

論文の内容の要旨

論文題目 An Object-Oriented Framework for Scientific Computations
- advantages in parallel computational fluid dynamics -

科学計算のためのオブジェクト指向フレームワーク
- 並列流体計算における利点 -

氏名 太田 高志

今日、並列計算は数値流体力学(Computational Fluid Dynamics: CFD)にとって重要な手段となり、将来的には、中、大規模計算のための主要な計算環境となることが予想される。並列計算のためのプログラム開発に関しては、様々な言語、コンパイラ、ライブラリが存在し、それらを用いたプログラムの「並列化」の技術というものが考案されている。しかしながら、このような状況にあつて並列計算のためのプログラム開発は単なる作業としての捉えられ方が一般的であり、プログラムの設計に関する方法論が存在していないように思われる。個々のプログラムに対してそれぞれに対応した並列化をその都度行わなければならない等、のプログラムの並列化に関する困難さは、単なる作業上の問題以上のものとして考えられなければならない。並列計算機の計算能力が十分活かされるべき計算対象は大きな計算処理能力を必要とするものとして単純に格子数を増加したものに留まらず、より現実的な問題に対応するための複雑なモデルを持つものが考えられる。そのような計算が、元からのアルゴリズムの複雑さに加えて並列化の困難さによって実現が困難となることは、CFD、またはCFDによる研究にとって大きな障害である。同時に、これまで並列計算に関して、その計算自体ばかりが主題とされてきたように思われるが、数値的解析プロセスにおいてその環境の恩恵を十分に甘受するにはプロセス全体で並列計算に対する考慮を行う

ことが重要である。これまで困難であった計算を行うのに十分な計算能力は刻々と向上されており、それを有効に使う手段の確立が望まれる。

一方、CFDにおける研究の成果は、解法のスキームや、アルゴリズム、またはそれらによる計算結果という形で提示されてきた。それらは主に論文を通して参照され、従って、新しいスキーム等を検証するには、論文の理論式等からプログラムを再構成する必要がある。これは成果の共有のありかたとしては非効率であると同時に不正確でもある。同様に、計算で使用される格子データや条件等は論文にも記載されないことが多く、自身の研究との比較を行うことが簡単ではない。また、同じ研究所やグループ内においても、個々の研究者がそれぞれ互換性なくプログラムを開発していることが多く見聞きされる。CFD研究のさらなる推進のためには成果の検証と再利用を容易にする共通指針が必要である。

航空学における空力現象の解析に対してCFDの果たす役割の重要性を鑑みたときに、以上に述べたような諸問題の解決を図り、研究者が本来の目的である計算スキームの開発や物理現象の解析に集中出来るようにすることは非常に重要な要件である。これらの状況を背景として、本論文においては上記の問題への一つの解決を提示するため並列数値流体計算プログラムの設計指針を提案し、それに基づくプログラミングフレームワークを設計する。フレームワークはプログラム構造を提供することによってプログラム設計の指針を提示し、また、並列プログラムの作成を容易にする手段を与える。同時にそれはプログラム作成のための型紙（テンプレート）として作用し、それに沿って書かれたプログラム間の互換性、再利用性を保証する共通基盤となる。

本論文ではプログラムの基本設計として以下のようなものを提案する。

- 全体として並列性を有する計算スキーム
- 計算と並列処理の分離
- 計算、並列処理等から独立したデータのモジュール化

全体としての並列性は、何らかの手法で計算の負荷全体を分割することにより達成される。このアプローチにより、逐次計算は分割数が一つの場合として実現する。すなわち、同じ取り扱いの中での単なる条件の違いとして逐次と並列計算を扱うことが出来る。計算と並列処理の分離、及び、データモジュールの独立を達成するためには、それぞれの設計に対して、負荷の分割数に依存しない一定のインターフェースを持つような抽象化が要求される。

以上のような指針によるプログラムを実現するためにプログラミングフレームワー

クを設計した。フレームワークは以下を目標とする。

- 並列化のための部品を提供する汎用なライブラリではない
- プログラム全般の構造をCFDコード主要要素の結びつきにより与える
- オブジェクト指向による設計を行う

