

## 4.4.2 構造解析

### (1) 現在行われている課題

現在、産業界において製品およびその生産や維持工程を設計し評価する際に、シミュレーション技術が不可欠なものとなりつつある。例えば自動車の騒音・振動解析や衝突解析、航空機や船舶の構造健全性評価、大規模プラントや建物・橋梁や高速道路などの耐震解析、更にものづくりにおける生産過程、特に塑性加工やプレス成型を再現するためのバーチャルマニファクチャリングなど、多種多様なものづくり・設計シーンでのシミュレーション技術の導入、すなわち CAE の普及が進んでいる。有限要素法（FEM : Finite Element Method）[1]に基づく構造解析ソルバーは対象となる人工物・構造物の機能や応答性、安全性、健全性などを評価するためのツールであり、こういった CAE を実現するための最重要コンポーネントとして、今やものづくりにとって不可欠なものとなっている。

FEM ベースの構造解析ソルバーは、その数値計算スキームおよびこれを用いたビジネスモデルの特性により、狭い範囲の課題や特定の問題を解くための専用コードよりも、多数の機能を集積した汎用目的のコードが一般的である。こういった汎用コードは、機械・電気・化学・建築土木をはじめ、船舶海洋・航空宇宙・原子力、そして現在では医療・環境など驚くほどさまざまな分野で用いられている。また、主に大企業や研究所を中心に、これらの構造シミュレーションにおける解析の高速化やモデルの大規模化・詳細化を目的として、PC クラスタや中規模スパコンなどの HPC 環境が導入されつつある。

残念ながら現状では、この構造解析分野の多くで NASTRAN、ABAQUS や LS-Dyna など欧米発の商用ソフトウェアが用いられている。しかしながら、これらの多くでは数万コアオーダーの超並列技術の導入が遅れており、これが逆に産業界でのさらなる HPC 技術の導入を阻害している。一方、最新 HPC 技術の存在を前提に、超並列技術の導入に関する新たな技術開発が国内でも現在進行中である。今後のペタおよびエクサスケールのスパコンの登場と普及により、国産ソフトウェアにも新たな普及の道が開ける可能性があるのみならず、シミュレーションを通じてわが国のものづくり技術の発展に寄与すると考えられる。

### (2) 長期的目標

FEM による構造シミュレーションはものづくり・設計現場においてすでに数十年の長い歴史を有する。しかしながらその具体的内容は、大型計算機における弾性応力解析、弾塑性解析、ベクトルスパコンを用いた衝突解析や振動解析、UNIX サーバやワークステーションでの設計者向け CAE、そして PC クラスタでの流体構造連成解析など、時代時代のニーズやその時点で台頭してきた新しい計算機アーキテクチャとともにさまざまに変遷してきた。

これまでの過程を見る限り、この分野では、例えば数年前にスパコンで行われていた解析が、現在は部門サーバ上で行われており、そしてその数年後にはワークステーション上で行われるといった形での、いわゆる下方展開が実現されてきた。すなわち、新規のニーズとそれに応えるための新規コードの多くは、その当時における最先端の HPC 技術を用いてまずスパコン上で開発・利用されてきたといえる。実際、10 年前に TOP500 クラスであったテラスケールの

PC クラスタが現在多くの大企業で導入・利用されており、この流れは今後も続くと思われる。例えば、エクサが登場する時点ではペタクラスマシンが普及しており、現時点において、ペタクラスマシンを用いることでようやく実現可能になるような課題が、数年後には多くのセクションで普通に取り扱われるようになるものと考えられる。したがって、エクサクラスマシンにおいて、現時点においてきわめて困難な課題を扱うことにより、近未来の産業界への普及が見えてくるはずである。

例えば、主な産業分野における部品数は、一般的な家電製品で数千、自動車や重機では数万、そして航空宇宙や大規模プラント業界ではこれが百万のオーダーである。従来は部品単位での構造解析が主流だったものが、近年の傾向として、それらをアセンブリした構造物全体のまるごと解析が可能となりつつある。これにより大規模解析のニーズは飛躍的に増大している。

また、近年産業界において流体構造解析のニーズが増大しつつあるが、構造解析ソルバーに対して近年急速に普及しつつある熱流体解析ソルバーを連成させることでこれが実現できる。この場合、高精度化のために流体側の解像度が細くなるにつれ、対応する構造側も詳細化・大規模化していく傾向がある。

