利用申請からログインまで」(<https://www.ss.cc.tohoku.ac.jp/first-use/>) をご参照ください。

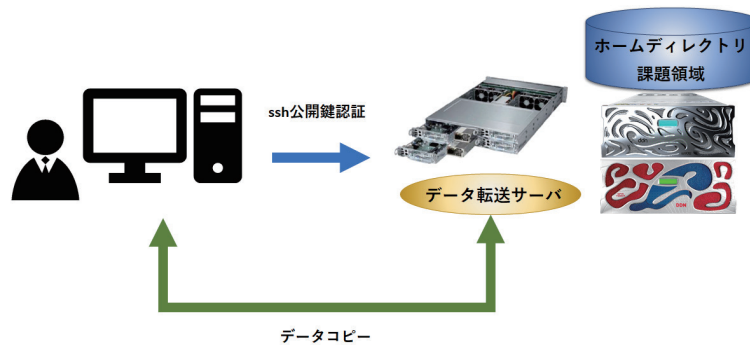


図1 アクセスイメージ

3.1 Powershell(Windows)・MAC・Linux

標準で SSH クライアントがインストールされています。インストールされている各 Terminal ソフトでデータ転送サーバへログインします。

• scp コマンド

SSH 利用し、ネットワーク・ホスト間でファイルを安全にコピーするためのコマンドです。

ローカル PC からリモート (ストレージ環境) に転送

\$ scp -i 秘密鍵ファイル ローカル PC のファイル名 利用者番号@file.cc.tohoku.ac.jp:リモートの保存先パス

scp コマンドの実行例

(ローカル PC 上にある sample.txt ファイルをホームディレクトリへ転送)

\$ scp -i ~/.ssh/id_rsa_cc sample.txt 利用者番号@file.cc.tohoku.ac.jp:sample.txt

パスフレーズを聞かれますので入力します。

リモート (ストレージ環境) からローカル PC に転送

\$ scp -i 秘密鍵ファイル 利用者番号@file.cc.tohoku.ac.jp:ファイル名 ローカル PC の保存先パス

scp コマンドの実行例

(ホームディレクトリ上にある sample.txt をローカル PC へ転送)

\$ scp -i ~/.ssh/id_rsa_cc 利用者番号@file.cc.tohoku.ac.jp:sample.txt ./

詳しい用例については man コマンド等を利用し、scp コマンドのマニュアル閲覧をお願いします。

\$ man scp

- **sftp コマンド**

SSH 利用し、対話的なファイル転送を行うことができるコマンドです。

ローカル PC からリモート (ストレージ環境) に転送

```
$ sftp -i 秘密鍵ファイル 利用者番号@file.cc.tohoku.ac.jp
```

パスフレーズを聞かれますので入力します。

sftp> と表示されたら成功です。

```
sftp> put ファイル名 保存先フォルダ
```

sftp コマンドの実行例

(ローカル PC 上にある sample.txt ファイルをホームディレクトリへ転送)

```
sftp> put パス名/sample.txt ./
```

リモート (ストレージ環境) からローカル PC に転送

```
$ sftp -i 秘密鍵ファイル 利用者番号@file.cc.tohoku.ac.jp
```

パスフレーズを聞かれますので入力します。

sftp> と表示されたら成功です。

```
sftp> get ファイル名 保存先フォルダ
```

sftp コマンドの実行例

(ホームディレクトリ上にある sample.txt をローカル PC へ転送)

```
sftp> get ./sample.txt ./
```

詳しい用例については man コマンド等を利用し、sftp コマンドのマニュアル閲覧をお願いします。

```
$ man sftp
```

3.2 WinSCP(Windows ソフト)

標準で scp, sftp に対応したソフトウェアがインストールされていないため、はじめにインストールする必要があります。

ここでは、代表的なソフトウェアである WinSCP を利用したファイル転送方法を説明します。

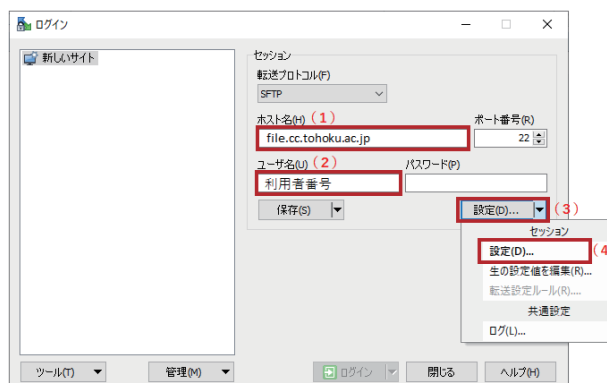


図 2 WinSCP 設定画面

1. WinSCP を起動します。
2. ホスト名（上図（1））に file.cc.tohoku.ac.jp と入力します。
3. ユーザ名（上図（2））に利用者番号を入力します。
4. 設定（上図（3））のプルダウンメニューから設定（上図（4））をクリックします。

