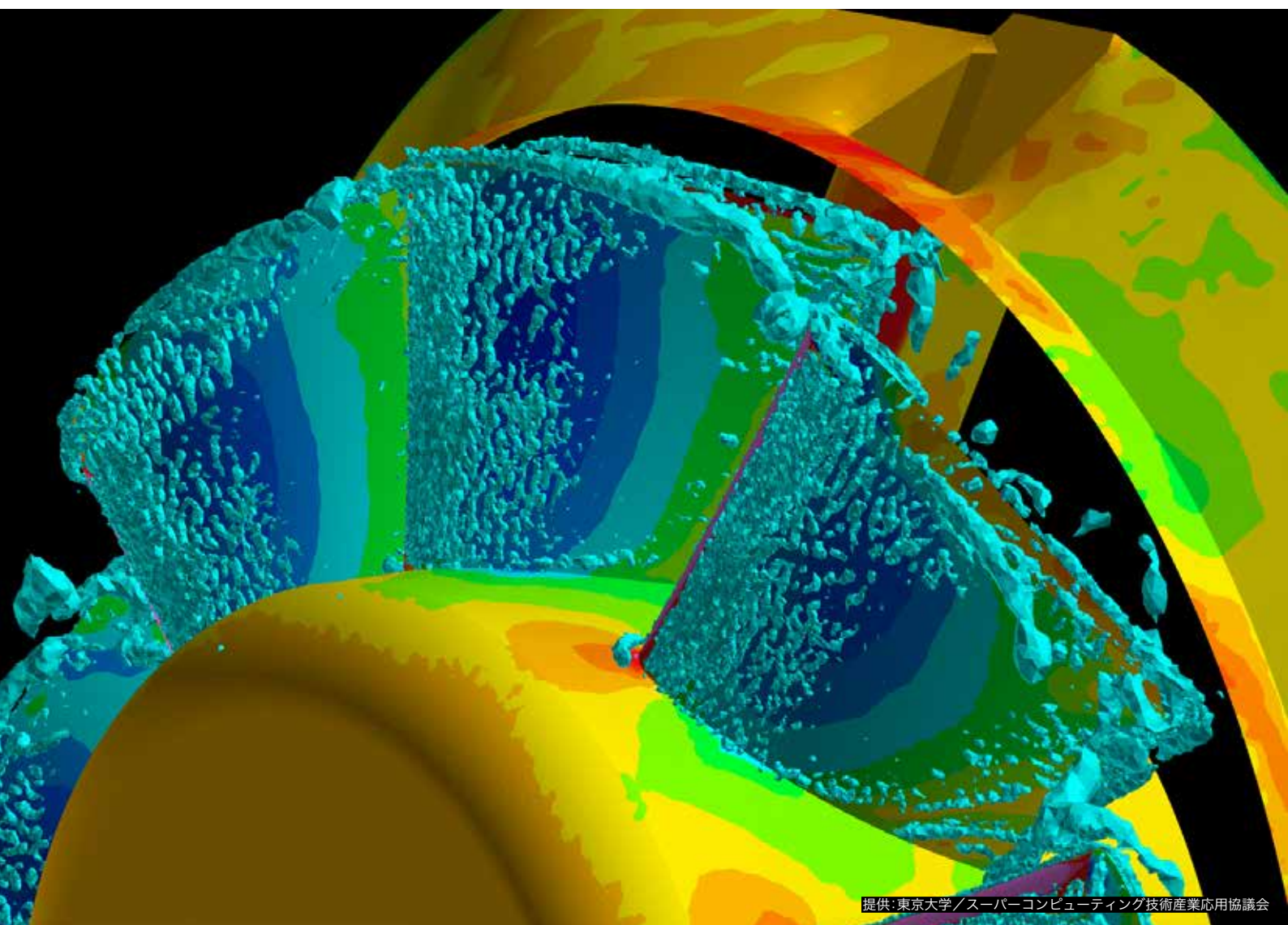


ものづくりにHPCを活用するためのツールとケーススタディー

計 算 工 学 ナ ビ

計算工学ナビ・ニュースレター2018年春号



提供: 東京大学 / スーパーコンピューティング技術産業応用協議会

設計を革新する多目的設計探索・
高速計算技術の研究開発

JAXA 大山 聖 / 九州大学 小野謙二 / 九州大学 飯塚幹夫 / 理研 AICS 川鍋友宏

VOL. **14**

設計を革新する多目的設計探査・ 高速計算技術の研究開発【前編】

最適な設計パラメータを最適化計算によって導き出す多目的設計探査の技術は製品開発を行う上で強力な武器となる力を秘めています。この技術をより効果的で、使いやすいものに育てるため、宇宙航空研究開発機構(JAXA)の大山聖准教授は継続的に研究開発を進めてきました。ポスト「京」に向けて新たなステップに入ったプロジェクト(重点課題8 サブ課題A)の状況と展望を聞きます。



