二相流解析コードへの in-situ 可視化の適用と評価

Application and evaluation of in-situ visualization to two-phase flow analysis code

○ 大日向大地, 富士通株式会社, 長野市鶴賀緑町 1415 大通りセンタービル, E-mail: obinata.daichi@fujitsu.com
南部太介, 宇宙航空研究開発機構, 調布市深大寺東町 7-44-1, E-mail: nambu.taisuke@jaxa.jp
溝渕泰寛, 宇宙航空研究開発機構, 調布市深大寺東町 7-44-1, E-mail: mizo@chofu.jaxa.jp

Daichi Obinata, Fujitsu Limited, Odori Center Building, 1415 Tsurugamidori-cho, Nagano-shi, Nagano
Taisuke Nambu, Japan Aerospace Exploration Agency, 7-44-1 Jindaijihigashi-machi, Chofu-shi, Tokyo
Yasuhiro Mizobuchi, Japan Aerospace Exploration Agency, 7-44-1 Jindaijihigashi-machi, Chofu-shi, Tokyo

The simulation of liquid fuel primary atomization requires high spatial resolution. However, as the simulation becomes larger, data transfer such as file I/O becomes severer, and post-hoc visualization is difficult in this situation. One solution of this problem is in-situ visualization. The authors applied in-situ visualization to the two-phase flow analysis code used in the simulation of the primary atomization of aircraft engine fuel. Interactive visualization is required for the simulation, but applyng tightly coupled in-situ visualization to interactive visualization is difficult. In the present study, the gas-liquid interface is saved as VTK format by in-situ visualization using VisIt/Libsim, and the subsequent processes such as rendering are performed by post-hoc visualization using ParaView. This procedure enabled interactive visualization of 31.2 billion cell-scale simulation.

1. はじめに

スーパーコンピュータなど HPC システムの性能向上に伴い，CFD を始めとした数値解析も大規模化が進んでいる．従来，シミュレーション結果の可視化はソルバでの計算結果をストレージに出力してから，可視化アプリケーションを用いて post-hoc(事後) に行うものであった．多くの計算機システムでもそのような利用を想定しており，例えば，本稿の計算で用いる JAXA スーパーコンピュータシステム (JSS2)[(1)] では，Lustre ベースの分散ストレージシステムと PC クラスタを組み合わせ，FieldView や ParaView といった遠隔並列可視化が行える可視化アプリケーションを整備して，大規模解析に対応した可視化環境が提供されている．

しかし，計算性能の向上とストレージ性能の向上には乖離があり，さらなる解析の大規模化はストレージシステムの存在が性能上のボトルネックとなって post-hoc での可視化を困難にしつつある．その解決方法として，ソルバで解析計算を行いながら可視化処理まで行う in-situ(その場) 可視化 [(2,3)] がある．

筆者らは，航空機エンジンの燃料一次微粒化解析をターゲットとして，その解析に用いる二相流解析コード [(5)] へ in-situ 可視化を実装した．本稿ではその実装および実行で得られた知見を述べる．

2. 燃料一次微粒化解析とその可視化

航空機エンジンに対する排気規制の強化に伴い，その開発に用いられる数値解析に対してもより一層の精度向上が求められている．排気成分の高精度な予測を達成するためには，燃料の分布を正しく捉える必要があり，燃料噴射直後の一次微粒化を正確に解析できていることが重要となる．しかし，一次微粒化の解析と燃焼器全体の解析を同時に行うことはその空間スケールの差から現実的ではない．一次微粒化に関しては何らかのモデル化が必要であり，数値解析による貢献が期待されている．

計算機性能の向上により，界面の挙動を VoF 法や Level-set 法で解像した詳細な数値解析が行われ [(4)]，実験による測定が難しい微粒化機構の詳細を確認することができる．微粒化機構の確認は，Figure. 1 に示すような気液境界面をベースとした可視化により行う．例えば計算結果を PLOT3D で出力しておき，FieldView のような可視化アプリケーションを用いた post-hoc 可視化でレベルセット関数の等値面処理によって気液境界面を可視化する．

一方，一次微粒化の解析は必要な格子解像度が高いという特徴があり，解析が大規模化しやすい．特にガスタービンエンジンのような高ウェーバー数の条件下では，微粒化後の液滴の空間スケール比が非常に小さくなり，解析に必要な格子解像度が非常に高くなる．そこで，直交格子法及び Immersed boundary 法により高解像度に対応し，かつ任意形状での解析にも適用可能な非圧縮性二相流体解析コードの開発が進んでおり [(5)]，その可視化には in-situ 可視化を用いることが適当である．

Fig. 1: An example of gas-liquid interface visualization.

