北陸先端科学技術大学院大学
村上　舜

2025 年 12 月 23 日(火) HPC 分科会 2025 年度会合 15:00-18:00

Fujitsu Uvance Kawasaki Tower(JR 川崎タワー) 20 階大会議室

テーマ「AI と開発・利用支援の融合 ～人と HPC をつなぐ新基盤～」

講演 1

「生成 AI を援用した主に計算力学プログラミングに関する話題」

三好 昭生 (株式会社インサイト)

～学習目的から実用レベルまでの活用事例～
ここ数年で、Claude Code 等のコーディング支援 AI が実用段階に到達した。そこで、本発表では実例に基づく知見の共有を試みる。

まず初めに、学習目的の簡単な問題での活用として一次元の CAE シミュレーションを実装した。
プロンプトとしては「浮体式洋上ウィンドファームを 1D CAE で模擬する Python コードを生成せよ」と入力することで、実際にシミュレーションを生成することができた。また、機能追加・図示も行うことができた。また OpenFAST との機能比較も行った。簡単な問題では、段階的な機能追加と改善を行うことができることが分かった。

次に、Web チャット UI の CAE アプリの作成を行えるか試した。ADVENTURE on Windows のオリジナル GUI（2000 年製）をマウスクリックによる操作から自然言語で操作可能にして CAE エンジニアの利用障壁を下げることが目的である。具体的に、ユーザー「この形状で応力解析して」と入力した場合、AI エージェントは「形状確認⇒メッシュ生成⇒解析実行⇒結果表示」という操作を自動で行うことができた。

3 つ目として、C++ (60700 行&4580 の assert 文、125 ファイル) で書かれた AdvCAD の既存コードの不具合修正・機能拡張を試した。AdvCAD は 3D ソリッドモデリング&メッシュ生成ライブラリであり、メンテナンス困難で長期間放置されていた。原因としてビルドシステムが古く依存関係が不明確である。また、ドキュメントが不足して動作原理の把握が困難であった。不具合を起こしている位置はわかっていたので Claude Code でデバッグを試した。根本原因の特定には多くのやり取りが必要であったが修正することができた。修正箇所としては Robust CDT ロジックが誤った段階に適用されていたため、必要な

部分のみにロジックを適用する必要があった。次に、機能拡張の一環として近代化（C++17）を行った。具体的には例外処理システムの導入・メモリ安全性の向上・性能最適化を生成 AI の提案に従い行った。

４つ目として、Python GUI の作成を行った。PyQt5、PyOpenGL、NumPy を用いて生成した。成果として、人間が作るのをあきらめる分量を数日で生成することができた。

5 つ目に、2D FEM 接触解析を Python を用いてゼロからの新規開発を行った。具体的な手順として「1D 接触アルゴリズム(30 回反復) ⇒2D FEM 基盤構築・検証⇒2D 接触統合⇒収束最適化⇒Coulomb 摩擦実装」の順で行った。すべての実装には 8,024 行が必要であり、シミュレーションの残差も反復回数も削減できた。
また、実際のコードの生成には、検証（理論解との比較・物理法則（エネルギー保存則））も同時に生成している。エビデンスとなる検証がない場合、Claude Code はすぐにウソをついてしまうため、検証が合格したら次に進むようにすることでこの問題を解決している。また、文書化戦略が有効であり、CALUDE.md を随時アップデートしつつ、フェーズ報告書の生成、GOOD_PRACTICE.md というファイルを作成し開発手法の標準化、検証結果文書も作成していった。これらの対策により、妥当性の向上・新しいセッションに移ってもスムーズに再開できる効果があった。

最後に、実用レベルのコード開発のための、専門 RAG「caeRag」の開発を行った。これは主に設計者・CAE エンジニア向けの文書検索支援のために、PDF の文書の段組みや表を自動解析、キーワードとベクトル検索、Golden Standard Q&A の自動生成ができる。

