

構造解析シミュレーションにおけるボトルネック解析および線形ソルバの閾値最適化による高速化の見積もり

ハデルバシュ アミル^{1,a)} 白幡 晃一¹ 中川 香苗¹ 奥田 洋司² 富田 安基¹

概要： 計算機ハードウェアの発展に伴い、理工学の広い分野において有限要素法の大規模・高速化が求められている。本研究では、構造解析シミュレーションにおいて、どのような処理がボトルネックになっているかをプロファイリングにより詳細に解析した。さらにその解析結果を基に、実用的な計算精度を担保しつつ、線形ソルバの残差の閾値を変えた場合の高速化の効果を見積もった。

1. はじめに

ムーア則と呼ばれるトランジスタのスケールアップによるコンピュータの性能向上が終焉を迎えようとしている [1]。一方で、IoT やビッグデータによりデータ解析のためにコンピューティング能力は更なる向上が求められている。このような状況下、コンピューティングアーキテクチャーは変革を求められており、deep learning のような膨大な演算を要するアプリケーション [2] では、そのアプリケーションの主要処理を得意とする GPU のようなアクセラレータを用いるドメイン指向コンピューティングが既に一般的となってきた。このアクセラレータを用いるアーキテクチャーは、近年では演算量が多い科学計算を行うための ABCI [3] のような最新のスーパーコンピュータ (HPC) でも同様にスタンダードになりつつある [4]。本研究では、このドメイン指向の考え方を HPC で実行されるシミュレーションにアルゴリズムレベルで適用することで、シミュレーション内部のソルバを高速化することを目指している。

ものづくり分野では低コストな設計を行うため、実機を試作する前に、流体解析や構造解析などの Computer-Aided Engineering (CAE) を用いたコンピュータシミュレーションによる設計と特性確認を行っている。その設計では、ターゲットとする物体の形状や材質などを設計するために、何千回、何万回もシミュレーションを実行する必要がある。1 回のシミュレーションも大規模で複雑な非線形解析を行うため解に収束するまでに大量の演算時間を必要とする。さらに上記設計パラメータ以外にも、線形ソルバの残差の閾値やタイムステップの刻み幅等、シミュレーション自体

を適切に進行させるための変数が多数存在し、これら全てを最適化するのは困難であり、従来、設計者はデフォルト値や経験に基づいた値に固定して利用している。

本研究では非線形ソルバの高速化を実現するために、一連のプロファイリングを行い、非線形解析の最内ループである線形ソルバがメインの性能ボトルネックであることを確認した。また、その結果から、シミュレーションの設定変数の一つである線形ソルバの残差の閾値を変化させることにより、閾値が小さすぎる場合には、閾値を大きくしても精度があまり劣化せずソルバの実行時間が短縮されることを確認した。これらの知見を用い、ここで線形ソルバの残差の閾値を適切に変更することで高速化の手法を提案する。提案する手法によって閾値を変更したシミュレーションでは、変更前のデフォルト状態に比べ、1.12~1.65 倍の高速化の可能性があることを見積もった。本論文がもたらす主要な貢献は次の通りである。

- 性能プロファイルにより非線形シミュレーションのボトルネックが線形ソルバにあることを確認した。
- その線形ソルバの収束残差の閾値変更により実行時間と精度の間にトレードオフの関係があることを明らかにした。
- 線形ソルバの収束残差の閾値を適切に変更することで、デフォルト状態と比較して精度を保ちながら高速化できる可能性を示した。

本論文は次のように構成されている。2 章と 3 章では関連研究と本研究のバックグラウンドとして非線形解析のメカニズムを解説する。4 章ではプロファイル性能の実験結果と考察、最後に線形ソルバの収束残差の閾値を変更することによる高速化の見積もりを示す。