3. in-situ 可視化の適用

3.1 適用の基本方針

CFD の可視化の処理フローは一般に，ソルバの計算結果 (空間ボリュームデータ) を入力として，(1) フィルタリング，(2) マッピング，(3) レンダリングの 3 段階で構成される．フィルタリングでは，ボリュームデータから等値面や断面などといった幾何形状を生成・抽出する．マッピングでは幾何形状に対し，各物理量に合わせて色や透過率を設定する．レンダリングでは視点など構図のパラメータから最終的な可視化画像を生成する．

in-situ 可視化の最も代表的な方式は Thightly coupled と呼ばれる，ソルバのプロセスで可視化処理の全てを行う方式であるが，Thightly coupled in-situ 可視化は比較的バッチに向いた方式とされている．例えば，文献 (7) では自動車の空力解析に Tightly coupled in-situ 可視化を適用し，レンダリングまでの全ての可視化処理を in-situ 処理で行っている．バッチ可視化としての性質が強く，性能面では時間ステップ 200 回ごとに in-situ 可視化を行った場合に解析時間の 63%が可視化処理に使われるようになったものの，可視化を含めたワークフローでは十分高速であるという評価を行っている．また，文献 (8) では，

Tightly coupled in-situ 可視化でインタラクティブ可視化を実用することの困難さ示している．

一方，微粒化解析では，微粒化の過程を様々な視点で可視化したい要請があり，その実現にはインタラクティブ性を要する．しかし，Tightly coupled で全ての可視化処理を in-situ 処理で行うと，ソルバ実行と可視化が同期せざるを得ず，視点を僅かに変えるだけでもソルバでの再計算が必要になり，大規模解析のジョブ待ちなど計算リソースの確保から考えるとインタラクティブ性の確保は現実的でない．そこで，in-situ 可視化の適用にあたっては，可視化処理フローの一部をソルバ側での in-situ 処理として行うことを検討した．

前節で示したように，燃料一次微粒化解析の可視化では気液境界面を使うことが確定的であり，気液境界面の抽出はバッチ的に処理できる．そこで，可視化処理フローのうち，フィルタリングまでを in-situ 処理で行って気液境界面を抽出し，マッピング以降を post-hoc 処理でインタラクティブに行う方法をとることにした．この方法を用いると，可視化のフィルタリングとマッピング以降が分離されて非同期で行えるため，視点変更などの操作は高いインタラクティブ性が期待できる．また，ソルバの出力データは気液境界面を示す等値面の幾何形状と面上の物理量となり，空間ボリュームで出力する場合に比べて出力データを小さくすることができる．

3.2 使用する計算機とソフトウェア，データ形式

前節の基本方針に基づき，ソルバとなる二相流解析コードに in-situ 可視化を適用した．使用する計算機とソフトウェア環境およびデータ形式は次の通りである．

ソルバは JSS2 のメイン計算システム SORA-MA (FUJITSU Supercomputer PRIMEHPC FX100) で実行し，in-situ 可視化によって気液境界面となる等値面データを抽出する．ソルバに組み込む in-situ 可視化処理は既存ライブラリを使用する．in-situ 可視化ライブラリは，著名なものとして VisIt の Libsim[3] や ParaView の Catalyst[6] などがあり，今回は SORA-MA で使えることが確認できている VisIt 2.10.2 の Libsim を用いることとした．

等値面データの post-hoc 可視化は，JSS2 のプレポストシステム SORA-PP (PC クラスタ) でインタラクティブに行う．SORA-PP は可視化アプリケーションとして ParaView や FieldView を使って並列可視化を行うことができ，また GPU を搭載しておりレンダリング処理をハードウェアでアクセラレートすることができる．

