



2006 年度下期未踏ソフトウェア創造事業 採択案件評価書

1. 担当PM

高田 浩和(株式会社ルネサステクノロジ)

2. 採択者氏名

開発代表者:安藤 英俊 (山梨大学大学院 医学工学総合研究部)
共同開発者:鳥山 孝司 (山梨大学大学院 医学工学総合研究部)

3. プロジェクト管理組織

日本エンジェルズ・インベストメント株式会社

4. 委託金支払額

12,930,000 円

5. テーマ名

GPU 上での CIP 法に基づく数値シミュレーション環境の開発

6. 関連Webサイト

なし

7. テーマ概要

本プロジェクトでは並列演算装置である GPU(Graphics ProcessingUnit)上で最新数値シミュレーション手法である CIP(Cubic Interpolated Propagation/Constrained Interpolation Profile)法を実装し、身近な PC 部品だけで最新の数値シミュレーション

手法を高速に実行可能にすることを目的とする。CIP 法は従来手法に比べて問題を大幅に高速・高精度・安定的に解くことができ、世界的にも注目される日本発の数値シミュレーション技術である。

GPU は本来 3DCG 処理を高速化することに特化したハードウェアで、内部は高度に並列化された演算装置の集合体である。単一コア CPU の性能向上が近年頭打ちとなる一方で GPU は毎年 2 倍近い性能向上を達成してきており、身近で高性能な計算資源として特に注目されてきている。この GPU を汎用計算に応用する研究は GPGPU(General Purpose GPU)として先端的研究分野の一つとなっており、古典的数値シミュレーション手法の実装・高速化等も海外の研究者により報告されている。

日本発の画期的な数値シミュレーション手法である CIP 法を GPU 上で実装しライブラリ化して提供すれば、ユーザは GPU に対する高度な知識無しに CPU に比べて大幅に高速な数値シミュレーションの実行が可能になる。またシミュレーション結果の可視化も GPU 上で高速に行う。

8. 採択理由

GPU(Graphics Processing Unit)をベクトル型並列演算装置として使用し、高性能な数値シミュレーション環境を実現するとの提案である。地球シミュレータでも利用されている最新数値シミュレーション手法である CIP(Cubic Interpolated Propagation)法を GPU 上に実装し適用することで、市販の PC を用いて数値シミュレーションを高速・高精度・安定的に解くことを可能とする。

本プロジェクトでは、GPU を用いた数値シミュレーション結果の高度な 3 次元可視化技術についても開発を行う予定であり、これらの開発に成功すれば、容易に入手可能なハードウェアを用いた高性能数値シミュレーションを誰もが利用できる環境が実現する。開発成果は無償公開される予定であり、ゲームや CG 製作のみならず、その他の分野においても数値シミュレーションの活用と普及が一気に加速される可能性がある。

以上のように、非常に有用な技術であり、応用範囲も広く、また学術的にも意義のある研究テーマであることから、未踏プロジェクトとして推進することが適切と判断し、採択を行った。

9. 開発目標

本プロジェクトでは、ベクトル型並列演算装置である GPU 上で最新数値シミュレーション手法である CIP 法を実装し、身近な PC 部品だけで最新の数値シミュレーションを高速に実行可能にすることを目的とする。

- GPU プログラミングの知識がなくとも CIP 法に基づく本格的なシミュレーションを利用可能とするために、数値シミュレーションのコア部分を GPU 上のシェーダプログラムとして実装し、GPU に関する知識がないユーザでも C/C++ から利用可能とする。
- シミュレーションのコア部分だけでなく、環境設定ツールおよび可視化ツールも提供することで、多くの人にとって使いやすく本格的な数値シミュレーション環境の構築を目指す。

10. 進捗概要

採択時点において、すでに GPU を用いた基本的な流体方程式の数値計算コア部分および2次元可視化プログラムのプロトタイプが動作していたが、今回のプロジェクトを通してさらに性能を向上させることができ、数値計算結果の3次元可視化にも対応して、大幅に完成度を高めることができた。