¹ 株式会社富士通研究所

² 東京大学 新領域創成科学研究科

a) haderbache.amir@fujitsu.com

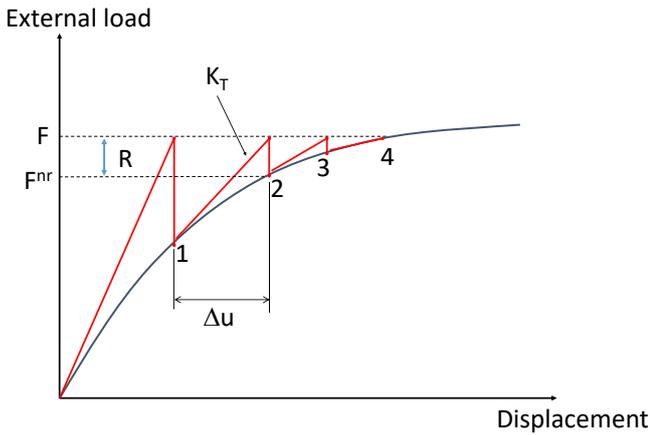


図 1 非線形解析におけるニュートン・ラプソン法

2. 関連研究

最近、力学 [5], [6], 地球科学および生命科学 [7], [8], 物理学 [9], 化学 [10] などの様々な計算科学分野 [11] に AI が適用され始めている。ここでは、その例を紹介する。その例としては、偏微分方程式をより速く解くこと [12], N 体問題を解くこと [13], またはマルコフ連鎖モンテカルロカーネルのパラメータ化 [14] などにおいて、ニューラルネットワークを従来の数値解法に融合させる新しいアプローチが挙げられる。また、他にも、タスク実行時間のような性能データを用いた大規模マシンでの動的ジョブスケジューリングなど、「純粋な」HPC アプリケーションでも機械学習は使用されている [15], [16]。一方、A. Marathe らは、大規模環境でのアプリケーションの最適性能条件を予測することを、大量の小規模マシンで学習したモデルを用い転移学習で実現した [17]。

線形ソルバに対して AI を適用する最近の研究では、機械学習モデルを使用して、最も速くなるソルバや前処理の組み合わせを選択するものがある [18], [19]。特定の入力問題に対し、その行列特性（非ゼロ要素の構造, 数値特性）と各ソルバや前処理での性能結果を学習データのセットとして用い、構築した予測モデルにより最適なソルバや前処理を選択する。このアプローチは、類似の特定の問題が与えられたとき、最も高性能になる線形ソルバと前処理をユーザが使えるようにニューラルネットの分類問題に落とし込んでいる。一方、本論文は、材料力学における非線形解析を解くための有限要素法やニュートン・ラプソン法のような数値解法に関するものである。シミュレーションに必要な設定（線形ソルバの残差の閾値）を最適値に変更することで高速化を可能としており、我々のアプローチは前処理などには依存しないので、上記研究で選択した最適なソルバや前処理に対しても適用可能であり、線形ソルバの反復を何度も行う非線形解析の高速化に有用である。

表 1 評価に用いる FrontISTR モデル

	自由度	行列の要素数	行列の密度
Plastic Cylinder	1,887	3.56×10^6	3.36×10^{-2}
Plastic Can	42,357	1.79×10^9	1.65×10^{-3}
半田	88,281	7.79×10^9	7.98×10^{-4}

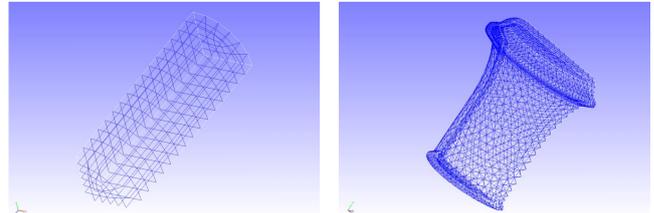


図 2 各チュートリアルモデルのメッシュ表示 (左: Plastic Cylinder, 右: Plastic Can)

3. 準備: 非線形解析

本章では非線形方程式を解くための数値解法である非線形解析について述べる。

3.1 有限要素法における線形ソルバ

本研究のシステムは3つの項目から構成される。

- シミュレーションのターゲット物体となる有限要素（メッシュと呼ばれる）に分割された空間領域
- シミュレーションの一般的な制約（タイムステップ, 材料特性, 負荷など）を設定する境界条件
- シミュレーション中に発生する特定の物理現象を記述する偏微分方程式

有限要素法 (FEM) は、上記の項目から線形システム $Kd = F$ を構築する。行列 K は剛性マトリックスと呼ばれ、任意の負荷に対するシステムの応答を表している。ベクトル F は境界条件において定義される外部からの負荷であり、 d は我々が求めたい変位節点の値を表す未知の解ベクトルである。FEM は、システムの要素単位の局所的な振る舞いから、システム全体を表現するグローバル行列ベクトル方程式を作成し、線形ソルバは、このシステムの解を計算する。本論文では、構造解析（材料力学）のシステムを使うが、提案する手法は他の非線形解析にも適用可能であることに留意されたい。