宇宙航空研究開発機構
宇宙科学研究所
大山 聖 准教授

プロジェクト名には「多目的設計探査」と「高速計算技術」という2つのキーワードが含まれています。これらはどういう関係なのでしょうか？

多目的設計探査を産業界で普及させるためには、最適化アルゴリズムの開発だけでなく、コンピュータシミュレーションの評価回数をたくさん稼ぐための高速化も一緒にやらないといけない課題です。また、研究者だけでなく現場の設計者が使えるような形で技術を提供することも必要です。我々のプロジェクトでは、最適化手法、高速化、ワークフローという3つの要素の研究開発を同時並行で進めています。

「京」からポスト「京」へと計算環境が変化することで、多目的設計探査はどう変わるのでしょうか？

「京」では、車両構造の最適化(マツダ株式会社)、高速鉄道車両の空力音響最適化(東海旅客鉄道株式会社)、タイヤの音響最適化(横浜ゴム株式会社)、ロケットの射点形状の最適化(JAXA)といった問題に取り組みました。こうした大規模な数値シミュレーションを必要とする問題であっても、最適化手法が有効であることを示すことができたのが「京」での成果になります。

一方で産業界の最適化問題を実際に解いてみて新たにわかった課題もいくつかありました。ひとつはターンアラウンドタイムの長さ。計算を始めてから結果が出るまでの期間が非常に長かったのです。もうひとつは、制約条件が厳しい問題で適切な最適解を得ることが難しいという課題です。

ターンアラウンドタイムと制約条件に関する課題について、実際にあった事例を教えてください。

マツダと取り組んだ構造設計の最適化を例にとりましょう。まずターンアラウンドタイムについてですが、進化計算アルゴリズムの1世代の計算に1週間という時間がかかっていました。30世代の計算を実施しようとするすると30週間かかってしまうことになります。これでは開発スケジュールにマッチしません。ポスト「京」の大規模な計算資源をうまく利用して、できれば1/5くらいまで短縮するのが目標です。

制約条件については、設計パラメータが約200あって、それをランダムに振った場合、全制約条件を満たす解は1個も出てこないということがありました。今回はこのような複雑な問題も効率的に解けるようにします。進化計算の早い世代、たとえば3世代目くらいで最初の実行可能解を見つけて、制約条件を満たす解も積極的に探しにいくようなアルゴリズムにしたいと考えています。

ここで言う「世代」の意味を教えてください。

我々が使っている手法は多目的進化計算と呼ばれます。クルマを例にとると、まず最初にいろんな形のクルマをたとえば100個作ります。これが設計候補の初期集団です。次に数値シミュレーションを使って空気抵抗や構造重量といった値を評価して良いものを半分くらい残します。残ったものを元に次の世代の集団を

作り、また評価する。この繰り返しです。昆虫のように毎年世代が変わる生き物を考えていただければわかりやすいと思います。

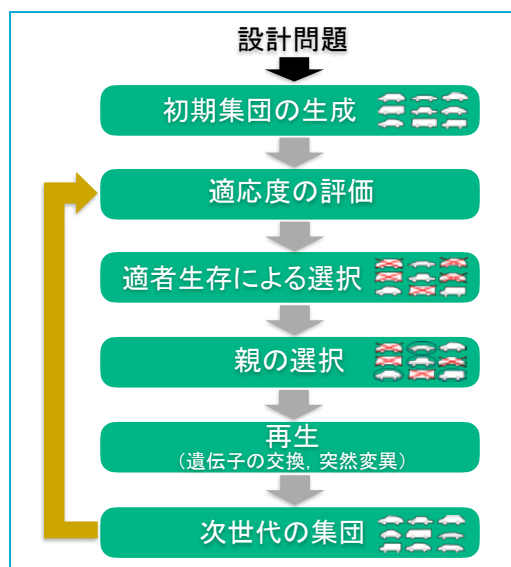
【図1】

たくさんのプロセッサを持つポスト「京」なら各世代の集団サイズを大きくとれますので、それを利用して少ない世代数で最適解が得られるアルゴリズムを作ろうとしています。【図2】