まとめとして、CAE コードの生成を通して、生成 AI が小さなプログラムしか書けなかった時代は完全に終了しており、1000 行超のコードが 1 プロンプトでバグなく動作することもまれではない。しかし、Production level ではなく、動作はするが保守性・拡張性に問題を抱えている。単に生成しただけでは、Fallback の嵐であり、バグを隠蔽しようとする傾向があることが多い。そのため、本来失敗すべきケースが黙って成功扱いになる。「動いている」と「正しく動いている」は異なる。また、長いセッションでの劣化が発生するため、重要な事実は都度確認し、AI の反省文を信用せず実際の出力で判断する必要がある。
「表面的な動作に騙されず根本原因を特定する力」
「出力結果だけでなく中間状態を検証」
「なぜ動くのかを説明できるまで理解」
が重要である。生成 AI の利用は従来手法と比較して、単純比較は困難だが、試行錯誤の速度はけた違いに向上した。

講演 2

「生成 AI が支える HPC 利用支援の新展開」

中村 宜文 (理化学研究所)

背景

富岳のユーザーは、技術文書や FAQ にアクセスすることができるものの、それらはフォーマットが統一されておらず、配置場所も分散している。たとえば、年に一度更新される文書があったり、GitHub 上に公開されていたり、公式ホームページに掲載されていたりと、情報の所在は多岐にわたる。このため、利用者が必要な情報に迅速にたどり着くことが難しく、実際には総当たり的に資料を探さざるを得ない状況にあった。また、従来の検索機能では文書の意味を理解した検索や、複数資料を横断した検索が困難であり、その結果、問い合わせ対応にかかる人的負荷が大きくなっていた。

目的

生成 AI と RAG を組み合わせたチャット型アシスタントの有効性を検証し、最終的に GFLOPS 社の「AskDona」を導入して、2024 年 7 月 9 日より提供を開始した。本取り組みでは、利用者による自己解決の促進、サポート業務の効率化とスケーラビリティの確保、ならびに自然言語による即時質問対応を通じた利便性向上を目指している。

AskDona について

AskDona は、会話型ジョブとして「やりたいこと」を質問できる仕組みを備えている。コマンド例もあわせて提示されるため、ユーザーはそれをコピー＆ペーストするだけで実行できる。 マニュアルなどのドキュメントは追加学習するのではなく、RAG（Retrieval-Augmented Generation）データベースとして保持しており、必要に応じて参照することで回答を生成している。LLM ベースであるためコードも自然に出力でき、富岳に特化した形で利用可能である。 さらに、ユーザーからのフィードバック機能を備えており、その内容をもとにマニュアルの修正や FAQ の追加を行っている。回答モードは、素早く回答する「Fast」、標準的な「通常」、時間はかかるが高精度な「Boost」の 3 種類を用意している。 理化学研究所がマニュアル等のドキュメントを提供し、GFLOPS 社がモデルの作成および提供を担当している。

成果

2022 年よりチケットシステムを導入し、新規チケットの発行数は年間を通じておおむね 500 件程度で推移している。4 月および 10 月は新規ユーザー（課題立ち上げ）が多いため、チケット数が増加する傾向にある。チケットの種別は、質問チケット、申請チケット（リソース要求など）、通知チケット（サポートスタッフからの問題の指摘など）、非会員チケット（ログイン不可ユーザー対応）に分類され、特に質問チケットについては AskDona 導入による減少が想定される。導入初期は、LLM による対応に不慣れ

な利用者が多く、有人対応へ直接問い合わせるケースが見られた。このため、有人対応の「問い合わせ窓口」を廃止し、ユーザーはまず AskDona に「質問する」ようサポートサイトのデザインを変更した。その結果、AskDona 導入後の 2025 年 2 月以降、質問チケット数は約 5 割減少し、最大で 6 割の減少を確認した。ユーザー数が年々増加している状況を踏まえると、良好な成果が得られたといえる。
AskDona の利用状況としては、月あたり約 600 件の質問があり、セッション数（ユーザー数とほぼ同値）は約 100 であった。なお、最近は内部システムの改善により、セッション当たりの質問数は減少傾向にある。また、AskDona に対するユーザーフィードバックでは「良い」と評価する回答がじゃっかん多いものの、全体としてはおおむね半々程度で推移している。また、RAG と Deep Research を活用した研究成果閲覧支援サービス「スパコン成果ナビ」においても、利用者に応じた表現の最適化を行っている。小学生から高齢者、非専門家までの幅広い利用を想定し、「HPCI 研究成果利用報告書」および「JHPCN 採択課題報告書」を横断的に参照することで、利用者の疑問に対して関連する研究成果を自然言語で分かりやすく提示できるようにしている。