3.2 線形解析と非線形解析

構造解析等における解析は、境界条件（与えられた負荷）とその効果（変位）との関係に依存して、線形解析または非線形解析に分類される。線形解析では、与えられた負荷とその変位の関係は線形であり、その剛性マトリックスは一定となる (FEM によって単一の線形システムが生成される)。それを解くための線形ソルバは 1 回 call され、処理時間も短い。一方で、非線形解析では負荷状態によって剛

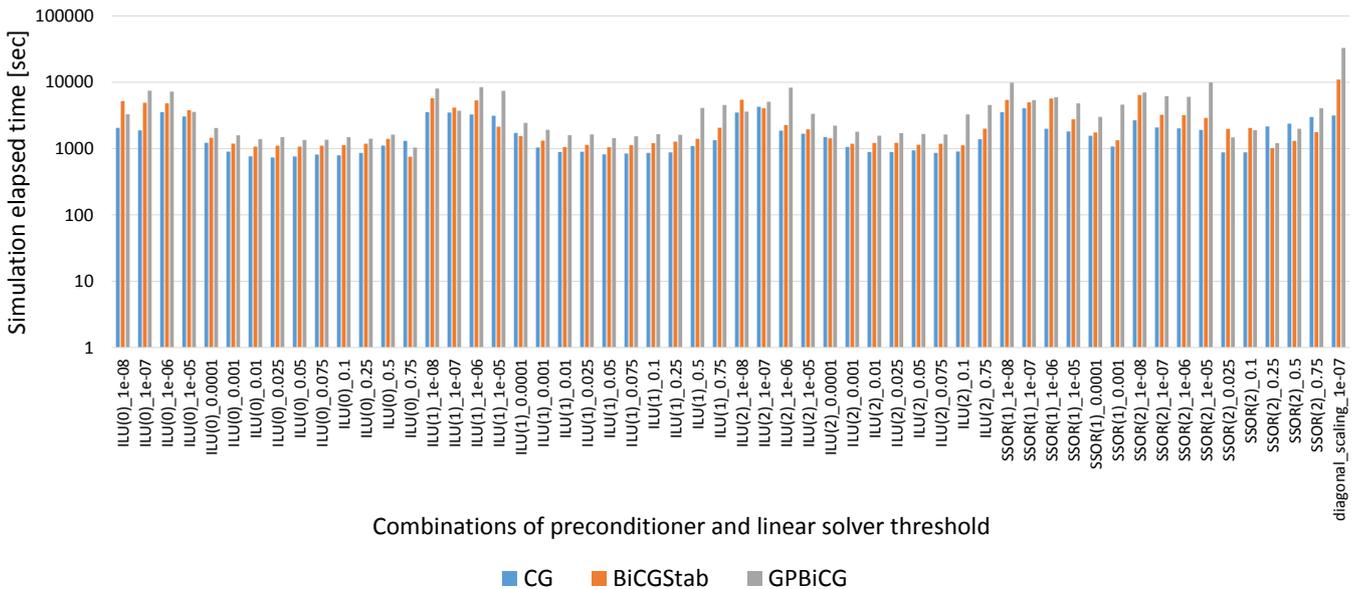


図 3 線形ソルバの各反復法とでの半田モデルのシミュレーション実行時間の比較: 複数の線形ソルバと線形ソルバの閾値に対してシミュレーションの実行時間を測定

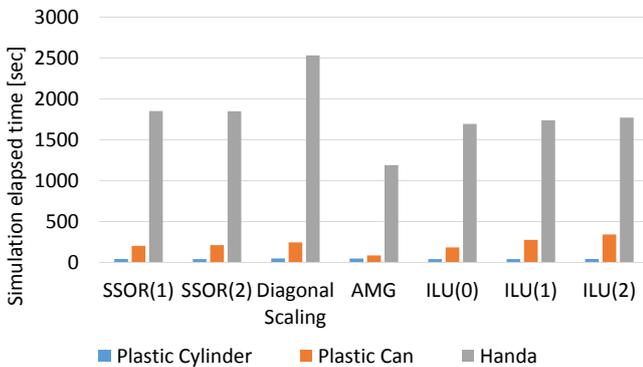


図 4 CG 法における各前処理でのシミュレーション実行時間の比較

性マトリックスは一定ではなく、それを解く処理もより複雑である。例としては、ターゲットとする物体が、与えられた負荷により大きく変形するような幾何学的非線形問題などがある。このような問題の解析では、物体の形が非線形に幾何学的変化することを考慮に入れる必要がある。