フレームワークは、データオブジェクトの設置とその使用のための仕組みを核として、様々なモジュール（オブジェクト）と、それらの関連を規定している。上位のモジュールとして、流れ場、計算スキーム、境界条件、等のCFDでの主要な概念となるものを用意し、個々の実装、それらの組み合わせによる全体のプログラムの構成を容易にしている。

本フレームワークはここで述べた設計の結果、計算と並列処理の分離を達成する。これは、計算スキームのコードの記述において並列化のための処理を考慮せずに行えることを意味しており、目的である並列プログラム作成の容易化を実現する。また、分離したモジュールによる設計によって、個々の変更を他の部分へ影響を与えずに行うことが可能となり、殆どの部分の可搬性を保ちながら様々な機能拡張を追加出来るという利点をもたらす。計算スキームのモジュールの可搬性としては、プログラム作成における並列処理のためのモジュールとの組み合わせの有無で、並列、逐次計算のどちらでもそのままの形で使用することが出来る。従って、計算スキームの研究等で、開発段階では応答性の高い個人的なワークステーションなどの環境により行い、完成した後、同じコードをそのまま用いて並列計算機で大規模な計算を実行する、という従来の並列化とは異なるより効率的なプロセスが可能となる。プログラム開発の基盤としては、共通のインフラとしての可能性と共に、同様の仕組みで格子生成、可視化、等を含めた全解析プロセスの統一した方法論を与える。以上のように、本論文は並列計算のためのプログラム開発において通常行われている「並列化」とは異なる方法論を提案し、そのための具体的な実現の一例を示した。また、このようなアプローチによって解析プロセス全体がどのように構成されるか、その利点、重要性を含めた検証を与えている。

本論文第一章では、数値流体力学計算における並列計算の現状と、その一般的なアプローチがかかえる問題点を概観し、それらに対する本研究の背景、目的、意義が述べられている。また、他の並列流体計算およびオブジェクト指向による並列計算に関する研究との比較においての当研究の独自性が明確にされる。第二章では、本フレームワーク設計の基盤となる並列プログラム設計の基本概念についての説明がされる。その主な

アプローチは、計算スキームの並列性を前提として、計算で使用されるデータ類とそれを対象とする処理をそれぞれ独立したモジュール（オブジェクト）とすることであり、その設計のための適切な抽象概念の抽出についてが述べられる。第三章においては、上記設計方針の具体的な実現手段としてのフレームワークについて具体的な記述が行われる。まず、フレームワーク設計の必要性に続き、その性格を定義する設計指針が述べられる。つづいて、フレームワークの個別の局面についての設計の詳細な解説がある。第四章では、設計されたフレームワークの利用について解説される。フレームワークによりどのようにプログラミングを行うかが述べられる。また、具体例としてフレームワークを利用して構築された流体計算プログラムの並列実行による計算例、性能評価が紹介される。第五章では、本論文におけるアプローチについて、その利用に際して多様な側面に対しての議論がなされる。フレームワークを利用した独自の計算スキーム等のプログラミングについて、コードの再利用性、可搬性などの問題、また、フレームワークにより構築される計算コードの計算性能についてオブジェクト指向による設計を採用した上での対策が検討される。また、計算自体の並列実行だけではなく、全数値解析プロセスを構成する、格子生成、可視化を含めた統合的な計算環境に対する対応や、提言を行う。第六章は、総括として、本論文で提案される設計、フレームワークにより並列流体計算にもたらされる利点が述べられ、意義が強調される。付録 A として、通常の並列化アプローチによるプログラムと本フレームワークを利用した場合のプログラムを、それぞれの場合に要するプログラミング・コストの比較として掲載している。付録 B にはオブジェクト指向による計算性能向上への試みの一つとして論文内でとりあげている C++ の技術を利用した性能測定用のプログラムを、他の比較用のプログラムと併せて掲載している。