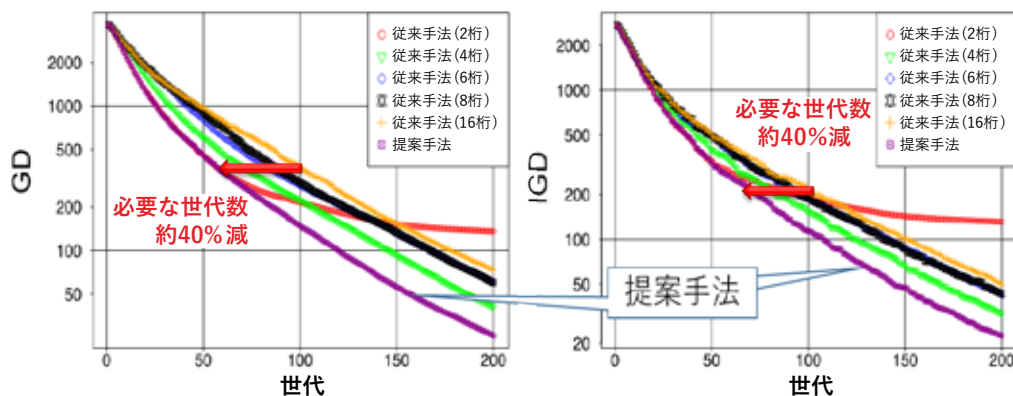
ポスト「京」に向けて、他のチームとの連携も深まっていますね。

2018年度はサブ課題Bと連携して自動車の設計最適化について実証研究を集中的に進めます。その翌年度はサブ課題Cとターボ機械に取り組む予定です。【図3】

ただし、まだポスト「京」は使えませんので、当面は「京」を使って、車両の空力設計に取り組む予定です。ポスト「京」が動き始めた後は、風の変化に対するロバ



【図1】多目的進化計算の過程



【図2】ターンアラウンドタイムの削減
図中のGDはパレート最適解の収束性を評価する指標で、小さければ小さいほど優れたパレート最適解が得られていることを示す。IGDはパレート最適解の収束性と多様性を評価する指標で、同様に小さいほど優れた解が得られていることを示す。従来手法では100世代必要だったパレート最適解を60世代程度で得ることができており、ターンアラウンドタイムが約40%削減できていることを示している。

ト性であるとか、デザイン、美しさといった要素も加味したより高度な最適化を実施したいと考えているところです。

美しさをどう評価するのか興味があります。

この部分はポスト「京」プロジェクトではなくマツダの研究となりますが、人間が見たときの美しさを定量的に評価する手法を開発しています。ポスト「京」が動き出す2、3年後にはこの手法も完成する予定と聞いていますので、うまく組み合わせて、美しさを保ちながらその他の性能を向上させていくな最適化を実現したいですね。

「京」のプロジェクトでは多目的進化計算のためのソフトウェアCHEETAHを開発し、公開しました。

CHEETAHは株式会社ヴァイナスのiDIOS^[1]という商用パッケージの一部として利用可能になる予定です。多目的設計探索の活用を考えている産業界のユーザーが必要とするサポートやトレーニングを得られる体制ができました。

ポスト「京」に向けては、名前はまだ決めていませんがCHEETAHの発展版となるソフトウェアを開発しています。これまでと同様にオープンソースとしても公開する予定で、WHEEL【本誌6ページ】からも利用できるようになります。

近年、多目的設計探索は色々な分野で導入され始めていますが、まだまだ使う余地は残されていると思います。我々が開発するソフトウェアによって日本の産業界での利用者がもっと増え、競争力の強化につながることを期待しています。

自動車の燃費・高速操縦安定性に関する多目的最適化計算

自動車の空力設計では、燃費性能と高速操縦安定性を両立させる等、多目的設計探索による最適化が不可欠です。現状は風洞実験やレイノルズ平均モデル（RANS）による汎用シミュレーションを用いた最適設計が行われていますが、限られた開発時間と汎用シミュレーションの精度では設計変数に制約も多く、十分な多目的最適化がなされているとは言えません。サブ課題Bでは、開発中のリアルタイム・リアルワールドシミュレーションによる統合設計システムを自動車空力設計に適用し、サブ課題Aと連携して燃費性能と高速操縦安定性に関する多目的最適化を実施することで、システムの有用性を実証する予定です。システムの最大の特徴の一つである高速な計算モデルの作成技術を用いることで、車両形状の異なる1,000ケースを超えるLarge Eddy Simulation（LES）高精度空力解析を実施し、形状最適化を行います。ここでは京コンピュータ程度の計算機を用いるため、現状の設計開発現場で用いられている解析モデルと同等の車両形状再現性で最適化を行います。システム自体はポスト「京」を想定して、実車CADデータからの直接形状最適化の実現を目指し、その道筋を示します。



神戸大学 教授／理化学研究所計算科学研究機構 チームリーダー
坪倉 誠

ターボ機械設計システムの実証解析

サブ課題Cでは開発中のターボ機械設計システムの実証解析として、サブ課題Aと連携しファンの性能および騒音の最適化計算を実施する予定です。ファンの設計においては、性能の向上を目的とした最適化計算が実施され一定の効果があることが確認されていますが、サンプル数が数十サンプル程度であるため、探索空間が限定的です。また、設計現場では、時間平均ベースのRANS解析が主に用いられています。RANS解析では騒音の定量予測は困難であるため、低騒音化を目的とした最適計算にまで至っていません。本実証解析では、空力騒音等の乱流現象を高精度に予測することができるLESをベースとし、ファンの性能および騒音の予測に実績のあるFrontFlow/blueを用います。これをサブ課題Aで開発する収束性に優れた最適化計算システムと連携することにより、1,000サンプル程度を対象とした最適化計算を実施します。これにより、ファンの性能向上と騒音低下を同時に実現する設計技術の確立を目指します。