3.3 非線形解析とニュートン・ラプソン法

非線形解析を解く方法の一つに、実関数のための反復アルゴリズムであるニュートン・ラプソン (NR) 法がある。線形解析では負荷はターゲットの物性値や変形状態に影響を及ぼさないが、非線形解析では多くの "time step" (または "sub step") を通じて負荷が徐々に与えられる。各 time step では現在の「解」から最終の「解」に向かうために、負荷を増加させながら多数の NR 法の反復が行われる。このとき、各 sub step では負荷が徐々に与えられ、各 sub step 内の各 NR step では負荷は一定である (すなわち、図 1 における F は各 NR step で一定である)。NR 法の反復の収束点は、外部からの負荷 F と反力としての内力 F^{nr} の差が、ある閾値より小さいときに得られる。この差を、残差

R と呼ぶ (すなわち、図 1 における $F - F^{nr}$ が残差 R となる)。NR 法の各反復では、FEM は新しい接線剛性マトリックス K_T を生成し、新しい線形システム $K_T * \Delta u = R$ を生成する。次に NR 法のアルゴリズムが線形ソルバを呼び出し、新しい変位の増加分 Δu を計算する。この値は、変位の値 (解ベクトル) を更新するために使用される。このとき、現在の残差 R を計算し、それを閾値と比較することによって収束を判定する。収束と判定されると、NR 法は終了する。そうでない場合は、次の新しい反復が始まる。図 1 はこのメカニズムを収束するまで 4 回 NR 法の反復が行われている例を示している。

4. 性能プロファイリング

非線形解析シミュレーションの性能プロファイリングを、2 個の Intel Xeon E5-2695v4 CPU (72 コア) と 128GB DDR4-2400RAM を備える 1 つのシングルノードマシンを用いて実施した。今回使用したシミュレーションソフトウェアは、構造解析を行うオープンソースソフトウェアの FrontISTR [6] で、それ自身にソルバライブラリ (hecmw) の一機能を備えている FEM のプログラムである。今回、プロファイリングを実施したアプリケーションモデルは FrontISTR に同梱されているチュートリアルから Plastic Cylinder と Plastic Can, 実例題として熱荷重を与えて半田に生じる非線形ひずみを評価するためのモデル (以下半田モデルと呼ぶ) の 3 種類を利用した (図 2)。各モデルの自由度、行列の要素数、行列の非ゼロ要素の密度は表 1 に示す通りである。