開発に際しては、演算結果の検証やデバッグ容易化の点からも、まずは数値シミュレーション結果の2次元・3次元可視化プログラムから開発を進めた。

5月の中間報告時点には、すでに2次元・3次元の CIP 法の実装がほぼ完了しており、その後 CCUP 法についても実装を行った。その後は、8月の最終成果報告までに、環境設定ツールの開発と、数値計算ルーチンのチューニングを実施した。

開発は順調に推移した。開発内容についても当初の計画に対して未達成の部分はない。逆に、以下の部分では計画以上に開発を進めることができた。

- CIP 法について当初は基本的な C 型や M 型のみの実装を計画していたが、有理関数 CIP や保存型 CIP などの様々に拡張された CIP 法についても実装することができた。
- 圧力ポアソン方程式の解法についても、当初は共役勾配法についてのみ実装する計画であったが、種々の解法について実装することができた。
- 環境設定ツールおよび可視化ツールについては閉じた統合環境として実装す

る計画であったが、それにとどまらず外部の既存ツールによって作成されたデータの読み込みや書き出しにも対応することができた。これによって個別のツールの利用価値が高まった。

11. 成果

本プロジェクトでは、以下のハードウェア/ソフトウェアからなる、高速数値シミュレーション環境を開発した。

- GPU 上での CIP 法に基づく数値シミュレーションコア部分
- 可視化ライブラリとツール
- 環境設定・モデリングツール

CIP 法に基づくシミュレーションコア部分、シミュレーションの環境設定およびモデリングを行うツール、そしてシミュレーション結果の高速・高品位可視化ライブラリを全て GPU 上で高速に動作するように実装した。実行・開発環境としては Windows 上の DirectX および HLSL を前提とすることとした。

- コア部分については様々な種類の CIP 法を効率よく実装し、CPU と比べて圧倒的に高速にシミュレーションが実行可能であることを確認した。移流以外の部分については CCUP 法の GPU 上での実装も行った。
- 環境設定およびモデリングツールでは 2 次元・3 次元シミュレーション用の環境設定、障害物の配置や速度の設定、形状モデリングを可能にし、シミュレーションの準備を判りやすく行えるようにした。
- 可視化ツールではシミュレーション結果を高品位にリアルタイム表示することで、結果の理解を助けることを可能にした。
- 複数 GPU による高速化については NVIDIA の SLI 環境で性能を評価し、一定レベルの高速化を実現した。

このように本プロジェクトでは当初の目標を全て達成し、高速・高精度で使いやすいシミュレーション環境の開発に成功した。

本プロジェクトの開発成果であるシミュレーション環境の特徴は、既存技術に基づく製品と比較して以下の通りである。

- 既存製品よりもはるかに高速に実行可能であるため、シミュレーション時間を大幅に短縮できるか、同じ時間内にずっと精度の高いシミュレーションを実行

可能であること。

- 高速・高品位なリアルタイム可視化も可能で、環境設定ツールも付属するなど、シミュレーションのコアのみならず「環境」として既存の高価な製品よりも性能・機能共に大幅に優れていること。
- これらの利用条件としては、身近にあるごく一般的な GPU を備えていることのみであり、多くの人に恩恵があるということ。また GPU は毎年2倍以上の高速化を続けており、将来的にも有望な計算資源であるため、開発成果の利用価値はますます向上すると期待できる。

以下、具体的な成果について説明する。

1. GPU 上での CIP 法に基づく数値シミュレーションコア部分

シミュレーションのコア部分としては、CIP 法による高精度な移流計算、差分法による粘性計算、そして圧力ポアソン式の計算等を GPU 上で効率よく実装した。まず CIP 法による移流については標準の C 型 CIP 以外に M 型 CIP も実装した。また有理関数、デジタイザ、保存型にも対応した。