みずほ情報総研株式会社 サイエンスソリューション部 シニアコンサルタント／東京大学生産技術研究所 協力研究員
山出 吉伸

設計を革新する多目的設計探査・ 高速計算技術の研究開発【後編】

大規模な多目的設計探査の実現に不可欠な高速計算技術について、3人の研究者から最新の状況を聞きます。最初に、九州大学の小野謙二教授から、担当領域の概要と一つ目の高速化技術「低B/Fアルゴリズム」について伺い、続いて九州大学の飯塚幹夫研究員から、もう一つの高速化手法である「時間並列化」について、理化学研究所の川鍋友宏氏からはソフトウェアプラットフォーム「WHEEL」の開発状況を伝えてもらいます。



九州大学
情報基盤研究開発センター
小野謙二 教授

JAXAの大山先生と共同で進めているプロジェクトですね。【本誌2ページ】

はい。まず「上流設計プラットフォーム」という仕組みがあって、プラットフォームの上に大山先生の多目的設計探査技術が乗り、さらに我々の高速化技術が乗って、ソフトウェアパッケージができあがるイメージです。このパッケージを産業界のユーザーに提供し、設計活動にパラダイムシフトを起こすことが目標です。

ターゲットマシンはポスト「京」ですか？

「京」の50倍あるいは100倍と言われるポスト「京」の能力を引き出せる効率的なアプリケーションを開発することが現在の第一の目的です。「京」とポスト「京」はまったく違う計算機です。ポスト「京」はうまくプログラムを組まないと計算能力を引き出せない計算機。そこで計算技術がより重要になってきます。

振り返ってみると、「京」は性能を出しやすい計算機でした。ポスト「京」はその点で少しハードルが高そうです。ユーザーみんなが計算機のプロというわけではないので、ポスト「京」の性能を引き出すのに必要な要素技術をパッケージ化して、皆さんの使いやすいものにするのが我々の取り組みです。

先述の上流設計プラットフォームとはどういうものですか？

非常に大規模で高性能な計算機が手に入ったとして、その計算能力をどう使ったら上流過程を含む実際の設計現場の役に立

つかを考えました。

これまでのシミュレーションは、高精度な計算を1回やってその結果からメカニズムを引き出すものだったのですが、1回だけではなく、非常にたくさんの計算を行って、その結果生まれる解空間を探索することでよりよい設計パラメータや設計の方向性を見出すというアプローチを取ります。

必要な計算の量は1,000ケースとか10,000ケースといった規模になるかもしれませんが。そうすると計算データのハンドリングやジョブの投入といった作業も膨大で、もはや人の認識能力を超えてしまいます。システムティックにやる方法論が必要です。それをサポートするのが上流設計プラットフォームで、「京」の頃から続けてきたものをさらにブラッシュアップして提供したいと考えています。【本誌6ページ】

大量のケースを処理しようとすると、個々の計算スピードが重要です。

高速化のために研究しているふたつの要素技術があります。ひとつは飯塚先生に進めてもらっている時間並列計算法【本誌5ページ】。もうひとつはプログラムの実装技術です。後者の実装技術には、高い性能を持つGPU^[2]をうまく使うための技術とメモリウォール^[3]の問題に対する低B/Fアルゴリズムの技術が含まれます。

GPUに対応するんですね。

現状、アプリケーションからGPUをうまく使う技術はまだこれからです。特に各ノードが複数のGPUを持つマルチGPUマシンでフルに動かすためには、命

令だけでなくデータを継ぎ目無く供給しないといけません。そのための技術を開発します。

もう片方の実装技術「低B/Fアルゴリズム」についても教えてください。Byte per FLOP^[1]とはどういう意味でしょうか？

Byteはデータの移動量、FLOPは演算量のことです。最近の計算機の演算能力は非常に高くなってきている一方、メモリウォールの問題が顕在化しています。CPUの近くにある速いけれども小さな容量のメモリから、一番遠い所にある容量は大きいけれども遅いメインメモリまでをうまく使うためには、プログラムの特性として低B/Fである必要があるのです。データの移動量は小さく、その代わり演算量の多いプログラムを書かなくてははいけない。ところが従来の方で偏微分方程式を離散化してプログラムを書くと、B/Fは大きくなってしまいうんですね。