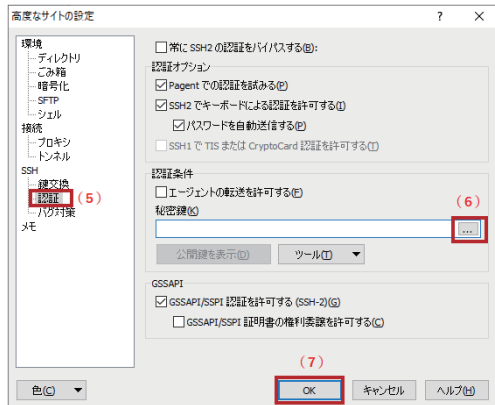


図 3 WinSCP 鍵認証設定画面

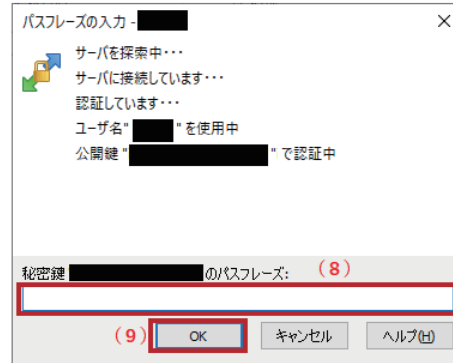


図 4 WinSCP 鍵認証画面

5. 左側ナビゲーションメニューの認証（上図（5））を選択します。
6. 秘密鍵のプルダウンメニュー（上図（6））をクリックし、ログインに使用する秘密鍵を指定します。秘密鍵 (.ppk ファイル) を未生成の場合、3.2.1 章をご参照ください。
7. OK（上図（7））をクリックし、鍵の設定を保存します。
8. 元の画面に遷移しますので、ログインをクリックしてください。
9. パスフレーズの入力画面が出ますので、パスフレーズを入力（上図（8））した上で、OK（上図（9））をクリックしてください。成功しますと WinSCP の画面が表示されファイル転送が可能になります。

3.2.1 WinSCP 用の鍵生成手順

1. WinSCP を起動した後、「ツール」をクリックし、「PuTTYgen を実行」を選択します。
2. PuTTYgen を起動すると、「PuTTY Key Generator」ダイアログボックスが表示されますので、「Load」をクリックします。

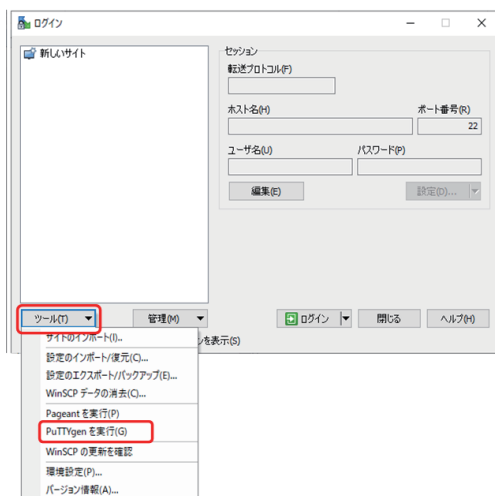


図 5 WinSCP 初期設定画面

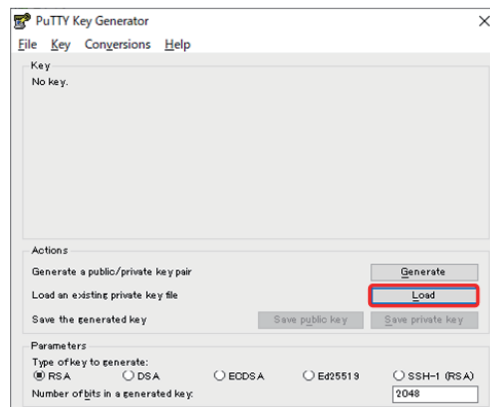


図 6 PuTTY Key Generator 画面

3. ファイルの選択画面が表示されますので、ポータルサイトで作成した秘密鍵「id_rsa_cc」を選択すると、パスフレーズの入力を求められます。
4. パスフレーズの内容が一致すると Notice(情報) メッセージが表示されるので、OK をクリックします。
5. 「Save private key」をクリックし、ファイル名を設定します。
6. 設定が完了しましたら、「PuTTY Key Generator」の画面は閉じてください。

4 おわりに

本稿では、ストレージシステムの利用法を紹介しました。ご不明な点、ご質問等ございましたら、お気軽にセンターまでお問い合わせください。問い合わせ先については「利用相談」(<https://www.ss.cc.tohoku.ac.jp/consultation/>)をご参照下さい。

[大規模科学計算システム] 【A0BA-A および A0BA-B の利用法】

大判カラープリンタの利用法

情報部デジタルサービス支援課

2020年10月1日 初版
2025年3月10日 第6版

1 はじめに

センターでは2020年10月から新大判カラープリンタシステム（ヒューレットパッカード社、HP DesignJet Z6 PS dr V-Trimmer）を導入しました。

光沢紙（600円/枚）またはソフトクロス紙（1,200円/枚）に、A0サイズまでの印刷が可能です。従来の大判カラープリンタでは印刷後に左右の余白をカットする必要がありましたが、今回導入した大判カラープリンタには自動的に左右の余白をカットする機能 V-Trimmer が搭載されており、余白のカットの手間がなくなりました。

大判カラープリンタは、サイバーサイエンスセンター本館1階の利用相談室にあります。利用時間は平日9時～21時です。大判カラープリンタを利用するためには、利用負担金が発生しますのでサイバーサイエンスセンターの利用者番号が必要です。

本稿ではセンター利用相談室の Windows 端末から PowerPoint や PDF などのアプリケーションデータを大判カラープリンタで印刷する方法について説明します。



図1 大判カラープリンタシステムの構成 (2020年10月1日～)

2 利用負担金

光沢紙への印刷は、1枚（1部）あたり600円、ソフトクロス紙への印刷は、1枚（1部）あたり1,200円の利用負担金が発生します。いずれの利用負担金も1枚（1部）あたりの金額ですので、出力サイズの大小によりません。

3 印刷に必要なもの

■ **サイバーサイエンスセンターの利用者番号** センターに利用登録をしていない場合は、利用申請が必要です。利用者番号の取得までは平日3日間程度かかります。日数に余裕を持って利用申請を行ってください。

「利用申請からログインまで」 <https://www.ss.cc.tohoku.ac.jp/first-use/>

■ **印刷データ** 利用者のPC/Mac等で作成したポスター用に作成したデータ（パワーポイント、PDF）をUSBメモリ等に入れてお持ちください。PDFを作成する場合は、フォントを全て埋め込んでください。Macで作成したパワーポイントのファイルや、Googleスライドで作成したPDFファイルは、フォント不足やレイアウトが崩れることがあります。