気液境界面となる等値面のデータは VTK (Visualization Tool Kit) 形式で出力することとした．従来，燃料一次微粒化解析の post-hoc 可視化には FieldView を用いていたことから，当初は等値面データを XDB 形式[9] で出力することを考えた．しかし，Libsim の XDB 出力モジュールが FX100 の CPU である SPARC アーキテクチャに対応していないこと，VTK 形式にネイティブに対応する ParaView が SORA-PP で使えること，VTK 形式自体の汎用性の高さなどの理由から VTK 形式を選択した．

3.3 適用上の課題

上記のように二相流解析コードに in-situ 可視化を実装したところ，いくつか対処すべき問題が生じた．以下にその問題と行った対処について述べる．

(1) 等値面データが Legacy VTK 形式になる

VTK 形式には古典的な Legacy VTK 形式 (以下，Legacy 形式) とモダンな VTK XML 形式 (以下，XML 形式) がある．特に，領域分割で並列化されたソルバを使うような解析では，1 領域 (ブロック)1 ファイルのマルチブロックデータで書き出しておくと ParaView の並列可視化を効果的に使える．ただし，VTK でマルチブロックデータを構成するには XML 形式である必要がある．

Libsim の API 仕様上は Legacy 形式，XML 形式のどちらの形式でも出力可能であるが，実際に XML 形式で出力しようとすると等値面の要素が無い領域の書き出しで処理がブロックされる現象に遭遇した．一方，Legacy 形式ではそのようなブロッキングは発生しない．そこで，in-situ 可視化で出力する等値面データは Legacy 形式で書き出しておき，Legacy 形式から XML 形式に変換することとした (Figure. 2)．

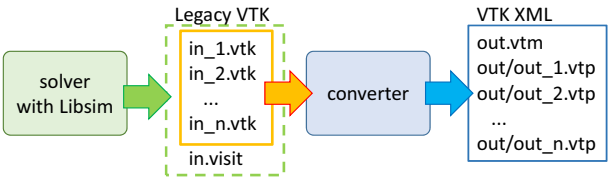

Fig. 2: Conversion from Legacy VTK format written by solver to VTK XML format.

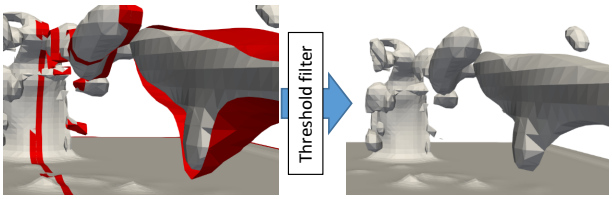

Fig. 3: Removement of “burr” elements of iso-surface (colored in red) by Threshold filter.

この変換処理はソルバ実行後 1 回だけ行い，変換後の XML 形式のデータをソルバの計算結果として残しておく．

(2) 袖領域が可視になる

ソルバは領域分割で並列化されているため，分割後の各領域には袖領域 (Ghost zone) がある．そのため，in-situ 可視化で抽出した等値面は袖領域まで広がりをもつことになるが，このような等値面をファイルに出力するときに Libsim は袖領域を除去しない．また，ParaView で可視化すると袖領域は不可視にはなっておらず，袖領域にかかった等値面が領域境界付近にバリのような不連続面として現れる (Figure. 3)．

Libsim では袖領域の除去は行わないが，avtGhostZone というセル定義の整数値で袖領域であることの情報を与えている．そこで，ParaView での post-hoc 可視化時に Threshold フィルタを用いて $avtGhostZone = 1$ の面要素を除去することとした．

(3) 実行時ファイル数が増える

Libsim はその依存ライブラリ類も含めると多数の shared object ファイルから構成されており，ソルバ実行時にはそれら多数の shared object ファイルが各プロセスにおいて動的にリンクされる．

ソルバはフラット MPI で並列化されていため，計算資源の利用効率で言えば，ノードあたりの CPU コア数 (SORA-MA の場合は 32) だけプロセスを詰めて実行するのが良い．しかし，ソルバに Libsim を組み込んだ状態でノードあたりプロセス数を 32 で実行しようとすると，ノードあたりのオープン可能なファイル数の上限 (FX100 用 OS の制限値) に達してしまい，ライブラリを正常にリンクできずソルバを実行できない事象が発生した．