この問題を我々は多ベクトル化という方法で解決します。連立一次方程式を解く時、 $Ax=b$ という式が出てきます。行列ベクトル積と言われるもので、 x はベクトル、 A は係数行列です。 x を計算するために A をメインメモリからロードしてくるのですが、 x の数だけロードしているとB/Fは大きくなってしまいます。そこで、複数の x ベクトルを束ねることにします。係数行列 A は共通ですから使い回すことができるようになってデータ移動量が少なくなり、その結果B/Fは小さくなる。これが多ベクトルによる低B/Fアルゴリズムの考え方です。

開発した高速化技術は他のプロジェクトの

- [1] 浮動小数点演算
[2] Graphics Processing Unitの略。ここではCPUと併用する演算加速器のこと
[3] メインメモリとプロセッサの間のスループット不足により実行性能が制限されること

ソフトウェアにも適用されますね。

我々はキラーアプリと呼んでいて、たとえば、重点課題6サブ課題Cで開発されている風力発電のための風況予測ソフトウェアRIAM-COMPACTのHPCバージョンに、この多ベクトル技術を適用します。ある地点の風況を予測するには、16方位あるいは32方位の風向きを計算する必要がある

のですが、これが多ベクトルと相性が良いのです。従来のように1方位ずつ計算する代わりに、16方位を一度に計算することができるわけですね。【本誌6ページ】

このようにして、我々の高速化技術を重点課題6と8の他のプロジェクトにも提供していく予定です。

性能向上率の目標値を教えてください。

問題の種類によりますが、多ベクトルで6倍、GPUで3倍、そして空間方向の並列化が飽和したあとは時間並列でさらに7倍程度の性能向上を目指しています。これらは独立していますので、効果は相乗的に現れます。解く方程式の種類にもよりますが、全体で100倍程度の高速化が可能だと考えています。

逐次計算を並列計算する「時間並列計算技術」

時間の進みというのは常に一本道で、勝手に切り分けて、それらを同時に進めるなんてことはできないと思っていました。計算も並列化できるのは空間的に分割できる場合だけだと……。

わかりにくいので、これまで色々な説明の仕方を試してきました。仮の初期値を与えて反復しながら領域間の誤差を収束させていくんですよ、という説明をすると、皆さんなんとなく「そうなのかな」という顔になりますね。

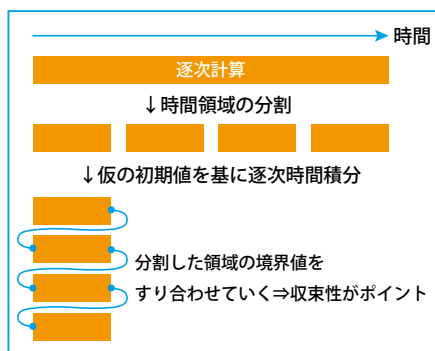
ここで言う領域とは時間を区切ってできた領域ですよ。たとえば1秒前と今と1秒後で区切った領域があるとして、今必要な初期値は1秒前の計算が終わってからでないとわからないのではないですか？

Parareal法といって、コース(粗い)ソルバとファインソルバという2つのコードを組み合わせて解く計算手法があります。コースソルバで初期値を計算し、ファイン

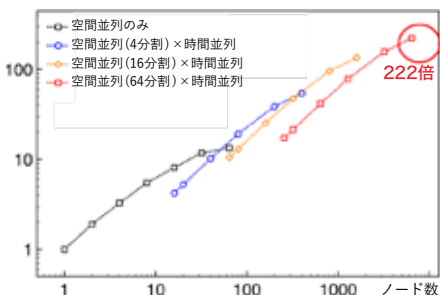
ソルバがそれを基に十分正確なシミュレーションをします。粗い初期値が原因で領域間に誤差が生じるので、コースソルバを使ってNewton-Raphson法的に摺り合わせていくという解き方になります。【図1】

すぐに出る粗い計算の答を使って、次の計算を始めるわけですね。遠回りのようにも思えますが、それで速くなるのですか？

並列効率は良くありません。空間並列なら80%の性能向上が得られる問題でも、時間並列では10%程度です。Parareal法が発表されたのは2001年ですが、当時は効率の悪さに落胆されてあまり注目されませんでした。しかし、2000年代後半から並列計算機のコア数がどんどん増えて使い切れないほどになると、効率は良くないが空間並列と独立に適用できる時間並列も使い道が出てきたのです。空間並列の性能が飽和した後でも時間並列によってさらに加速が可能になる、というストーリーですね。【図2】



【図1】 時間並列計算における時間領域の分割



【図2】 空間並列と時間並列を併用した場合の性能
空間並列のみでは約70並列で飽和する問題も、時間並列と併用することでさらなる性能向上が見込める。



九州大学
情報基盤研究開発センター
飯塚幹夫 特任教授

だんだんと有用さがわかってきました。開発の状況を教えてください。

基礎的な拡散方程式を対象に並列性能の評価を行ったところ。その結果、オリジナルのParareal法で8.6倍、待ち時間を減らすパイプライン化という処理を加えると16.6倍という並列性能が得られました。今後はより大規模な問題にこの手法を適用してレベルアップを図る段階に入ります。また、これまで困難であった移流方程式への応用もやり方がだんだんとわかってきました。