4 印刷手順

4.1 Windows 端末へのログイン

利用相談室の3台のWindows端末から印刷が可能です（図2）。

サイバーサイエンスセンターの利用者番号と、LDAPパスワード（初期パスワードは利用承認書に記載のもの）でWindowsにログインしてください（図3）。

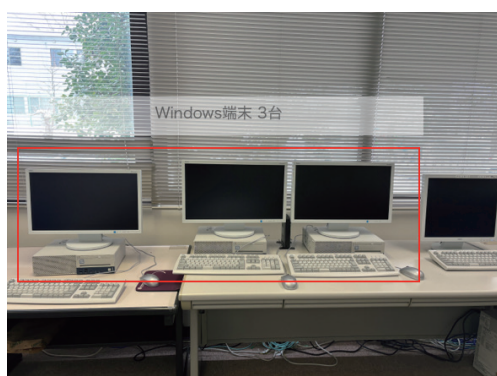


図2 Windows 端末

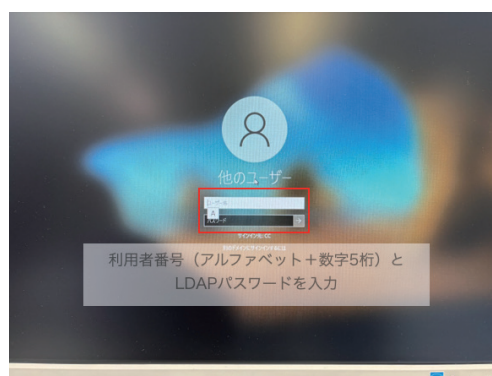


図3 ログイン画面

4.2 印刷データを開く

Windows 端末のUSBポートにUSBメモリ等を接続し、印刷データを開きます。各アプリケーションの印刷メニューを選択してください。PDFファイルから印刷する場合は、Adobe Acrobat ReaderもしくはMicrosoft Edge Browserが利用できます。Googleスライドで作成したPDFファイルはレイアウト崩れが発生するため、Microsoft Edge Browserから印刷を行ってください。

4.3 プリンタの選択

1. 使用するプリンタを選択します（図4）。「Z6dr 44in Left」と「Z6dr 44in Right」のどちらのプリンタを選択しても、光沢紙とソフトクロス紙での印刷が可能です。
2. 印刷する用紙サイズ等を設定するために、「プロパティ (P)」を選択します。

