

原子力施設全体規模の構造解析に向けた要素毎有限要素接触解析手法: 階層化手法における加算順序の自動最適化

An element-by-element Finite Element Contact Analysis Method toward Structural Analyses of
an Entire Nuclear Facility: Automatic Optimization of Ordering to Hierarchy Technique

*鈴木 喜雄¹, 井原 遊², 奥田 洋司²

¹原子力機構, ²東京大学

原子力施設全体規模の構造解析に向けて実施した、並列かつ省メモリ解が可能な要素毎有限要素接触解析手法について報告する。これまでに提案した並列化手法[1]に対して、更なる性能向上のために Frontal 行列の加算順序を自動最適化する方法を提案し、有効であることを確認した。

キーワード：有限要素法、Multiple Front 法、並列化、オーダリング

1. 緒言

原子力施設全体規模の構造解析に向けて、大規模複雑構造物を構成している複数の部品を個別にメッシュ生成した場合に、部品間でメッシュが不整合メッシュな場合でも解析可能な、要素毎有限要素接触解析手法の研究開発を進めている。ここで、行列解法に直接法の一環である Multiple Front 法[2,3]を用いている。Multiple Front 法では、全体剛性行列を作成することなく、各部品に対応する部分剛性行列を並列処理できる。一方、これまで、部品間の接触面に対応する行列を解く処理が並列化されておらず、ボトルネックとなる場合があった。そこで、行列二つずつを同時に加算し、生成される行列 (Frontal 行列) の分解処理を並列に行い、さらにその部品群の行列二つずつを同時に加算し、新たに生成される行列を並列に分解処理するという、階層的に処理を進める並列化手法 (階層化手法) を提案した[1]。また、Frontal 行列のサイズを事前に求めるシンボリック分解と Frontal 行列の加算順序を階層化手法に最適化することで、更なる性能向上を確認した。しかし、加算順序の変更は試行錯誤し手動で行っていたため、本研究では、自動的に最適化する手法を提案する。

2. 加算順序の自動最適化

加算順序を決定するため、従来方法では、 $P_i = -W_1 ngain_i + W_2 d(i, e) - W_3 nadj_i$ が大きな順に部品 i を順次配列していく[4]。ここで、 $ngain_i$ 、 $d(i, e)$ 、 $nadj_i$ はそれぞれ、「部品 i が持ち込む変数の数-消去される数」、「『エンド部品 (部品間距離が最長となる部品の片端)』と『部品 i 』間の距離」、「部品 i の未配列隣接部品数」である。 W_n は重みで、文献[2]では、 $(W_1, W_2, W_3) = (10, 5, 1)$ が推奨されている。しかしながら、この方法は階層化手法には必ずしも適さない。そこで、加算順序の自動最適化に向けて、第一に、従来手法の重みの値を変更すること (①) を考えた。ここで、既配列部品に近い部品をより選び易くするため、第 3 項の影響がより強くなる重みの値 $(W_1, W_2, W_3) = (1, 0, 10)$ を試した。第二に、各階層での加算時の「変数の個数」を小さくすること (②) を考えた。そのため、各階層 $n (= 1, 2, \dots, m)$ (m は全階層数-1) において 2^{n-1} 個の部品を含む j 個の部品群を、 $P_j^n = -ngain_j$ が大きな順に配列するという過程を階層毎に繰り返し、更に、低階層での再構成のために、第 k 階層 (今回は $k = 3$ を使用) で $\sum_{n=1}^k P_j^n$ が大きな順に再配列するという手法を試した。

3. 結果

図 1 に示すモデルを用いて性能評価を実施した。ここで、モデルを複数の部品に分割し、各部品を各コアに分担させるように並列化した (現状は、2、4、8、16、32 並列の 5 通り)。手動で加算順序を最適化した場合の結果と比較することにより、①の場合には完全には自動化できず、問題サイズに依存して W_3 の大きさを調整しなければ最適な加算順序を得られないこと、②の場合には、手動の場合と同等の処理時間となることで、加算順序の自動最適化が行えることが分かった。

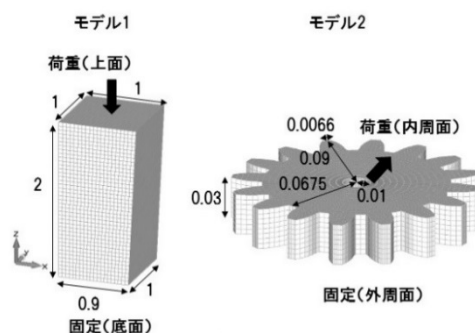


図1

参考文献

- [1] “原子力施設全体規模の構造解析に向けた要素毎有限要素接触解析手法:性能改善のための並列化手法開発”, 鈴木喜雄, 日本原子力学会 2017 年秋の大会 (2017).
- [2] “The use of multiple front in Gaussian elimination”, I. S. Duff and J.A. Scott, RAL-94-040 (1994).
- [3] HSL, a collection of Fortran codes for large-scale scientific computation, See <http://www.hsl.rl.ac.uk/>
- [4] “On ordering elements for a frontal solver”, J. A. Scott, Commun. Numer. Meth. Engng, 15, 309-323 (1999).

*Yoshio Suzuki¹, Yu Ihara² and Hiroshi Okuda²

¹JAEA, ²The University of Tokyo.