キラーアプリはありますか？

次のステップではフェーズフィールド法への応用を研究します。時間並列のアルゴリズムが有効で、産業界からの期待も大きい手法です。材料開発や災害対策といった分野での応用が考えられます。

時間並列は開発の現場にどんな変化をもたらすのでしょうか？

多目的最適化のために1,000ケースの計

算を行うとしましょう。従来の方法では、数か月に及ぶ計算の後で1,000個の結果が一度に出てきます。待ち時間が無駄ですし、計算機を長期間占有するになってしまいます。そのような問題であっても、時間並列を併用することで、一つ目の答をすぐ得られるようになります。まずそれを分析して探索方向を決めてから次の計算に進むといった使い方ができるようになるでしょう。待ち時間の少ない柔軟な計算機の運用が可能になれば、設計の現場における多目的最適化をもっと使いやすいものに変えられると考えています。

リ ア ム ・ コ ン パ ク ト HPC版RIAM-COMPACTの開発

ポスト「京」重点課題6サブ課題Cの研究開発において、九州大学応用力学研究所のグループは、九州大学情報基盤研究開発センター小野謙二教授の協力を得て、Large-Eddy Simulation (LES) をベースとした数値風況予測シミュレータ「RIAM-COMPACT」の大規模並列化のチューニングに取り組んでいます。特に今回の開発では、RIAM-COMPACTに対して多ベクトル解法を取り入れ、一度のジョブで複数の計算を同時に実行できるように改良を加えます。FX10によるベンチマークテストでは、6倍の性能向上を確認し、理論ピーク230GFLOPSに対して、128本の解ベクトルで65GFLOPSを達成しました。今回の研究開発は、RIAM-COMPACTの大幅な性能向上に寄与し、予測精度や計算時間の短縮が期待できます。また、風車のウエイクを考慮したオフショア・ウィンドファーム（洋上風力発電所）の大規模数値風況シミュレーションが可能になります。現在、上記のチューニング作業に加えて、ドイツ北部、北海に浮かぶボルクム島から沖合45kmに実在するドイツで初めてとなるオフショア・ウィンドファーム（Alpha Ventus : 5MW出力の大型風車合計12基、総出力は60MW）への適用を準備中です。



九州大学応用力学研究所 准教授
内田孝紀

上流設計プラットフォームのコアシステム ～ワークフローツール WHEEL の紹介

ものづくりは、着想してから製品になるまで、企画、設計、試作、製造など様々な段階を経ます。また、それぞれの段階において制約条件を満たす必要がでてきます。例えばコスト・価格の制約は企画段階から課せられることがほとんどでしょうし、製造段階では製造機械の制約により製造形状が制約されることもあります。ある段階で制約を満たせない深刻な問題が発生すると、前段階に戻ってやり直す（手戻り）などして解決しますが、手戻りは貴重な時間のロスとなります。手戻りを最小限に押さえるには、ものづくりの上流段階の時点において下流で課せられる制約条件を的確に把握する必要があります。つまり、後工程全体をシミュレーションできれば、手戻りを減らせるはずです。

しかし現実には後工程で何が起こるかを正確に予測することは困難です。このような「不確かさ」をシミュレーション等により定量的に取り扱おうとする取り組みは、UQ(Uncertainty Quantification) と呼ばれています。製品の製造段階では材料品質や工作精度のバラつきなどが不確かさの要因として挙げられます。バラつきがあることを前提として、モンテカルロ法や実験計画法などを用いたシミュレーションを行い、そのバラつき要因が結果、たとえば製品性能に与える影響度を上流工程の段階で推定し、手戻りを減らす取り組みが大きく期待されています。

UQの実現には、大量のシミュレーションを実行するため、大規模な計算資源が必要となります。「京」コンピュータの登場



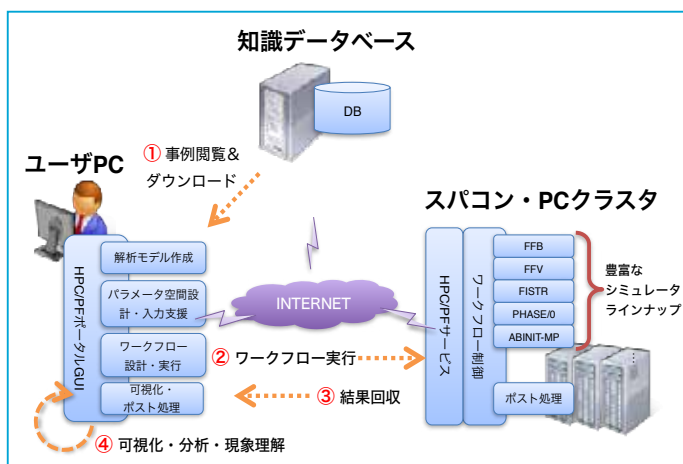
理化学研究所 計算科学研究機構
可視化技術研究チーム
川銅友宏

うに、重点課題8サブ課題Aでは、ポスト「京」の能力を最大限に引き出すことを目的に、アプリケーションソフトウェアを更に高速化する技術や、多目的設計最適化アルゴリズムの研究を行っており、それらの技術を産業界の実設計、とくに製品企画時など上流設計段階で活用できるようにするための研究開発も行っています。これらの統合的な技術を「上流設計プラットフォーム」と位置づけています。