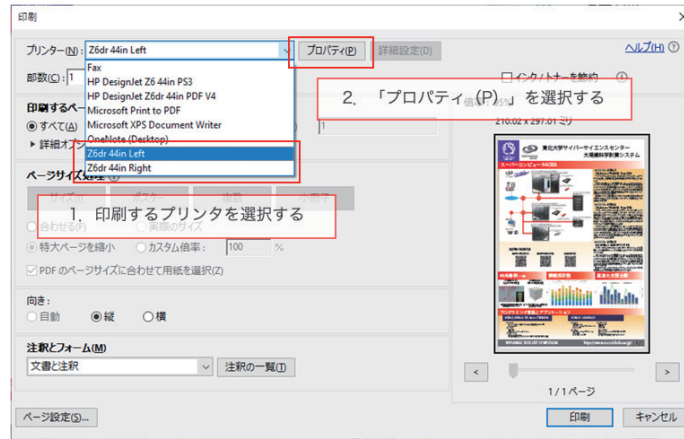


図4 印刷メニュー

4.4 印刷用紙の選択

それぞれのプリンタには2種類の印刷用紙が取り付けられています。印刷する用紙に合わせて給紙方法を選択してください（図5）。

- 光沢紙に印刷する場合 「ロール紙1」を選択
- ソフトクロス紙に印刷する場合 「ロール紙2」を選択

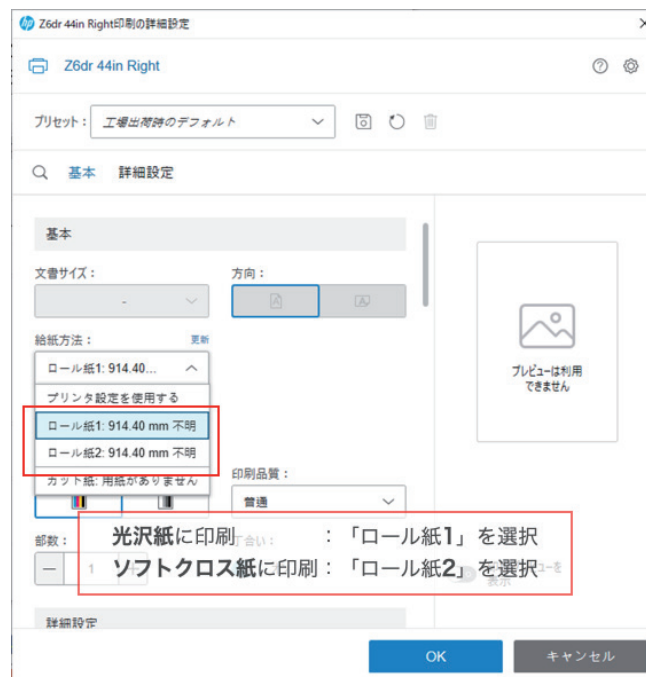


図5 印刷用紙の選択

4.5 印刷品質の変更

必要な場合は印刷品質を変更してください（図6）。

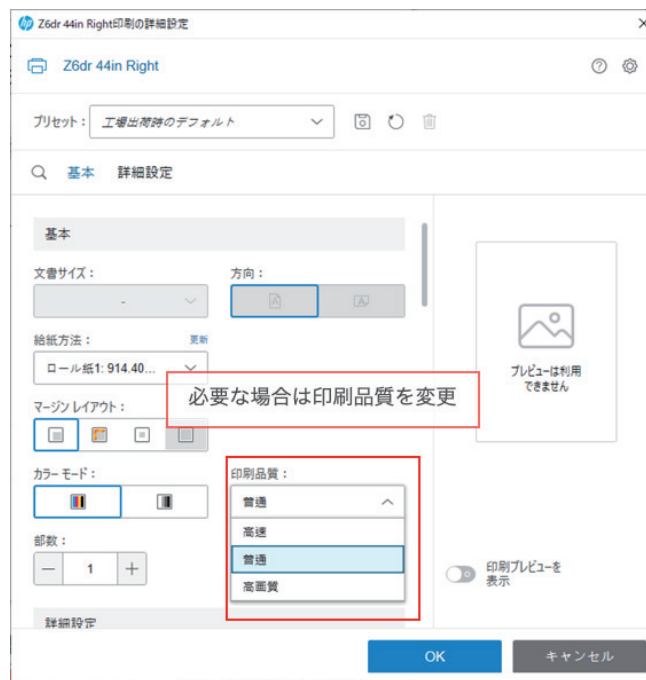


図6 印刷品質の変更

4.6 自動カッター機能を有効にする

垂直トリマー（縦方向の自動カッター機能）をオンにしてください（図7）。

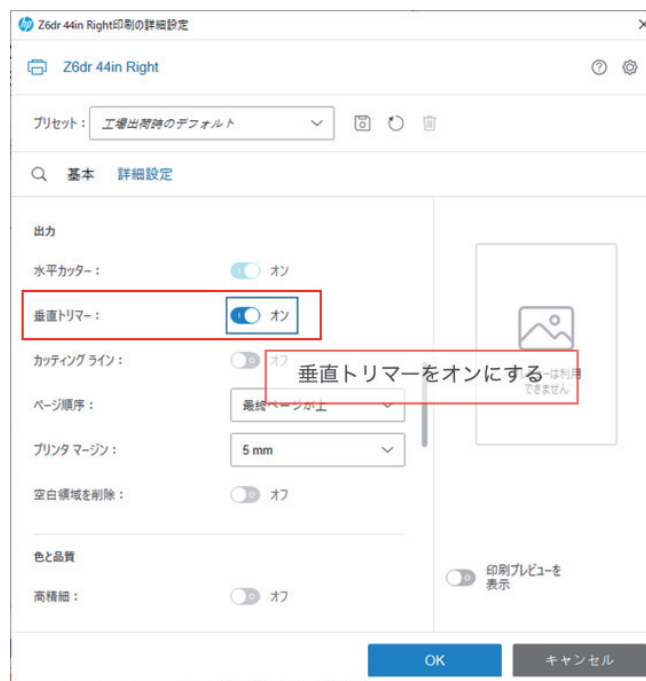


図7 自動カッター機能

4.7 出力サイズを選択

「スケール」で出力サイズを選択してください。A0 サイズまで出力可能です。プリンタ名と出力サイズ、その他設定を確認後、「OK」を押してください（図8）。

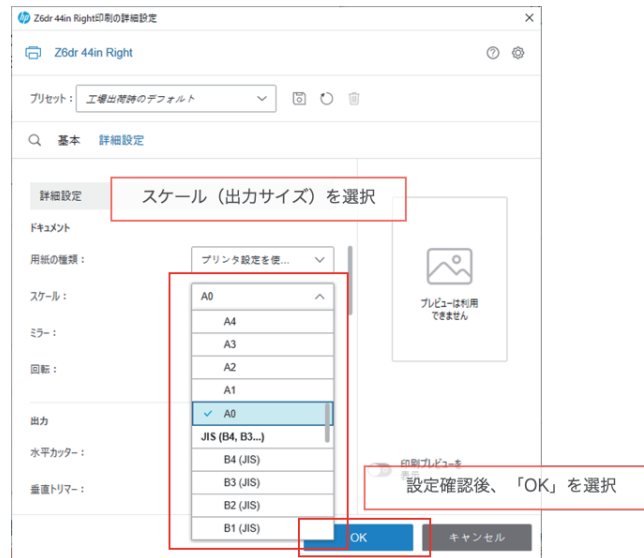


図 8 出力サイズを選択

4.8 印刷開始

「印刷」を選択すると印刷を開始します（図9）。

光沢紙への印刷の場合乾燥時間が必要なため、印刷完了後にプリンタが数分停止します。印刷が完了すると自動的に切断されますので、絶対に用紙を引っ張らないでください。プリンタ故障の原因になります。



図 9 印刷メニュー

5 その他

- 印刷で出た用紙の切れ端は、利用相談室内のゴミ箱に捨ててください。
- 印刷用紙が無くなった場合は、利用者での用紙交換が可能です。利用相談室内のロール紙をご利用ください。
- 印刷終了後は Windows 端末をシャットダウンしてください。
- 使用した USB メモリ等は忘れずにお持ち帰りください。
- 最後の退出者の方は、電灯とエアコンの電源を OFF にしてください。
- 利用について問い合わせがある場合は、利用相談室の電話機から内線 832 または 829 におかけください。担当者不在の場合は当日の対応が出来ませんのでご了承ください。

6 おわりに

今回センターに導入した大判カラープリンタは V-Trimmer 機能により、余白のカットの手間がなくなりました。プリンターで印刷可能なソフトクロス紙は、織り目がつきにくく持ち運びに非常に便利です。また、光沢紙は色鮮やかな印刷が可能です。研究成果のポスター印刷にぜひご活用ください。

[お知らせ]