図 3-1 は 2 次元移流問題として有名な Zalesak ディスク問題を例題とし、2 次元 CIP で正確な移流が行えることとの検証と処理時間の CPU との比較を行っている様子である。移流はきわめて高精度に行われ、また処理時間は CPU と比較して数倍～数十倍という結果であった。実際にはシミュレーションのグリッド数が増大して計算負荷が増えるほど GPU が相対的に高速に処理可能という結果であった。これは GPU での処理は CPU からのドライバの呼び出しによって行われることから、シミュレーショングリッド数と無関係に一定の時間がかかってしまうため、グリッド数の増加による計算時間の増大によってドライバ呼び出しのオーバーヘッドが相対的に減少するためである。

図 3-2 は 3 次元空間内での CIP 移流の実行例で、回転速度場による移流によっても CIP 法できちんと境界面の追跡が行われていることを示している。

次に差分法による粘性計算の実装を行ったが、これは比較的単純な作業であるため特筆すべき点は特にない。圧力ポアソン式の計算のために、GPU 上でヤコビ法、ガウスザイデル法およびレッド・ブラック・ガウスザイデル法を実装した。また共役勾配法の実装も行った。ただし一般の数値シ

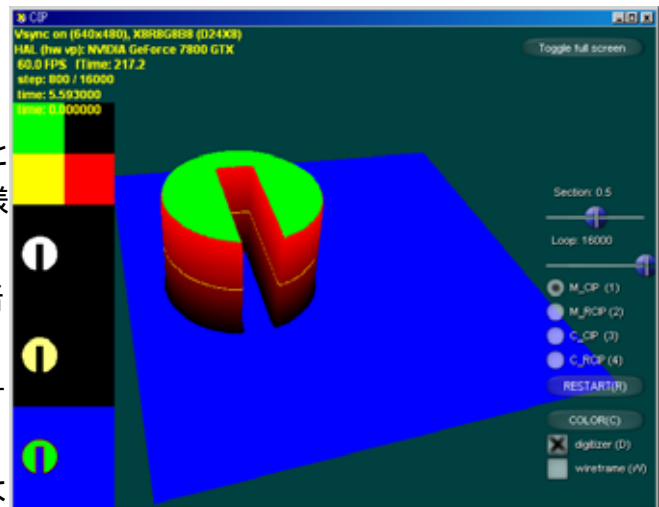


図 3-1: 2 次元 CIP の動作確認画面 (Zalesak ディスクによる検証)

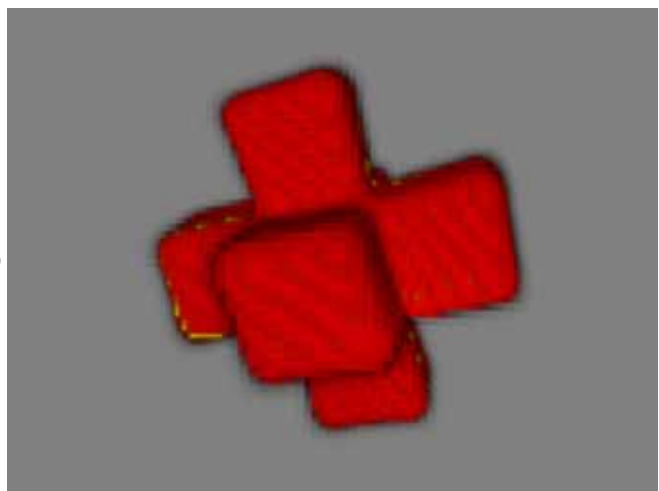


図 3-2: 3 次元 CIP の動作確認画面

ミュレーションで広く使われる、前処理付きの共役勾配法については、前処理部分で GPU に馴染みにくい逐次処理を伴うものが多いために、前処理無しの共役勾配法のみを実装した。GPU 上では一般にヤコビ法が広く使われているが、実験の結果レッド・ブラック・ガウスザイデル法が速度と精度と安定性の点で優れていると判断し、これを主として用いることとした。

2. 可視化ライブラリとツール

GPU 上で CIP 法に基づくシミュレーションが可能になったとして、次に重要なのは結果の可視化である。GPU 上でのシミュレーション結果は GPU 上に保持されているため、これを GPU を用いて可視化するのは非常に効率がよく、また GPU は本来表示処理用のハードウェアであるため可視化は得意である。我々は GPU プログラミングの知識と経験を活かし、他に類のない高速・高品位な可視化を実現した。