HPC/PF から WHEEL へ

HPCI戦略プログラム分野4「次世代ものづくり」(H23～H27年度)では、「京」コンピュータを始めとするHPCI計算資源と国プロ開発の解析アプリケーションを産業界ユーザに手軽に使ってもらうために、HPC/PF (HPC Platform) を開発しました。【図3】

簡単に言えば、Webブラウザを操作するだけでスパコンを使って解析できる仕組みと、そのノウハウ集である公開事例データベースです。各種解析アプリと組み合わせたハンズオンセミナーを各所で開催し、解析アプリ利用者からHPC/PFの使い勝手



【図3】 HPC/PF システム概要

の感想や要望をヒアリングしました。その結果明らかになったのは、キャパシティ・コンピューティング用途における有効性と、柔軟なワークフロー機能の必要性でした。キャパシティ・コンピューティングとは、多種多様な計算を同時に大量実行することです。創薬や物質材料系分野では候補物質探索のため、計算条件を少しずつ変えたシミュレーションを大量に実行します。こういった用途にHPC/PFはとても便利だと評価頂きました。一方で、商用の解析パッケージソフトの利用者からはワークフローを組み立てる機能が貧弱であると指摘を受けました。具体的には繰り返し処理や分岐処理を含むワークフローをGUI上で組み立てることがHPC/PFではできませんでした。現在開発中のワークフローツールWHEEL(Workflow in Hierarchical distributEd parallel)は、HPC/PF開発で得た知見をもとに、上流設計にも応用可能なように再設計したものです。

WHEEL 機能紹介

WHEELは九州大学と理化学研究所が共同で研究開発を行っているワークフローツールです。HPC/PF同様、WebブラウザをGUIとして利用し、Windows、Mac、Linuxなど様々なデスクトップ環境で利用可能です。プログラム本体はNode.jsというJavaScriptのフレームワークで開発されており、通常はユーザPCにインストールしてスタンドアロンで使いますが、リモートサーバにインストールして、いわゆるクライアント・サーバ型アプリとして利用することも可能です。GUI上では処理の単位

FMO計算に基づく χ パラメータ算定を効率化するWHEEL

ポスト「京」重点課題6サブ課題Bの研究開発において、立教大学グループは燃料電池電解質膜に関するマルチスケールシミュレーションの研究開発を行っています。基盤となるのは、ABINIT-MPプログラムを用いたフラグメント分子軌道(FMO)計算に基づき、粗視化シミュレーションの手法である散逸粒子動力学(DPD)で用いる有効相互作用(χ)パラメータを非経験的に算定するワークフロー^[1]で、2017年度に汎用の処理システム(FCEWS)としてまとめることが出来ました^[2]。算定のコストを決めるのは、FMO計算になりますが、最も単純な2成分系としても、組み合わせとしては3通り(1-1, 2-1, 2-2)になり、各々2,000個程度の構造サンプルを扱う必要があります。実際に必要となる一連のFMO計算ジョブを実行するのに、研究室レベルのサーバを用いるより、「京」に代表される超並列資源を使う方が望ましいのは明らかです。一方で、数千～数万の数のジョブを投入して結果を回収する作業を手で行うのはきわめて困難なタスクとなります。そこで、重点課題8サブ課題Aで開発されているWHEELを用いてこうしたバルクジョブの処理を自動化し、FCEWSとの連携で χ パラメータ算定を効率化する試みを始めています。



立教大学理学部 教授
望月祐志

[1] K. Okuwaki, Y. Mochizuki, H. Doi, T. Ozawa, J. Phys. Chem. B 2018, 122, 338-347.
[2] 奥脇弘次, 土居英男, 望月祐志, J. Comp. Chem. Jpn., in press. (DOI: 10.2477/jccj.2017-0048)

「タスク」を箱で表現し、箱と箱を矢印で接続することでタスクの順序を指定することができます。また、タスク間のデータフローは、ファイル単位で行い、ディレクトリやワイルドカードを使った複数ファイルの受け渡しが可能です。【図4】