2026 年度 サイバーサイエンスセンター講習会計画

No.	講習会名 開催方式	開催日時	募集 人員	内 容
1	はじめての Linux	5月26日(火) 13:00-15:00	15	<ul style="list-style-type: none"> Linux システムの基本的な使い方 エディタの使い方
	オンライン			
2	はじめてのスパコン	5月28日(木) 13:00-15:00	15	<ul style="list-style-type: none"> スーパーコンピュータの紹介と 利用法入門
	オンライン			
3	ネットワークとセキュ リティ入門	8月4日(火) 13:30-15:30	上限なし	<ul style="list-style-type: none"> ネットワークの基本的な仕組 み ネットワークの危険性と安全 対策
	オンライン			
4	はじめての Linux	9月10日(木) 13:00-15:00	15	<ul style="list-style-type: none"> Linux システムの基本的な 使い方 エディタの使い方
	オンライン			
5	並列プログラミング 入門 I (OpenMP)	9月11日(金) 13:00-16:00	15	<ul style="list-style-type: none"> 並列プログラミングの概要 OpenMP による並列プログラミ ングの基礎 利用法
	オンライン			
6	並列プログラミング 入門 II (MPI)	9月中旬 13:00-16:00	15	<ul style="list-style-type: none"> MPI による並列プログラミングの 基礎 利用法
	オンライン			
7	Gaussian 入門	9月30(水) 13:00-16:30	15	<ul style="list-style-type: none"> Gaussian の基本的な使い方
	対面			
8	はじめての Fortran	10月2日(金) 13:00-16:00	15	<ul style="list-style-type: none"> Fortran の入門編
	オンライン			
9	SX-Aurora TSUBASA の 性能分析・高速化	10月中旬 13:00-16:00	15	<ul style="list-style-type: none"> スーパーコンピュータでの性 能 解析から最適化まで
	ハイブリッド			
10	AOBA-B における性能 分析と最適化	10月下旬 13:00-16:00	15	(調整中)
	オンライン			
11	NLCPy(NumPy 互換ライブ ラリ)の利用法	10月下旬 13:00-14:15	15	<ul style="list-style-type: none"> NLCPy(NumPy 互換ライブラリ) を使用した Python スクリプト の SX-Aurora TSUBASA での高 速化
	オンライン			

[報 告]

2025 年度 サイバーサイエンスセンター講習会報告

No.	講習会名 開催方式	開催日時	受講者数 () 内はオンライン参加者数	内 容
1	はじめての Linux	5 月 21 日 (水) 13:00-15:00	5 (5)	<ul style="list-style-type: none"> Linux システムの基本的な使い方 エディタの使い方
	オンライン			
2	はじめてのスパコン	5 月 23 日 (金) 13:00-15:00	9 (9)	<ul style="list-style-type: none"> スーパーコンピュータの紹介と利用法入門
	オンライン			
3	ネットワークとセキュリティ入門	8 月 4 日 (月) 13:30-15:30	23 (23)	<ul style="list-style-type: none"> ネットワークの基本的な仕組み ネットワークの危険性と安全対策
	オンライン			
4	はじめての Linux	9 月 9 日 (火) 13:00-15:00	8 (8)	<ul style="list-style-type: none"> Linux システムの基本的な使い方 エディタの使い方
	オンライン			
5	並列プログラミング入門 I (OpenMP)	9 月 11 日 (木) 13:00-16:00	7 (7)	<ul style="list-style-type: none"> 並列プログラミングの概要 OpenMP による並列プログラミングの基礎 利用法
	オンライン			
6	並列プログラミング入門 II (MPI)	9 月 16 日 (火) 13:00-16:00	5 (5)	<ul style="list-style-type: none"> MPI による並列プログラミングの基礎 利用法
	オンライン			
7	Gaussian 入門	9 月 25 日 (木) 13:00-16:30	4	<ul style="list-style-type: none"> Gaussian の基本的な使い方
	対面			
8	はじめての Fortran	9 月 30 日 (火) 13:00-16:00	2 (2)	<ul style="list-style-type: none"> Fortran の入門編
	オンライン			
9	SX-Aurora TSUBASA の性能分析・高速化	10 月 7 日 (火) 13:00-16:00	中止	<ul style="list-style-type: none"> スーパーコンピュータでの性能解析から最適化まで
	ハイブリッド			
10	AOBA-B における最適化技法	11 月 12 日 (水) 13:00-15:00	6 (6)	<ul style="list-style-type: none"> Intel コンパイラの概要説明 Intel コンパイラでの最適化技法
	オンライン			
11	スーパーコンピュータ AOBA を使った高速な統計的機械学習	11 月 22 日 (金) 13:00-15:00	5 (5)	<ul style="list-style-type: none"> 機械学習の概要説明 Frovedis の紹介と AOBA での Frovedis の使い方
	オンライン			
12	NLCPy の利用法	11 月 25 日 (火) 13:00-14:45	5 (5)	<ul style="list-style-type: none"> NLCPy の概要説明 AOBA での利用方法
	オンライン			
参加者合計			79 (75)	

[スーパーコンピュータ AOBA のお知らせより]

東北大学サイバーサイエンスセンター大規模科学計算システムウェブサイトに掲載されたお知らせの一部を転載しています。

<https://www.ss.cc.tohoku.ac.jp/information/>

コンパイラのバージョンアップについて

2026年4月2日にAOBA-S・AOBA-A・AOBA-Bのコンパイラをバージョンアップいたします。

システム	コンパイラ名	旧バージョン	新バージョン	ドキュメント
AOBA-S	Fortran Compiler	5.4.0	5.4.1	マニュアル&リリースノート
	C/C++ Compiler	5.4.0	5.4.1	
AOBA-A	MPI※1	3.8.0	3.10.0	
	Intel Compiler (VH用プログラム) ※2	oneAPI 2025.2.0	oneAPI 2025.3.1	oneAPI マニュアル関連
AOBA-B	Intel Compiler ※2			

※1 MPI を利用するプログラムは再コンパイルが必要

※2 Intel oneAPI 2025.3.1 の環境変数設定ファイルは、bash 向けのみの提供

(情報部デジタルサービス支援課)

商用アプリケーションのバージョンアップについて

数式処理システム「Mathematica」のバージョンアップを行いましたのでお知らせいたします。

新機能の概要、機能の詳細については開発元 Web サイトをご参照ください。

Mathematica

- バージョン : 14.3
- バージョンアップ日 : 2026年4月2日
- サービスホスト : フロントエンドサーバ
- 起動コマンド : (GUI版) mathematica (コマンドライン版) math
- 開発元 Web サイト : <https://www.wolfram.com/mathematica/new-in-14/>