加えて、先に医療やバイオメカニクス分野において別途示されているように、この分野において骨や筋肉、臓器などの生体構造をシミュレーションするニーズが高まりつつある。これらの対象では非常に複雑な形状モデルと非線形性の強い材料物性を扱う必要があり、モデルのさらなる大規模化と計算時間の増大が不可避となっている。

こうした課題に対する現実的な時間でのシミュレーション実施のため、HPC 技術が必須である。

### (3) 次世代に解決すべき課題

さきに述べたように大規模構造解析のニーズとしては、すでに産業界の広い分野においてさまざまなものが顕在化しつつあるが、ここで一例として挙げるのは、原子力プラントの耐震解析である。鉄筋コンクリート壁、配管、圧力容器、燃料集合体などすべてがソリッド要素で表現されている。解析規模としては、原子炉圧力容器単体でも数十億自由度を要し、配管や発電機器を含むプラント全体と付近の地盤を合わせて数百億自由度と見積もられる。また、解析は弾塑性の動解析であり、時間方向および非線形計算の繰り返しからなる数万から数十万ステップの計算が必要となる。

「京」コンピュータをもってしても、自由度や計算ステップに関して各々これらの 10 分の 1 程度のものを扱うのが限界となっているようであり、現時点では、圧力容器や少数の配管群など、特定箇所にはフォーカスした解析を行うか、あるいはアドホックなモデル化や線形解析に頼らざるを得ない。今後の HPC 技術の進展により、プラント全体の計算が可能になるのみならず、地球物理・大気海洋分野で行われている地震シミュレーションや津波シミュレーションとの連携により、中越沖地震や東日本大震災における原子力プラントのより厳密かつ高精度なシミュレーションが可能となると思われる。

#### (4) 課題を解決するために必要なアプリケーション群

有限要素法を用いた構造解析の計算手法や解析の分類は、線形弾性解析、弾塑性解析、振動解析、熱応力解析、衝撃解析、塑性加工解析など多岐にわたる。ここではこれらを、HPC 技術との関連という見地から、連立一次方程式を解かないもの（陽解法系）と、連立一次方程式を解くもの（陰解法系）の二つに分類して議論する。これらはどちらも重要であり、個々の問題の特性や要求精度に合わせて使い分けられる。

前者の場合、各有限要素の単位で得られる情報だけを用いた行列ベクトル積が主要な演算パターンとなり、計算時間の大半は要素剛性行列評価や応力積分などの要素単位での計算処理に割かれる。

一方、後者の場合、まず各要素の要素剛性行列を用いて、全体剛性行列と呼ばれる比較的大サイズの疎行列を組み上げる。非定常問題における陰解法ソルバーの場合、これに関する連立一次方程式を時間ステップあるいは非線形ステップごとに解くことになる。なおこの手順は、静解析、固有値解析含め、解析ソルバー内に連立一次方程式の解法が組み込まれているもの全般についてあてはまる。

数千万から数億自由度クラスの大規模解析の場合には、もっぱら反復型解法の系統が用いられる。その際、大規模な連立一次方程式を解くための超並列アーキテクチャ向けアルゴリズムに関して、並列反復法ソルバーおよび領域分割法の二つが考えられる。

以下、エクサスケール級計算によって可能となる超大規模問題を想定したうえで、エクサスケール級の HPC アーキテクチャ上で高い性能を発揮できると思われる計算スキームについていくつか取り上げる。

##### (i) 連立一次方程式を解かない場合（陽解法系）

陽解法は、衝突解析や塑性加工解析のように、非定常でしかもきわめて短時間に生じる現象を表現する場合に用いられる。例えば、商用ソフトで言えば LS-Dyna や PAM-CRASH などが有名である。本手法に基づくソルバーの場合、もっぱら個々の有限要素レベルでの比較的小規模な行列やテンソルに関する演算を数百万ステップ以上繰り返すことになる。FEM では一般にこのような演算パターンを Element-by-element (EBE) 型オペレーションと呼ぶ。コアごとの性能に関して、EBE 演算では要素ごとの限られた情報だけを用いて比較的大量の演算を行うため、キャッシュやレジスタを有効に利用することが可能であり、コンパイラによっては非常に高い性能を発揮する。B/F(Byte/FLOP)値が低い計算機でも有効である。並列性に関しては、要素単位での計算が主なのでその並列化は容易である。ただし、衝撃解析や塑性加工解析では要素間の接触を扱うことが多く、接触判定を行うため検索処理の並列化に注意する必要がある。大規模アセンブリ構造物の取り扱いでは部品間の結合が重要であり、この処理方法に関し、活発な研究開発が行われている。