4.1 線形ソルバ: 反復法と前処理

非線形解析シミュレーションでは、NR 法の各反復にお

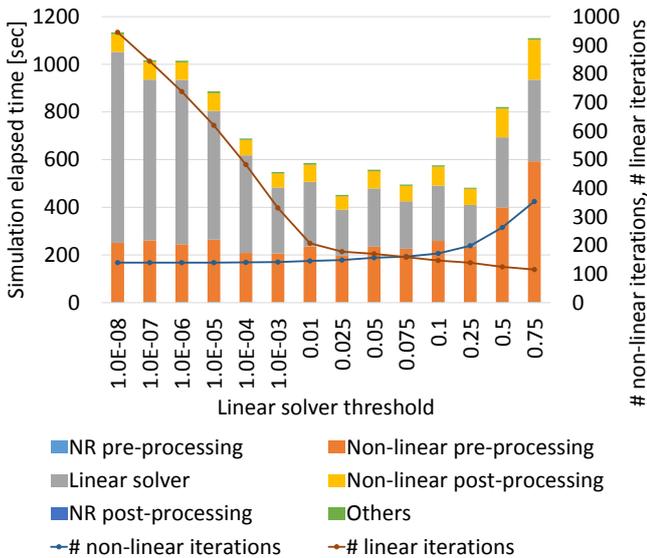


図 5 半田: 線形ソルバの各閾値に対する実行時間と反復回数

いて、線形方程式を解いている。一般的に、線形システム $Ax = B$ の解法には直接法と反復法がある。前者は比較的小規模な問題を解くのに適しており、後者は大規模問題を解く際に用いられる。本論文では、反復法を用いることを想定する。よく用いられる反復法として Krylov 部分空間法 (共役勾配法 (CG 法) およびその変形を含む) がある。Krylov 部分空間法では、解ベクトル x を繰り返し更新しながら解を収束させる。収束に達していない間は、各反復回数 k において、前処理行列 P を用いたベクトル $P^{-1}r_k$ を使用して $x_{k+1} = x_k + P^{-1}r_k$ から x_k を更新する。反復回数 k での残差を $r_k = b - Ax_k$ と定義すると、この値は $Ax = b$ の誤差である (x 自体の誤差ではない)。この残差値を各反復で計算し、定めておいた閾値 t と比較し、 $r < t$ の場合、収束したとして線形方程式は解かれたと見なす。ここで、NR 法の収束で使用される前述の非線形解析の残差 R と線形ソルバの収束で使用される線形残差 r を混同しないよう注意されたい。

このように、反復法の線形ソルバを用いるには、反復法だけでなく前処理行列 P も選択する必要があることが分かる。前処理の目的は、ターゲットとする問題をより適切な形式に変換し、反復法でより簡単に解けるようにすることである。理論的には、前処理行列 P は係数行列の逆行列 A^{-1} に近いほど、反復法が収束しやすい。線形ソルバの反復法及び前処理の組合せを選択するために、様々な条件で FrontISTR のシミュレーションを実行し性能評価を実施した。図 3 は、線形ソルバの各反復法での実行時間の比較である。CG 法が他の反復法と比較し、多数の条件 (前処理、閾値) でシミュレーション時間が 1,000 秒未満と高速に実行出来ていることを示している。さらに、CG 法に限定し、表 1 に示すアプリケーションモデルを使用して前処理の性能評価したものを図 4 に示す。AMG (Algebraic Multigrid Methods) の前処理が評価した全てのアプリケーションモ

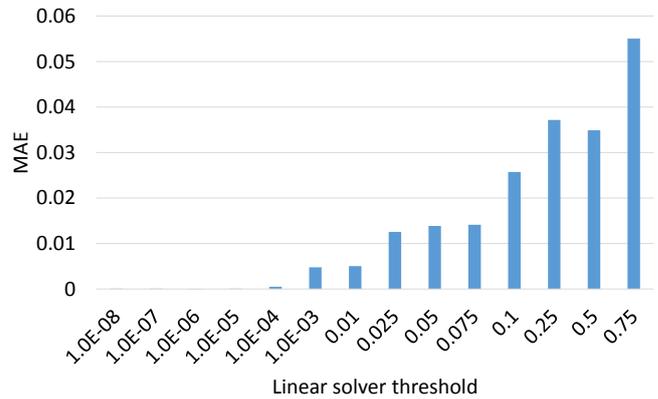


図 6 半田: 各閾値に対する平均絶対誤差 (MAE)

デルにとって最も高い性能を示している。これらの結果から、以下においては、CG 法-AMG 前処理を用いた非線形解析を対象としている。

4.2 線形ソルバの残差閾値による性能プロファイリング

上述したように、非線形解析は NR 法の各反復において線形ソルバを呼び出し反復処理を行う。4.1 節で示した CG 法-AMG 前処理でアプリケーションモデルの中で最も時間がかかっている半田モデルについて実行時間内訳のプロファイリングを行った結果を図 5 に示す。見て分かるように、線形ソルバの残差閾値がデフォルト値 1.0×10^{-8} である一番左の棒グラフの実行時間内訳を見ると、非線形解析の最内ループである線形ソルバの実行時間が 70.7% を占めておりボトルネックとなっていることが分かる。そこで、各アプリケーションモデルにおいて、CG 法の線形ソルバの残差 r の閾値を変えてシミュレーションを行い、同様に性能プロファイリングを実施した。線形ソルバの残差の閾値は、残差が収束したかどうかの厳しさの基準であり、設計者がシミュレーションを行う際に定義しなければならない実数パラメータである。閾値が小さければ小さいほど、反復法の収束判定が困難になり長時間を要する。しかし、閾値が小さければ、直接法で得られる解に近い正確な結果を得ることが出来る。したがって、閾値を選択する際は、計算時間と結果の正確さとの間のトレードオフを考慮しなければならない。