タスク内の処理はシェルスクリプト等のコマンドラインから実行可能なプログラムをタスクのプロパティ画面で登録します。タスクの実行場所(ローカル/リモート)や実行方法(コマンドライン/ジョブ投入)、ファイルI/O定義などもこの画面で設定します。

上述の通りHPC/PFではできなかった条件分岐や、繰り返し処理を含むワークフローの組み立てが可能となりました。その他、シミュレータの入力パラメータファイル内の指定した値の変化幅を任意に設定可能なパラメータエディタを装備しており、商用を含めさまざまなシミュレータに対応しています。これはパラメータ空間の単純なグリッド

サーチ(パラメータスタディ)支援をする機能で、既に重点課題6サブ課題Bで実証利用されており、その結果を開発にフィードバックしています。

また、本誌で紹介されているJAXA大山准教授らによる多目的設計最適化アルゴリズムCHEETAH

も、WHEELに組み込めるように設計しています。CHEETAHとはファイル経由の疎結合なAPIで連携しているので、他のアルゴリズムへの切り替えも容易です。

WHEELは開発中の段階ではありますが、オープンソースソフトウェアとして公開を予定しています。公開の際には計算工学ナビのサイトで告知します。

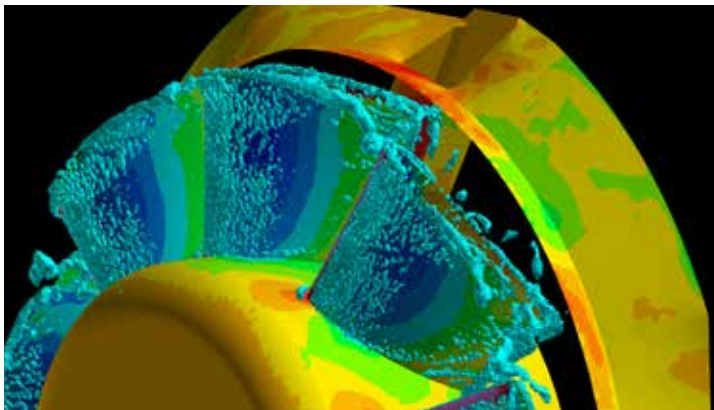
上流設計プラットフォームのこれから

上流設計にシミュレーションを活用するには、前述したUQの実現が重要ですが、最上流で活用するには、発想を支援する、というアプローチが必要です。閃きを得るためには物事を俯瞰したり、詳細に見たり、視点を変えたりといった試行錯誤が簡単にできること、これらはシステムの操作性が悪くは思考が中断してしまい実現できません。つまりユーザーインターフェイス(UI)やユーザーエクスペリエンス(UX)が重要です。

これまでスパコン応用の研究分野ではUI/UXについて無頓着な部分があったことは否めませんが、上流設計プラットフォームではWHEELをコアシステムとして、優れたUI/UXをもつデータ俯瞰・分析ツールなどを備える必要があると考えています。昨今のビッグデータ解析に対するUI/UX研究成果なども積極的に取り入れて、ポスト「京」が利用可能となったときにその計算能力を産業界のユーザーが存分に活用できるプラットフォームシステムとなるよう研究開発を続けます。



【図4】 WHEEL ワークフロー画面



今号の表紙

ファンまわり流れのLES解析

スーパーコンピューティング技術産業応用協議会（産応協／ICSCP）が実施する、ファンの性能および騒音の予測に関するベンチマークテストに参加し、ファンまわり流れのLES解析を実施した。FrontFlow/blue（FFB）により、要素数約4600万の計算格子を用いて計算した。図はファン表面における静圧と第2不変量等値面の瞬時分布である。これまでの計算により、性能および騒音を概ね良好に予測できることが確認されている。今後は、類似のプロペラファンを対象とし、FFBを用いた性能と騒音の多目的最適化計算を実施する予定である。

みずほ情報総研株式会社 サイエンスソリューション部 シニアコンサルタント
東京大学生産技術研究所 協力研究員
山出吉伸

編集後記

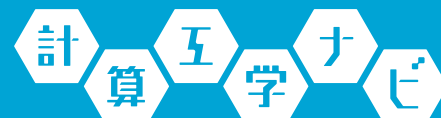
重点課題8「近未来型ものづくりを先導する革新的設計・製造プロセスの開発」と重点課題6「革新的クリーンエネルギーシステムの実用化」の各チームは相互に協力しながら研究開発を進めています。今号の記事ではそうした連携の最新状況もお伝えしています。(F)



計算工学ナビ オフィシャルサイト

本誌のPDF版やソフトウェアライブラリ、ニュースなどのコンテンツを提供しているWebサイトは下記のURLからアクセスできます

<http://www.cenav.org/>



計算工学ナビ Vol.14

発行日：2018年3月7日

発行：東京大学生産技術研究所

革新的シミュレーション研究センター

〒153-8505 東京都目黒区駒場4-6-1

office@ciss.iis.u-tokyo.ac.jp