(情報部デジタルサービス支援課)

2026 年度共同研究課題採択結果(第 1 回)について

本センターでは、大規模科学計算システムの利用者と共同でプログラムやアルゴリズムを開発する共同研究を行っています。2026 年度の第 1 回募集に応募されたものについて共同研究専門部会で審査の結果、以下の 12 件が採択されましたのでお知らせします。

[A] 若手・女性研究者支援課題

No.	申請者	所属	研究課題
A-1	平賀 優介	東北大学 大学院工学研究科	東北地方における線状降水帯の高精度予測モデルの開発
A-2	大野 詩歩	東北医科薬科大学 分子生体膜研究所	部位特異的な糖鎖修飾機構の解明に向けた糖タンパク質のダイナミクス研究
A-3	笹本 大空	東北大学 大学院薬学研究科	密度汎関数理論に基づいた有機超塩基が示す特異な反応性の理論的考察
A-4	松川 嘉也	東北大学 大学院工学研究科	微粒化過程における帯電・導電挙動に関する直接数値シミュレーション
A-5	屋山 巴	工学院大学 先進工学部	TDDFT によるフッ素終端した窒化ガリウムにおけるホールダイナミクスの解明
A-6	小野 梓	東北大学 大学院工学研究科	都市形態が積乱雲発生・発達に及ぼす影響に関する研究

[B] 萌芽型課題

No.	申請者	所属	研究課題
B-1	松岡 浩	技術士事務所 AI コンピューティングラボ	リカレント型ビット演算による流体・構造体統一解析手法の開発
B-2	佐々木 大輔	大阪公立大学 大学院工学研究科	適合格子細分化を伴う直交格子法による大規模非定常流体解析手法の高度化

[C] 一般課題

No.	申請者	所属	研究課題
C-1	平田 晃正	名古屋工業大学 大学院工学研究科	暑熱環境と電波・太陽光の複合ばく露に対する温熱評価
C-2	立川 智章	東京理科大学 工学部	データ駆動型手法と高忠実流体シミュレーションの融合による動的流れ制御とその根拠可視化
C-3	伊藤 純至	東北大学 大学院理学研究科	次期日本域長期領域再解析システムの開発
C-4	塚原 隆裕	東京理科大学 創域理工学部	Reynolds 数低下過程における局在乱流パターン形成の大規模直接数値計算

(スーパーコンピューティング研究部, 情報部デジタルサービス支援課)

計算科学・計算機科学人材育成のためのスーパーコンピュータ無償提供制度について

東北大学サイバーサイエンスセンターでは、計算科学・計算機科学分野での教育貢献・人材育成を目的として、無料で大規模科学計算システムを利用できる制度を用意しております。提供の対象は、大学院・学部での講義実習等の教育目的(卒業論文、修士論文、博士論文での利用を除く)に限ります。利用を希望される場合は以下の情報を添えて、講義開始の2週間前までに cc-edu-prog[at]cc.tohoku.ac.jp 宛お申し込みください。

- ・ 講義担当者氏名
- ・ 同所属
- ・ 同連絡先 (住所, 電話, 電子メール)
- ・ 講義名
- ・ 講義実施日時 (1セメスターの中で実習を予定している回数)
- ・ センター端末機室等での実習利用希望の有無 (必要であれば予定日時)
- ・ 講師派遣の希望の有無
- ・ 講義シラバス
- ・ 講義ウェブ (もし用意されていれば)
- ・ 受講者数 (予定)
- ・ 必要とする理由 (利用目的: 例えば、数値シミュレーションの研修を行うなど)
- ・ 期待できる教育効果
- ・ 居住性チェックリストの提出 (受講者に外国人が居る場合)
参照: <https://www.ss.cc.tohoku.ac.jp/apply-for-use/#toc3>
- ・ その他 (センターへの要望等)

なお、講義終了後、報告書 (広報誌 SENAC へ掲載) の提出をお願いいたします。

たくさんのお申し込みをお待ちしております。不明な点は、cc-edu-prog[at]grp.tohoku.ac.jp までお問い合わせください。

(スーパーコンピューティング研究部, 情報部デジタルサービス支援課)

大規模科学計算システムの機関 (部局) 単位での利用について

東北大学サイバーサイエンスセンターでは、大規模科学計算システムをご利用いただくにあたり、利用負担金を利用者単位のほか、機関 (部局) 単位で年間定額をお支払いいただくことで利用できるサービスも提供しております。このサービスは、機関 (部局) 単位でお申し込みいただくことにより、その構成員であれば、各研究室が個別に利用負担金を支払うことなく、下記システムを利用できる仕組みとなっております。

これまで計算機を利用する機会がなかった研究者による新たなニーズへの対応や研究室の計算機では実行できなかった大規模シミュレーションが実行可能であり、また自前で計算機を導入す

るためのコストや運用コストも削減可能です。すでにご利用いただいている機関（部局）からは、当初の予想を上回るご利用をいただき、ご好評をいただいております。

占有利用・共有利用については必要に応じて取り混ぜながら、ご予算に合わせて、年間定額により利用することが可能となっておりますので、ぜひご相談ください。

記

【利用可能なシステム】

- ・サブシステム AOBA-S
- ・サブシステム AOBA-A
- ・サブシステム AOBA-B
- ・ストレージシステム
- ・大判カラープリンター（光沢紙、ソフトクロス紙）

【問い合わせ先・申し込み先】

スーパーコンピューティングサポートユニット（cc-uketuke[at]grp.tohoku.ac.jp）

（情報部デジタルサービス支援課）

民間企業利用サービスについて

東北大学サイバーサイエンスセンターでは、社会貢献の一環として大学で開発された応用ソフトウェアとスーパーコンピュータを、民間企業の方が無償または有償にてご利用頂ける制度を用意しております。本サービスにおける利用課題区分は以下の2つとなります。