図 5、図 7、図 9 は、それぞれ半田モデル、Plastic Cylinder、Plastic Can とにおいて、線形ソルバの残差閾値の変化による実行時間の影響を棒グラフで、そのときの非線形解析及び線形ソルバの反復回数を折れ線グラフで示している。非線形解析の閾値はデフォルト値に固定しており、線形ソルバの閾値 (x 軸) は、 1.0×10^{-8} から 1.0 まで変化させている。見て分かるように、線形ソルバの閾値を小さい値から大きくしていくと、線形ソルバの反復回数が減っていき全体の実行時間が速くなっている。しかしながら、閾値を大きくし過ぎると、ある所から NR 法の反復回数が増幅に転じ、全体の実行時間が遅くなってしまっている。

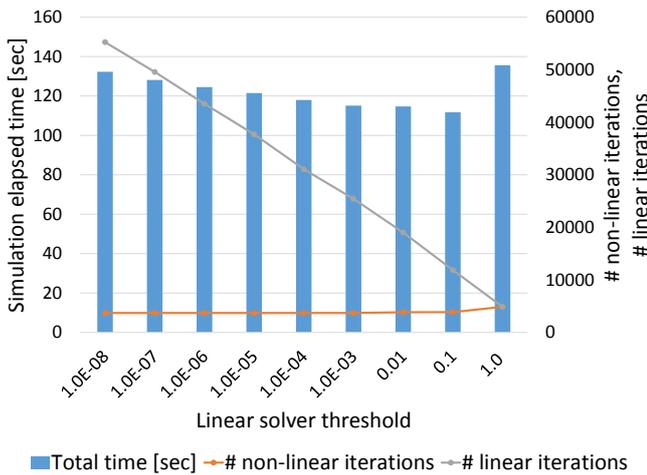


図 7 Plastic Cylinder: 線形ソルバの各閾値に対する実行時間と反復回数

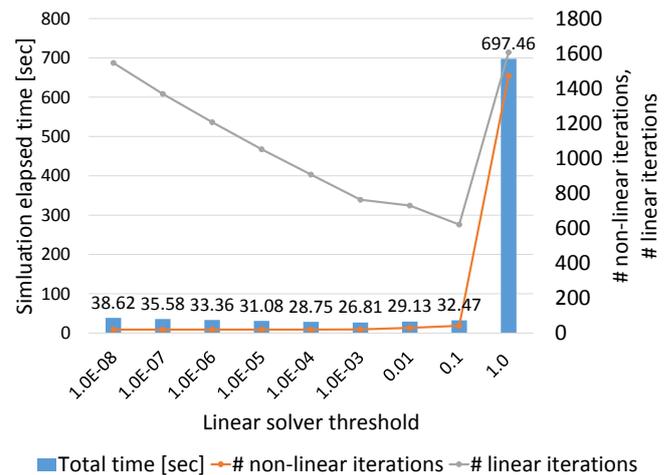


図 9 Plastic Can: 線形ソルバの各閾値に対する実行時間と反復回数: 各棒の上にシミュレーション実行時間を表示

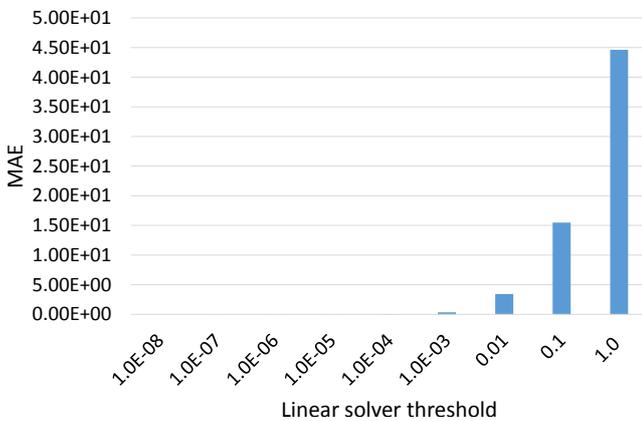


図 8 Plastic Cylinder: 各閾値に対する平均絶対誤差 (MAE)