対処として，Libsim や関連するライブラリを静的リンクライブラリとして再構築し，shared object ファイルの数を減らすことを試みたがうまくいかなかった．最終的には，ノードあたりのプロセス数を 20 まで減らして実行することとした．

4. 実行 · 評価方法

in-situ 可視化適用の評価は，Figure. 4 に示す解析ワークフローに沿って行う．ワークフローの in-situ 可視化適用前 (Conventional way) と適用後 (New way) それぞれ

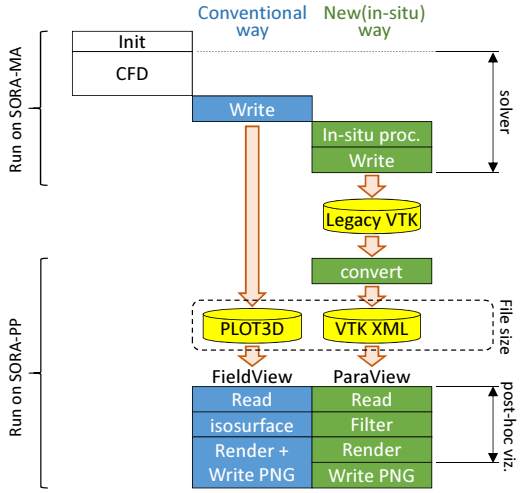

Fig. 4: Workflows of conventional way and new way using in-situ visualization. Performance evaluation sections are also shown in the figure.

Tab. 1: Size of problems to be measured.

Cells	Parallel (Process / Compute node)		
	solver	FieldView	ParaView
3.9 bil.	7712/ 386	16/ 2	24/2
31.2 bil.	56884/2845	24/24	48/4

のパスの各処理に要する時間と，書き出しファイルサイズを測定し，時間ステップあたりの平均値を求める．

in-situ 可視化適用前後の効果の評価は，次に示す 2 つの区間の時間とファイルサイズの比較で行う．

solver ソルバ側の CFD 計算，in-situ 可視化処理，ファイル書き出しに要する合計時間を比較する．

post-hoc viz. 可視化アプリケーション側での，ファイル読み込みとフィルタリング (FieldView の場合は等値面処理，ParaView は可視化パイプライン) およびレンダリング (マッピングを含む) の合計時間を比較する．この区間は，post-hoc 可視化処理をインタラクティブに行う場合に支配的になる部分である．

File size ソルバが出力するファイルのサイズ．in-situ 可視化で出力する VTK ファイルのサイズは XML 形式への変換後のサイズを用いる．ブロックごとのファイルを束ねるメタファイルのサイズを含む．

測定対象となる解析は，39 億セル (3.9 bil.) と 312 億セル (31.2 bil.) の 2 種類のサイズとした．39 億セルは JSS2 で実際に解析を行っている微粒化解析では最大規模のものであり，312 億セルは近い将来解析が必要になると想定される規模である．

それぞれの解析サイズで用いた計算リソース規模を Table. 1 に示す．312 億セルケースでのソルバ実行は SORA-MA(3240 ノード) の約 90%を使用するため，全系を占有して実行した．post-hoc 可視化で使用する FieldView と ParaView の並列数は，必要なメモリサイズ，実際の可視化性能および使用可能なライセンス (FieldView の場合) のバランスで決定した．

5. 実行結果 · 評価および考察

5.1 可視化結果

312 億セルケースを in-situ 可視化を用いて可視化処理した結果の画像を Figure. 5 に示す．細かく粒化する様子が解像できており，可視化処理が機能していることを確認できた．

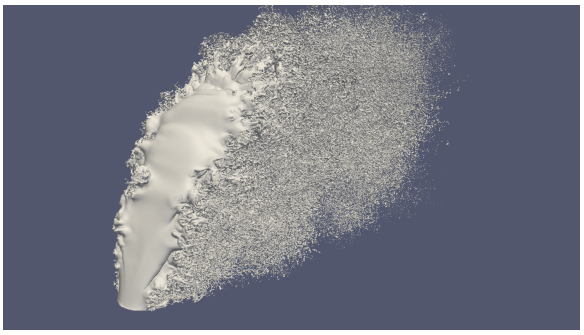

Fig. 5: Visualization of gas-liquid interference in 31.2 billion cells case with in-situ visualization.