- ・大規模計算利用（有償利用）
- ・トライアルユース（無償利用）

詳細については以下を参照ください。

<https://www.ss.cc.tohoku.ac.jp/business/>

【問い合わせ先・申し込み先】

スーパーコンピューティングサポートユニット（cc-uketuke[at]grp.tohoku.ac.jp）

（情報部デジタルサービス支援課）

2026 年度利用負担金について

2026 年度の利用負担金についてお知らせします。詳細は以下をご覧ください。

<https://www.ss.cc.tohoku.ac.jp/charge/>

（情報部デジタルサービス支援課）

執筆要項

— SENAC 執筆要項 —

1. お寄せいただきたい投稿内容

サイバーサイエンスセンターでは、研究者・技術者・学生等の方々からの原稿を募集しております。以下の内容で募集しておりますので、皆さまのご投稿をお待ちしております。なお、一般投稿いただいた方には、謝礼として負担金の一部を免除いたします。

- ・一般利用者の方々が関心をもたれる事項に関する論説
- ・センターの計算機を利用して行った研究論文の概要
- ・プログラミングの実例と解説
- ・センターに対する意見、要望
- ・利用者相互の情報交換

2. 執筆にあたってご注意いただく事項

- (1)原稿は横書きです。
- (2)術語以外は、「常用漢字」を用い、かなは「現代かなづかい」を用いるものとします。
- (3)学術あるいは技術に関する原稿の場合、200字～400字程度のアブストラクトをつけてください。
- (4)参考文献は通し番号を付し末尾に一括記載し、本文中の該当箇所に引用番号を記入ください。
 - ・雑誌：著者, タイトル, 雑誌名, 巻, 号, ページ, 発行年
 - ・書籍：著者, 書名, ページ, 発行所, 発行年

3. 原稿の提出方法

原稿のファイル形式はWordを標準としますが、PDFでの提出も可能です。サイズ*は以下を参照してください。ファイルは電子メールで提出してください。

—用紙サイズ・文字サイズ等の目安—

- ・サイズ：A4
- ・余白：上=30mm 下=25mm 左右=25mm 綴じ代=0
- ・標準の文字数（45文字 47行）
- ・表題=ゴシック体 14pt 中央 ・副題=明朝体 12pt 中央
- ・氏名=明朝体 10.5pt 中央
- ・所属=明朝体 10.5pt 中央
- ・本文=明朝体 10.5pt
- ・章・見出し番号=ゴシック体 11pt～12pt

*余白サイズ、文字数、文字サイズは目安とお考えください。

4. その他

- (1)一般投稿を頂いた方には謝礼として、負担金の一部を免除いたします。免除額は概ね1ページ1万円を目安とします。詳細はデジタルサービス支援課（受付）までお問い合わせください。
- (2)投稿予定の原稿が15ページを越す場合はデジタルサービス支援課（受付）まで前もってご連絡ください。
- (3)初回の校正は、執筆者が行って、誤植の防止をはかるものとします。
- (4)原稿の提出先は次のとおりです。

東北大学サイバーサイエンスセンター内 情報部デジタルサービス支援課（受付）

e-mail cc-uketuke@grp.tohoku.ac.jp

TEL 022-795-3406

スタッフ便り

令和8年の新年度が始まりました。

例年、年度末の3月は、退職されたり異動されたりする教職員の皆さんとお別れをしなくてはならない季節なのですが、特に今年の3月は、これまでセンターに密接に関わってくださった多くの方々が去られ、例年よりもさらに寂しさが際立った年度末となりました。センター長を歴任され現在のセンターの礎を築かれた先生、昔からTAINSの運営にご協力いただいた先生、スパコンをフル活用して多くの研究成果を出していただいたヘビーユーザの先生、機構の関係でセンター運営にも貴重なアドバイスを頂いた先生、ネットワークやセキュリティ関係で先頭に立って引っ張ってくれた職員の方、事務を支えてくださったスタッフの皆さん。中心的な役割を担ってこられた方々が去られ、世代交代を強く感じたこの春。

また、このSENAC編集にも長年に渡って携わっていただいたスタッフのSさんも3月をもって退職されました。長い間、誠にありがとうございます。センター玄関の生け花も見られなくなって寂しいです。

4月になり、これまた例年になく多くの教職員の方々を迎え、センターも新たなスタートを切りました。去られた皆さまにおかれましては、これまで支えていただいた本センターをより大きく発展させるために、みなで頑張っていこうと思いますので、見守っていただければ幸いです。新しい生活でのご活躍とご健勝を祈念しております(T.S)。

この春、再雇用期間が終了し東北大学職員を卒業することとなりました。サイバーサイエンスセンター広報誌SENACの編集・発行作業に20年間関わらせていただきました。前任者の気の重い仕事だったという言葉通り、記事に関してはほとんどの執筆者がお会いしたことがない教員の方々に、緊張のやりとりでした。また、発行目標時期がきまっているのに思うように記事の準備が進まずの状態でした。当初は私にとっても気が重い仕事の一つでした。利用相談関係の仕事も担当していましたが、ある日利用相談室でテクニカルアシスタントの大学院生と雑談中にSENACの仕事の愚痴をこぼしたところ、「そんなこと言わないで日本一の広報誌作りましょうよ。」と励まされました。「えっ！日本一、ないないない。」と即答しましたが、その会話をきっかけに、自分ももう少しがんばってみようという作業を見直したりして、それ以降執筆者とのやりとりや、いただいた投稿原稿の内容にも少しですが、楽しさや興味を覚えることが増えました。決して立派に発行できたとは思っていませんが、なんとか20年間途切れることなく発行できたことにほっとしています。この間での大きな変化は、冊子とオンラインでの発行から、オンラインジャーナルのみの発行(2020年)になったことだと思います。巻頭言、研究成果、各種資料、受賞のコメントなど多くの全国の皆様に執筆いただきました。また編集部会のメンバーと印刷業者の担当者にはいつも暖かい気持ちでご協力いただきました。関わっていただいたすべての方々に深謝申し上げます。近年、国内の機関で発行される広報誌が、激減しているように感じています。当センターのSENACもいつまで発行され続けるかはわかりませんが、今後はウェブサイトたまに覗いて、スタッフ便りなどを密かに楽しみたいと思っています。(K.S)