##### (ii) 連立一次方程式を解く場合（陰解法）：並列反復法ソルバー

並列反復法は、商用コードではあまり例がないが、国産フリーソフトの GeoFEM や FrontISTR などをはじめいくつかのコードで採用されている。並列反復法ソルバー[2]の場合、対称正定値であれば CG 法、非対称であれば Bi-CGSTAB あるいは GMRES 法などにより、全

体剛性行列に関する連立一次方程式を解く。構造解析では一部の非線形問題を除き、対称正定値行列を扱うことが多い。一方、構造物の大型化あるいは軽量化により、梁や板状の薄い形状を扱うことも多く、この場合係数行列の条件数は悪化する。したがって、強力な前処理が必要とされることが多い。並列反復法ソルバーでは、行列ベクトル積演算が繰り返される。有限要素法の場合、これは三種類の方法により実装することができる。まず、係数行列である全体剛性行列を要素剛性行列から組み上げ、その非ゼロ成分をメモリ上に保存する、非ゼロソルバー。次に、係数行列の全体化は行わず、要素剛性行列の形でメモリ上に保存する、EBE ソルバー。最後に、要素剛性行列すらメモリ上に保存せず、行列ベクトル積が必要とされる都度に評価する EBE-MSF (Matrix Storage-Free) ソルバー。以下、係数行列を対称とし、CG ソルバーにおいてこの 3 種類のソルバー実装の性能見積もりを行う。なお、行列ベクトル積部分の並列化は比較的容易であることから、通信パターンは隣接間が主なものとなる。

非ゼロソルバーの場合、全体剛性行列とベクトルとの積であることから、メモリアクセスは主に行列成分のみとなる。行列の 1 成分をメモリから読み出すごとに、対称性を考慮して 4 個の浮動小数点演算が必要となる。ただし、行列は疎でしかも非構造格子からのものであるため、メモリアクセスは不規則でベクトル化の効率は低く、しばしば節点単位でのレジスタブロッキングが不可欠となる。また CPU アーキテクチャおよびネットワーク通信の双方におけるハードウェア支援のギャザースキャット機構は性能向上に役立つ。

次に EBE ソルバーの場合、メモリ上に記憶された個々の要素剛性行列とベクトルとの積結果を組み上げていくことになる。要素剛性行列は密で対称であるため、非ゼロソルバーと同様、行列の 1 成分ごとに 4 個の浮動小数点演算が必要となる。EBE ソルバーは非ゼロソルバーより記憶量、演算量とも多いが、メモリアクセスパターンは比較的規則的で性能を出しやすい。

最後に、EBE-MSF ソルバーの場合、これは陽解法ソルバーの場合と同様、EBE 演算がメインとなる。B/F 値の低い計算機でも有効である。もし B/F 値が極端に低い場合、本ソルバーは主にメッシュデータのみをメモリから読み出すため省メモリであり、しかも他のソルバーよりメモリアクセス量が少なくなるために結果的に最速ソルバーとなる可能性がある。

### (iii) 連立一次方程式を解く場合（陰解法）：領域分割法

領域分割法 (DDM : Domain Decomposition Method) [3-5]は、国産フリーソフト ADVENTURE や米国の Salinas などで採用されている。先に述べた線形代数レベルでの並列化を行った並列反復法ソルバーと異なり、領域分割法は偏微分方程式レベルでの並列化手法である。まず全体領域を複数の部分領域に分割する。各部分領域に適当な境界条件を与えて独立に解き、領域間境界条件が落ち着いてくるまで反復的にこれを繰り返す。したがって、この方法は反復解法の一つである。構造問題においては、並列反復法ソルバーの場合と同様、強力な前処理が不可欠となる。DDM の場合、FETI (Finite Element Tearing and Interconnecting) や BDD (Balancing Domain Decomposition) などのマルチグリッド型前処理が多用されている。DDM の主要な演算パターンは部分領域ごとの有限要素解析（ローカルソルバー）である。DDM の並列化はローカルソルバーを単位として、その並列化は容易であり、通信パターンは隣接間がメインとなる。ローカルソルバー実装にはいくつかの方法が考えられる。まず一つ目は、部分領域ごとの剛性行列について、これをまず直接法ソルバーを用いて行列分解し、スカイライン形式などでメモリ上