Tab. 2: Elapsed times [sec] of respective analysis flow stages and file sizes [GB] at a single time step.

	3.9 bil.		31.2 bil.	
	Conv.	New	Conv.	New
CFD	12.6	12.6	71.4	71.4
in-situ proc.	—	1.5	—	6.3
Write	74.6	21.0	232.5	17.7
convert	—	3.4	—	21.0
Read	209.2	1.0	1127.1	4.1
post-hoc proc.	121.6	1.1	373.5	5.5
Render	1.5	0.5	9.5	3.1
Write PNG		1.9		17.7
File size	98.7	0.4	725.5	2.9

Tab. 3: Effectivity ratio by the new(with in-situ) way.

	3.9 bil.	31.2 bil.
solver	2.5	3.2
post-hoc viz.	127.3	118.6
File size	246.9	245.4

5.2 測定対象区間の評価

解析ワークフロー (Figure. 4) の各処理の時間測定の結果を Table. 2 に示す．また，評価区間ごとの in-situ 可視化導入の効果を Table. 3 に示す．

in-situ 可視化の導入によって，ソルバは 2~3 倍程度，post-hoc 可視化は 100 倍以上に高速化 (時間短縮) した．ファイルサイズは，出力データが空間ボリュームから等値面になったことにより，1/250 程度のサイズまで小さくなっている．ソルバや post-hoc 可視化の実行時間が短縮されたのは，このファイルサイズ縮小によって書き出し読み込みの負荷が軽減された効果が大きい．また，多くの演算を要する等値面抽出の処理をソルバ側で高並列に行っていることも，post-hoc 可視化の時間短縮に大きく寄与している．

なお，312 億セルケースで in-situ 可視化を行った場合のソルバの書き出し時間 (Write) が 17.7 秒と，39 億セルケースの 21.0 秒より短い．これは，39 億セルは JSS2 の通常運用中に実行し，312 億セルは JSS2 を全系占有して実行したため，ストレージシステムの負荷状態が著しく

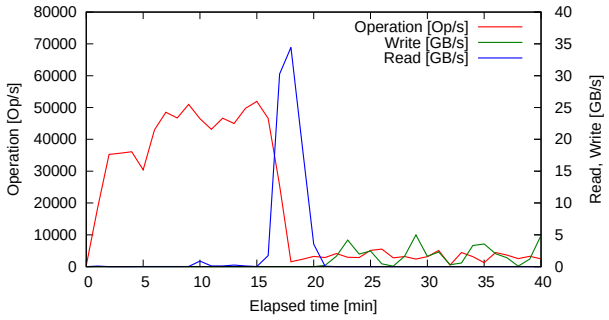

Fig. 6: Time history of file system load in 31.2 billion cells case.

異なっていたことによるものと考えられる．

5.3 ソルバの実行開始に要した時間

312 億セルのケースでジョブの実行開始からソルバプロセスが計算を開始するまで 20 分ほどかかった．

当該ジョブ実行時のファイルシステムの負荷状況を調べたところ，Figure. 6 に示すように，$t = 0$ (ジョブの開始) から Read の立ち上がり (ソルバがリスタートデータを読み込み始める) まで，ファイルの操作 (ファイルオープンなど) の負荷が高い状態が 17 分ほど続いていたことが判明した．これは，3.3 節において指摘した in-situ 可視化適用に伴い実行時ファイル数が増えたことによるものと考えられ，このことがソルバ実行開始時のボトルネックになっていたと推測される．

in-situ 可視化の導入は外部ライブラリに依存するところが大きく，shared object ファイルなど実行時に必要になるファイル数が増えることは避けられない．また，ライブラリ自体の機能の拡充によって，依存するファイル数が今後さらに増えることも予想される．

回避方法の 1 つとして，ソルバの並列化にハイブリッド並列を用いることでプロセス数を減らす方法が考えられるが，その場合，in-situ 可視化ライブラリ側でもスレッド並列に対応しなければ，in-situ 処理の性能が得られない可能性がある．in-situ 可視化の実装にあたっては，このようなシステムリソースに対する考慮も必要といえる．