【サイバーサイエンスセンター・情報部デジタルサービス支援課スタッフ 着任・採用のお知らせ】

2025. 4. 1 着任

小笠原 朋美 総務係 係長
川綱 美優 総務係 事務職員

2025. 4. 1 採用

垂石 ひなた 会計係 事務職員
西岡 保幸 総務係 事務補佐員
JIN YIFAN スーパーコンピューティングサポートユニット 情報系職員

SENAC 編集部会

滝沢 寛之 水木 敬明 後藤 英昭 河合 直聡
小笠原 朋美 佐々木明里 大泉 健治 小野 敏
奥島 忍

	2026 年 4 月 発行
編集・発行	東北大学 サイバーサイエンスセンター 仙台市青葉区荒巻字青葉 6-3 郵便番号 980-8578
PDF 作成	株式会社 東誠社

スーパーコンピュータ AOBAS システム一覧

計算機システム	機種
サブシステム AOBAS	SX-Aurora TSUBASA Type 30A
サブシステム AOBAA	SX-Aurora TSUBASA Type 20B
サブシステム AOBAB	LX 406Rz-2

サーバとホスト名

フロントエンドサーバ (AOBAS 用)	sfront.cc.tohoku.ac.jp
データ転送サーバ (AOBAS 用)	sfile.cc.tohoku.ac.jp
ログインサーバ (AOBAA, B 用)	login.cc.tohoku.ac.jp
データ転送サーバ (AOBAA, B 用)	file.cc.tohoku.ac.jp

サービス時間

利用システム名等	利用時間帯
サブシステム AOBAS	連続運転
サブシステム AOBAA	連続運転
サブシステム AOBAB	連続運転
各種サーバ	連続運転
大判プリンタ	平日 9:00～21:00

サブシステム AOBAS の利用形態と制限値

利用形態	キュー名	VE 数※	実行形態	最大経過時間 既定値/最大値	メモリサイズ
無料	sxsf	1	1VE	1 時間/1 時間	96GB
共有	sxs	1	1VE	72 時間/720 時間	96GB×VE 数
		1～2048	8VE 単位で確保		
占有	個別設定				

※ 2VE以上を利用した並列実行にはMPIの利用が必用

サブシステム AOBAA の利用形態と制限値

利用形態	キュー名	VE 数※	実行形態	最大経過時間 既定値/最大値	メモリサイズ
無料	sxf	1	1VE	1 時間/1 時間	48GB
共有	sx	1	1VE	72 時間/720 時間	48GB×VE 数
		2～256	8VE 単位で確保		
占有	個別設定				

※ 2VE以上を利用した並列実行にはMPIの利用が必用

サブシステム AOBAB の利用形態と制限値

利用形態	キュー名	ノード数※	最大経過時間 既定値/最大値	メモリサイズ
共有	lx	1～16	72 時間/720 時間	256GB×ノード数
占有	個別設定			

※ 2ノード以上を利用した並列実行にはMPIの利用が必用

目次

東北大学サイバーサイエンスセンター

大規模科学計算システム広報 Vol.59 No.2 2026-4

[研究成果]		
反応硬化分子動力学によるエポキシ硬化挙動の解析と材料設計への示唆	佐藤 雅尚 鈴木 謙太 鈴木 貴也	1
[共同研究成果]		
有機超塩基触媒を用いる炭素-酸素結合変換	笹本 大空 是永 敏伸 重野 真徳	6
分子動力学計算による抗iPS細胞抗体と糖鎖リガンドの相互作用様式の同定	大野 詩歩	11
液滴の帯電挙動を考慮したVOFシミュレーション	松川 嘉也 酒田 一帆 宮川 泰明 齋藤 泰洋 岡部 孝裕 城田 農 大黒 正敏 福野 純一 青木 秀之	19
直交格子積み上げ法を用いた五芒星グレイン形状を有する 固体燃料ロケット内部流れ場の数値流体解析	小川 秦一郎 奥 汐梨 佐々木 大輔	26
Deep Q-NetworkとLESを用いたDBDプラズマアクチュエータの 翼周り流れの剥離制御	Tan Kevin 浅田 健吾 藤井 孝藏 立川 智章	35
Level set法による相変化解析手法の構築・検証と円管内気泡への適用	平井 克樹 川本 裕樹 行光 実桜 佐々木 大輔 高橋 俊 福家 英之	46
[大規模科学計算システム]		
【AOBA-Sの利用法】		
サブシステム AOBA-S の利用法		51
【AOBA-AおよびAOBA-Bの利用法】		
鍵ペアの作成とログイン方法		75
サブシステム AOBA-A の利用法		82
サブシステム AOBA-B の利用法		94
ストレージシステムの利用法		102
大判カラープリンタの利用法		108
[お知らせ]		
2026年度サイバサイエンスセンター講習会のご案内		114
[報告]		
2025年度サイバサイエンスセンター講習会実施報告		115
[スーパーコンピュータAOBAのお知らせより]		
コンパイラのバージョンアップについて		116
商用アプリケーションのバージョンアップについて		116
2026年度共同研究課題採択結果（第1回）について		117
計算科学・計算機科学人材育成のためのスーパーコンピュータ無償提供制度について		118
大規模科学計算システムの機関（部局）単位での利用について		118
民間企業利用サービスについて		119
2026年度利用負担金について		119
執筆要項		120
スタッフ便り		121