に保存しておく。その上で DDM 反復ごとに前進後退代入を行うというものである。二つ目は、部分領域ごとに反復法ソルバーを用いて部分領域問題を解くというものである。このとき、メモリ上にはほとんどデータを記憶しない。三つ目は、部分領域ごとに対応する密行列をあらかじめ作成しておき、これらとベクトルとの積を繰り返すというものである。これは並列反復法ソルバーにおける EBE ソルバーのように、ローカル Schur 補元行列と呼ばれる、部分領域ごとの剛性行列を、静的縮約により陽に導出してメモリ上に保存しておくものである。

最初の直接法ソルバーアプローチでは、行列分解されたデータに対する前進後退代入演算が主である。この場合、行列の 1 成分を読み出すごとに 2 個の浮動小数点演算が必要となる。二つ目の反復法ソルバーアプローチでは、メモリからメッシュ情報を読んでキャッシュ上で反復法ソルバーを起動する。大量の演算が L2 あるいはラストレベルキャッシュ上で行われることになる。並列反復法ソルバーで説明した非ゼロ、EBE、あるいは EBE-MSF ソルバーのいずれも利用可能である。なお、前処理の導入により反復回数を削減したい場合、前処理行列をあらかじめ作成してメモリ上にストアしておくことも考えられる。最後のローカル Schur 補元行列保存の場合、EBE ソルバーと同様、対称な密行列とベクトルの積であり、1 成分ごとに 4 浮動小数点演算を要する。なお、部分領域ごとに静的縮約によりローカル Schur 補元行列を作成する際に、行列行列積を中心とした大量の浮動小数点演算が行われる。

これらをまとめると、エクサスケール級計算機のアーキテクチャ分類について、「メモリバランス」あるいは「リファレンス」型が理想である。ただし、陽解法ソルバー、EBE-MSF 型反復法ソルバーや DDM の場合には、「演算重視」あるいは「メモリ容量削減」でもある程度対応は可能であると思われる。なお、それぞれについて最適な実装形態を用意する必要がある。

課題	要求性能 (PFLOPS)	要求メモリ バンド幅 (PB/s)	メモリ量/ ケース (PB)	ストレージ 量/ ケース (PB)	計算時間/ ケース (hour)	ケース 数	総演算量 (EFLOP)	概要と計算手法	問題規模	備考
衝突解析	210	5	1	100	24	10	180,000	有限要素法(陽解法)	10 <sup>11</sup> 節点	
弾塑性解析	21	5	1	1	24	10	18,000	有限要素法(陰解法)	10 <sup>10</sup> 節点	
丸ごと詳細解析	210	50	10	10	24	10	180,000	有限要素法(陰解法)	10 <sup>11</sup> 節点	

※本見積もりは未だ精査中である。より精度の高い数値は Web 版 (→「1.2. 本文書の構成」) を参照のこと。

## (5) 他分野との連携

構造解析や有限要素法は、自動車や鉄道、航空機分野で求められている熱流体や電磁場ソルバーとの連成、バイオメカニクスにおけるソフトマテリアルの非線形解析、ナノ材料分野における均質化法などを介したマルチスケール解析、あるいは、地球物理・大気海洋分野における自然災害に対する構造物の健全性の評価など、大規模シミュレーション技術全般を通して不可欠なコンポーネントとなりつつある。

#### (6) 公的な意見交換とそのサマリー

主に機械学会や計算工学会、IACM 関連の計算力学コミュニティを中心に、エクス時代における構造解析の位置づけに関してディスカッションと意見交換が行われている。

#### (7) ロードマップ

構造解析分野のロードマップを図 4.4.2-1 に示す。

年代 課題	2012 ～	2014 ～	2016 ～	2018 ～	2020 ～	2022 ～
解析技術 研究						
	大規模構造物の まるごと解析技術の開発			熱・流体効果の取込みによる まるごと詳細解析技術の開発		

図 4.4.2-1 構造解析ロードマップ

#### 参考文献

- [1] Zienkiewicz O.C., Taylor R.L, The Finite Element Method, 5th ed., Oxford (2000).
- [2] Saad, Y., Iterative methods for sparse linear systems, PWS publishing company (1996).
- [3] Smith, B., Bjorstad, P.E., Gropp, W.D., Domain decomposition : Parallel multilevel methods for elliptic partial differential equations, Cambridge University Press (1996).
- [4] Toselli, A., Widlund, O.B., Domain decomposition methods : Algorithms and theory, Springer (2004).
- [5] Tarek. P.A. Mathew, Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations, Springer (2008).