5.4 in-transit 可視化との比較

ソルバと可視化を分離しながらファイル I/O レスで可視化を行う方式として，in-transit 可視化 [10] がある．これは，ソルバの計算結果データを可視化用計算機にステージング・転送しながら可視化を行う方式で，ソルバと可視化が同期する Tightly coupled 方式の in-situ 可視化のデメリットを解消し，非同期かつインタラクティブに可視化を行えるものである．その反面，実現にかかるコストが大きく，また計算機間でボリュームデータの転送を要するため通信負荷が高い [11]．

一方，5.2 節の結果を踏まえると，可視化の方針が決まっている場合は，可視化処理の一部を in-situ 処理としておくことで，たとえファイル I/O があっても十分軽量でインタラクティブな可視化が行える．また，in-transit 可視化を導入する場合にも，可視化処理の一部をソルバ側の in-situ 処理で行うことができれば，ボリュームデータの転送を必要とせず，軽量な面データの転送で済むようになることが考えられる．

6. まとめ

燃料一次微粒化の解析は必要とする格子解像度が高く，その可視化には in-situ 可視化を用いることが適当と考えられることから，航空機エンジンの燃料一次微粒化解析に用いる二相流解析コードに in-situ 可視化を実装した．

in-situ 可視化の実装にあたっては，可視化処理のうち等値面の抽出までをソルバ側での in-situ 処理とし，等値面データをストレージに書き出した後，post-hoc 処理で可視化画像の生成を行う方式を採用した．これにより可視化処理の時間短縮と，ソルバの出力データのコンパクト化が実現でき，312 億セルの大規模解析において，高い解像度の可視化が短時間かつインタラクティブにおこなえることを確認した．

謝辞

宇宙航空研究開発機構スーパーコンピュータシステム “JSS2” を利用した．312 億セルケースの測定にあたっては SORA-MA を全系占有するため，JSS2 の運用スタッフに協力いただいた．

参考文献

(1) “JSS2@JAXA,” `https://www.jss.jaxa.jp/`

(2) 松岡大祐, 荒木文明. “大規模シミュレーションデータ可視化における研究の動向,” *JAMSTEC Report of Research and Development*, Vol.13, pp.35–63 (2011)

(3) Whitlock, B., Favre, J., & Meredith, J. “Parallel In Situ Coupling of Simulation with a Fully Featured Visualization System,” *Eurographics Symposium on Parallel Graphics and Visualization, EGPGV 2011*, pp.101–109 (2011)

(4) Li, X., & Soteriou, M. C. “Detailed numerical simulation of liquid jet atomization in crossflow of increasing density,” *International Journal of Multiphase Flow*, Vol.104, pp.214–232 (2018)

(5) 南部太介, 溝渕泰寛. “ガスタービンエンジン燃焼器条件下におけるクロスフロー型燃料微粒化機構の詳細数値解析,” 第 57 回燃焼シンポジウム, D331 (2019)

(6) Ayachit, U., Bauer, A., Geveci, B., O’Leary, P., Moreland, K., Fabian, N. & Mauldin, J. “ParaView Catalyst: Enabling In Situ Data Analysis and Visualization,” *Proceedings of the First Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization - ISAV2015*, pp.25–29 (2015)

(7) 大西慶治, ヤンソン・ニクラス, 坪倉誠. “大規模自動車空力解析における In-situ 可視化,” 計算工学講演会論文集, Vol.24 (2019)

(8) 堤誠司, 藤田直行, 大日向大地, 伊藤浩之. “HPC 環境における in-situ 可視化の利用と課題,” 第 31 回数値流体力学シンポジウム, C10-1 (2017)

(9) “XDB – Intelligent Light and FieldView,” `http://www.ilight.com/en/products/fieldview-power-tools/xdb` (cited: Oct 16, 2019)

(10) Moreland, K., Hereld, M., Papka, M., Klasky, S., Oldfield, R., Marion, P., et al. “Examples of in transit visualization,” *Proceedings of the 2nd international workshop on Petascal data analytics challenges and opportunities - PDAC ’11*, pp.1–6 (2011)

(11) 堤誠司, 藤田直行, 伊藤浩之, 大日向大地, 井上敬介, 松村洋祐, 高橋慧智, Eisenhauer, G., Podhorszki, N. & Klasky, S. “In Situ/In Transit アプローチを用いた大規模数値解析におけるポスト処理効率化,” 第 33 回数値流体力学シンポジウム, B12-1 (2019)