まとめ

AI を単なる対話システムから「解決のパートナー」へと進化させ、HPC におけるユーザーサポートの質とスピードを革新する AI の構築が重要である。今後は、「初めての一歩」から「成果創出」まで、あらゆる場面でユーザーを即時に支援する AI の実現を目指す。

講演 3

「コード生成 AI による HPC アプリケーションの GPU 移植」

星野 哲也 (名古屋大学)

背景と目的
「HPC-GENIE」と呼ばれるコード生成 AI 技術により高性能 HPC ソフトウェアの開発効率を高める革新的 AI 基盤の開発を目指すプロジェクトがある。
名古屋大の GPU 搭載機は 2026 年 10 月運用開始である。そのため、レガシーアプリケーションの GPU 対応は待ったなしであるが、課題が多く進み具合はよくない。特にレガシーアプリに多い Fortran で記述されたプログラムが課題である。原因として、ベンダーによってまちまちな並列プログラミング言語とモデルを使用し、CPU 版と各ベンダーの GPU とプログラミング言語の組み合わせがあり、保守コストが増加している。
そのため、AI を使って GPU 化を行いたい。BLAS などの関数レベルは成功報告がたくさんあるが、HPC アプリケーションレベルはどうなのかを調査する。
Claude Code と Claude Opus 4。1（Claude Code と連携してコード開発からテスト実行まで一連のタスクを自律的に実行可能）をどれだけ使えるかを評価する。

Claude Opus 4。1 はログインノードや計算ノードにインストールして使用（推論は外部なのでネットワークの外部接続がいる）。ファイルシステム上のソースコードを直接編集・実行可能であり、ユーザー権限でコマンドも利用できる。しかし、推論にはランダム性があり、seed の固定ができず、再現性がないことが問題である。

アプリケーションとして GeoFEM/Cube（CG 法による 3 次元静弾性問題ソルバ―）の GPU 対応版を開発して評価を行った。

GeoFEM/Cube は MPI+OpenMP で並列化された Fortran ベースのアプリケーションであり、GPU を含め様々な環境向けに最適化された実績がある。そこで、Claude Code により生成されたコードの計算速度＆開発自体の時間を評価。
アプリケーションは行列生成部とソルバに二分されており、行列生成部は並列化して動かすだけなら難しくはないが、ループ構造が非常に複雑であり GPU で効率よく並列化するには 2 重ループの統合が必要など単純ではない。
ソルバは CG 法と呼ばれる手法であり、一般的な線形代数演算の組み合わせからできているため、並列化も簡単で計算速度も出すことができると予測できる。

実際に移植する上で、Claude Opus 4。1 の出力は、コマンドの実行結果だけで、Claude Code にプロンプトを与えない限りは直接コードから生成内容を判断する必要がある。

実験

初めに、単に GPU 化せよとだけ指示し出力する Fortran の形式を「固定長形式(。f)」「自由形式(。f90)」の 2 種類を試し、また入力する移植前のコードも「OpenMP 指示文あり」と「OpenMP 指示文なし」を試した。さらに、出力言語を OpenMP offloading、OpenACC、CUDA の 3 種類を指定し生成できるかを試した。
最後に「さらに早くせよと指示」することでより効率的な GPU 化が可能かを試した。

結果

固定長形式による移植では、Claude が CG ソルバを早くするべきと判断し、GPU 化におおむね成功した。生成時間は 800 秒から 1000 秒くらいと長く、人間がコーディングした場合と同様に文字数制限でコンパイルが通らないことが多かった。
係数行列の生成に関しては 1 度も GPU 化に成功しなかった。OpenACC の指示子を適切に使えたのは 2 回でそれ以外の試行では適切に設定されていなかった。