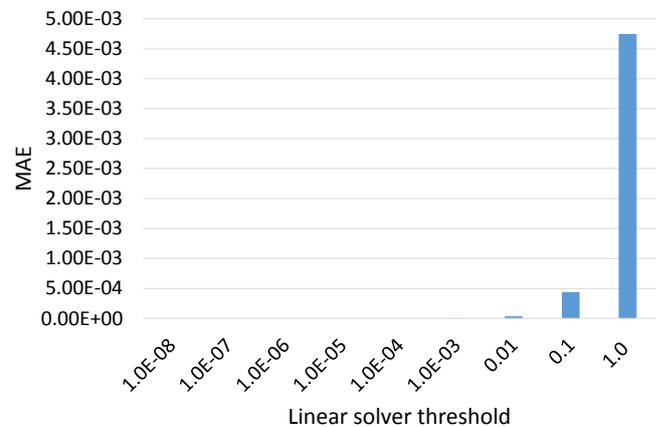


図 10 Plastic Can: 各閾値に対する平均絶対誤差 (MAE)

図 6, 図 8, 図 10, は, 図 5, 図 7, 図 9 に対する各アプリケーションモデルでの線形ソルバの閾値の変化に対する平均絶対誤差 (MAE) を示している. MAE は, 非常に小さい非線形解析の NR 法の閾値 1.0×10^{-9} での反復法と, 直接法を用いて得られた最終解との差分を比較することによって計算を行った. 予想通り, 線形ソルバの閾値が大きくなるに従って, MAE も増加していることが分かる. 工学的見地からは実測における精度も誤差を含むことを考慮し, MAE が 1.0×10^{-2} 未満であれば結果の精度は許容できると考えると, 次のように結論付けることができる.

- 線形ソルバの閾値は, 線形ソルバの反復回数と非線形解析の NR 法の反復回数が共に小さい場合の閾値が最速のシミュレーションとなる (半田モデルでは, 図 5 から, 閾値が 0.025 のときが最速となる).
- しかしながら, そのときの精度が許容できない場合は, 精度が許容できる範囲で最速となる閾値を選ばなければならない (半田モデルでは, 図 5 および図 6 から, 例えば閾値が 1.0×10^{-4} のときが, MAE において 1.0×10^{-3} 未満の精度を維持される中で最速となる).

4.3 線形ソルバの残差閾値最適化による高速化の見積もり

線形ソルバの残差閾値を最適化することで, 初期値の閾値を用いる場合に対して高速化が可能になると考えられる. 初期値の閾値と最速時の閾値を比較すると, Plastic Cylinder では閾値 0.1 使用時に 1.18 倍, Plastic Can では閾値 1.0×10^{-3} 使用時に 1.44 倍, 半田では閾値 0.25 使用時に 2.51 倍までの高速化が可能であると見積もられる. また, 初期値の閾値と, MAE が 1.0×10^{-3} 未満になる場合の最速時の閾値と比較すると, Plastic Cylinder では閾値 1.0×10^{-4} 使用時に 1.12 倍, Plastic Can では閾値 1.0×10^{-3} 使用時に 1.44 倍, 半田では閾値 1.0×10^{-4} 使用時に 1.65 倍までの高速化が可能であると見積もられる.

5. おわりに

非線形解析の高速化を実現するために, 一連のプロファイリングを行い, 非線形解析の最内ループである線形ソルバがメインの性能ボトルネックであることを明らかにした. また, その結果から, シミュレーションの設定変数の一つである線形ソルバの残差の閾値を変化させることにより, 閾値が小さ過ぎる場合には, 閾値を大きくしても精度

があまり劣化せずソルバの実行時間が短縮されることを確認した。また、平均絶対誤差 (MAE) が 1.0×10^{-3} 未満を満たす場合に 1.12~1.65 倍の高速化の可能性を見積もった。線形ソルバの閾値を自動的に最適化することで非線形解析を用いるアプリケーションの高速化が可能になると考えられる。

謝辞 FrontISTR の半田モデルのご提供および技術デスクセッション頂いた富士通アドバンステクノロジー株式会社の坂入 慎氏, 植田 晃氏, および富士通株式会社の稲垣和久氏に感謝する。