自由形式で生成すると 600 秒くらいで移植が完了した。しかし、一度も適切な OpenACC のコードが出力されなかった。
入力するコードの OpenMP 指示文がない場合、疎行列生成の移植はそもそも試行されなくなった。移植に成功するのは 10 回に 1 回であった。
出力形式も OpenMP を使えと指定すると GPU 化に成功したのは 5 回であり生成時間は早かった。しかし、計算性能は悪いコードが生成された。
指定言語に OpenACC を指定した場合の、計算性能はいまいちであるが、実行可能となるコードを生成する可能性は高かった。CUDA Fortran を指定した際は、実行可能なコードを生成することが一度も成功しなかった。

「さらに速くしなさい」という指示はかなりの確率で適切なコードに変更された。性能もよくなった。二度目の指示では「さらに最適化を入れろ」と入力した結果、async 節の利用などで計算性能がよくなることが確認できた。同様に 3 度目の指定では無理に最適化しようとした結果、むしろ計算速度が遅くなってしまった。

まとめ

CG 法の部分は OpenMP 指示文がなくてもおおむね GPU コードの生成に成功
係数行列生成は OpenMP 指示文があるにも関わらず GPU 化を行うことができなかった。
追加で行列生成部分を並列化しなさいと指示すると人間が書いたのよりは遅いが、GPU 化はできた。こ

のとこから、プログラマが持っている知識を効率よく与えてあげる方法が必要であると考えられる。

講演 4

「大規模言語モデル「Takane」が拓く HPC の未来」

白幡 晃一 (富士通株式会社)

富士通のエンタープライズ向け LLM の「Takane」を HPC に利用する。
専門業務を支援するエージェントや材料探索における因果関係の自動説明などを紹介
Fugaku-LLM や Fujitsu ナレッジグラフ拡張 RAG を開発・提供している。垂直統合した AI を目指して、お客様が主権を持てる Sovereign（Security, Flexibility, Domain Specific）を提供していく。そのために、LLM の強化・プラットフォームの開発/提供を行っていく。
富士通の AI 戦略は Kozuchi と呼ばれる（特化型 AI 蒸留・量子化、ナレッジグラフ拡張 RAG 等を搭載した）AI プラットフォームを開発している。NVIDIA との連携、MONAKA や、グローバル標準のセキュアな AI 基盤を提供。

生成 AI による組織・企業活動のより高度な業務を支援することを目指している。
1 個人業務の効率化
2 組織業務の自動化
3 経営業務の支援

汎用化から産業化するため、領域特化・個別化することにより、高コストパフォーマンスでエンタープライズへ適用できる。そのためのベースモデルとして、モデルサイズは single GPU から利用可能な軽量でが、カスタマイズ性の高い Takane 2.0 を提供している。そのために次にあげる技術を開発している。

・1bit 量子化技術（QEP: Quantization Error Propagation）
従来手法 GPTQ と比較して 80 tokens/s から 100 tokens/s に高速化した。推論精度も 89%に落ちるだけ。(従来は 0%)。メモリ使用量最大 94%削減した。
従来法ではレイヤー単位で量子化しているが、開発手法はレイヤー間で量子化誤差を考慮しながら伝搬する。サンプルデータをあらかじめ用意して安定位置をあらかじめ決めておく。

・特化型 AI 蒸留
余計な知識を削減した軽量 AI モデルに対して大規模モデルから必要な知識のみを転送する。社内実践としては商談予測のタスクで教師モデルよりも良い精度を達成。モデルアーキテクチャ探索も開発し、モデルから蒸留後の性能予測を行うことが可能になった。

・生成 AI トラスト
ハルシネーション対策：モデルを解析して説明性を付加、判断根拠を出力できるようにする。

・偽情報対策
生成 AI や合成コンテンツによる偽情報の流通が社会問題化。判定に AI を使う。

まとめ
富士通のエンタープライズ向け LLM は汎用から専門化の実践を提供している。設計製造分野では iCAD と連携させて自然言語でオペレーションできることにより本質的なところに時間をさけるようになる。材料開発ではメカニズム解析・施策立案に活用し、因果グラフの生成　化学反応のメカニズム説明が可能。創薬分野では電子顕微鏡画像から立体的・時間変化な分子構造を生成している。（純粋な LLM ではなく Alpha Fold 系である）