参考文献

- [1] Lu, C.: AI, Native Supercomputing and The Revival of Moore's Law, *CoRR*, Vol. abs/1705.05983 (online), available from <http://arxiv.org/abs/1705.05983> (2017).
- [2] Krizhevsky, A., Sutskever, I. and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25* (Pereira, F., Burges, C. J. C., Bottou, L. and Weinberger, K. Q., eds.), Curran Associates, Inc., pp. 1097-1105 (online), available from <http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf> (2012).
- [3] Feldman, M.: Fujitsu to Build Japan's Fastest Supercomputer, ABCI (オンライン), 入手先 <https://www.top500.org/news/fujitsu-to-build-japans-fastest-computer/> (参照 2019-04-16).
- [4] top500: Top 500 lists, top500 (online), available from <https://www.top500.org/lists/> (accessed 2019-05-08).
- [5] X. Guo, W. L. and Iorio, F.: Convolutional Neural Networks for Steady Flow Approximation, Technical report, KDD'16, Dan Francisco (2016).
- [6] Ichimura, T., Fujita, K., Yamaguchi, T., Naruse, A., Wells, J. C., Schulthess, T. C., Straatsma, T. P., Zimmer, C. J., Martinasso, M., Nakajima, K., Hori, M. and Madgededara, L.: A Fast Scalable Implicit Solver for Nonlinear Time-Evolution Earthquake City Problem on Low-Ordered Unstructured Finite Elements with Artificial Intelligence and Transprecision Computing, *SC18: International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 627-637 (2018).
- [7] S. K. Kim, S. Ames, J. L. C. Z. A. C. W. and Williams, D.: Massive Scale Deep Learning for Detecting Extreme Climate Events, Technical report, 7th International Workshop on Climate Informatics, Colorado (2017).
- [8] Alipanahi, B. and DeLong, A.: Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning, Technical report, Nature Biotechnology, vol. 33, pp. 831-838, 2015 (2015).
- [9] M. Paganini, L. d. O. and Nachman, B.: Accelerating Science with Generative Adversarial Networks: An Application to 3D Particle Showers in Multi-Layer Calorimeter, Technical report (2017).
- [10] G. B. Goh, N. O. H. and Vishnu, A.: Deep Learning for Computational Chemistry, Technical report (2017).
- [11] Jeff Adie, Yang Juntao, X. Z. S. S.: Deep Learning for Computational Science and Engineering, Technical Report 11, Nvidia (2017).
- [12] Jaber, A. K.: Solving Heat transfer equation by using feed forward neural networks, Technical report (2014).
- [13] M. Quito, C. M. and Saloma, C.: Solving N-Body problems with neural networks, Technical report, Physics Review, 2001. (2001).
- [14] D. Levy, M. D. H. and Sohl-Dickstein, J.: Generalizing Hamiltonian Monte Carlo with Neural Networks, Technical report (2017).
- [15] Carastan-Santos, D. and de Camargo, R. Y.: Obtaining Dynamic Scheduling Policies with Simulation and Machine Learning, Technical report (2017).
- [16] Gaussier, E. and Glesser, D.: Improving Backfilling by using Machine Learning to Predict Running Times, Technical report (2015).
- [17] Marathe, A.: Performance Modeling under Resource Constraints Using Deep Transfer Learning, Technical report (2017).
- [18] Yamada, K., Katagiri, T., Takizawa, H., Minami, K., Yokokawa, M., Nagai, T. and Ogino, M.: Preconditioner Auto-Tuning Using Deep Learning for Sparse Iterative Algorithms, pp. 257-262 (online), DOI: 10.1109/CAN-DARW.2018.00055 (2018).
- [19] Yeom, J., Thiagarajan, J. J., Bhatele, A., Bron-evetsky, G. and Kolev, T.: Data-Driven Performance Modeling of Linear Solvers for Sparse Matrices, *2016 7th International Workshop on Performance Modeling, Benchmarking and Simulation of High Performance Computer Systems (PMBS)*, pp. 32-42 (online), DOI: 10.1109/PMBS.2016.009 (2016).
- [20] FrontISTR: FrontISTR, FrontISTR Commons (online), available from <https://www.frontistr.com/> (accessed 2019-04-11).
- [21] Okuda, P.: Hiroshi Okuda / Professor / Division of Environmental Studies, The University of Tokyo (online), available from <https://www.k.u-tokyo.ac.jp/prose/person/hiroshi-okuda/hiroshi-okuda.htm/> (accessed 2019-04-10).
- [22] Chollet, F. et al.: Keras, <https://keras.io> (2015).
- [23] Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, (online), available from <http://arxiv.org/abs/1412.6980> (2015).