図 3-3 は 2 次元流体シミュレーションにおいてカルマン渦が発生している様子を可視化したものである。流れ場のようなベクトル場の可視化については、パーティクル(粒子)を流れ場に沿って移動させることで流れの様子を分かりやすく表示する方法と、流れの大きさと向きを表す矢印を画面内に配置する方法を実装した。

これらは全てリアルタイムに可視化可能であるため、ユーザはシミュレーション中に自由に可視化方法を変更することが可能である。粒子や矢印には速度や圧力などに応じて色を付けることも可能で、これにより一層シミュレーション結果の理解を手助けできる。また圧力や温度などのスカラー値については、値を色に対応させて表示する方法や、等値線を表示する方法を実装した。これによってスカラー値の分布の変化をリアルタイムに把握可能になった。これらの可視化手法はそれぞれライブラリとして実装され、必要に応じてアプリケーションに組み込むことも可能である。また可視化手法

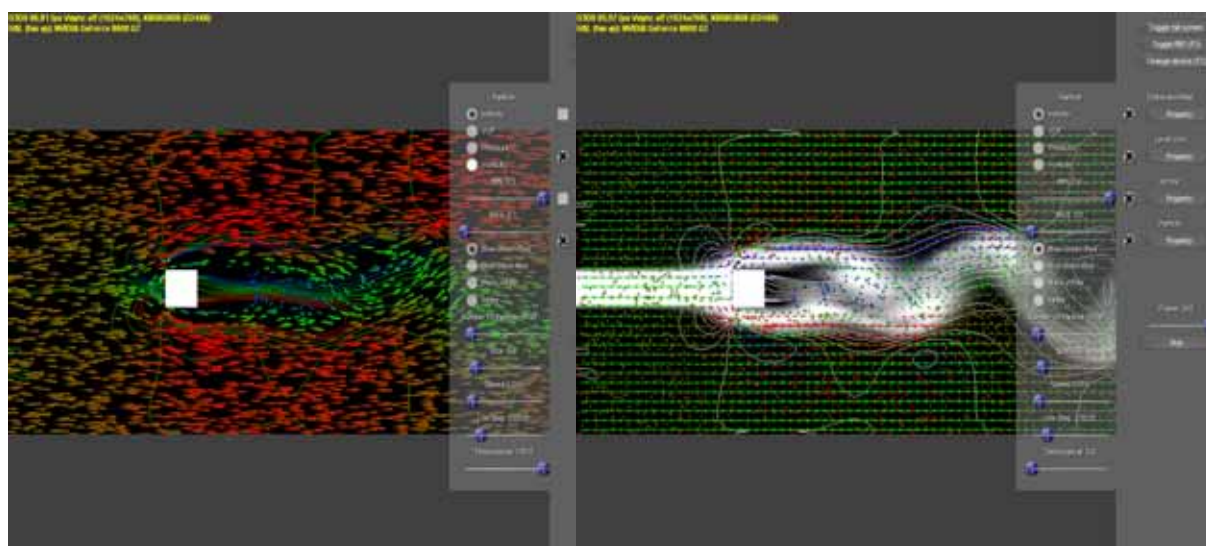


図 3-3: 2次元流体シミュレーションによるカルマン渦の可視化例(左:粒子追跡、右:矢印と密度)

は自由に組み合わせることで表示可能であるので、矢印と粒子と等値線を一度に表示することも可能である。可視化のための種々のパラメータは GUI によりユーザが自由に変更可能である。

図 3-4 は 3 次元シミュレーションの可視化例である。3 次元シミュレーション結果についてもパーティクルや矢印によって流れ場をリアルタイムに可視化することが可能である。またスカラー場については指定平面上に分布図や等値線を表示することも可能である。また本プロジェクトで開発された GPU 上のシミュレーションによる結果の表示だけでなく、別の外部のシミュレーションシステムによる計算結果の表示にも対応するため、外部ファイルからの読み込みにも対応した。

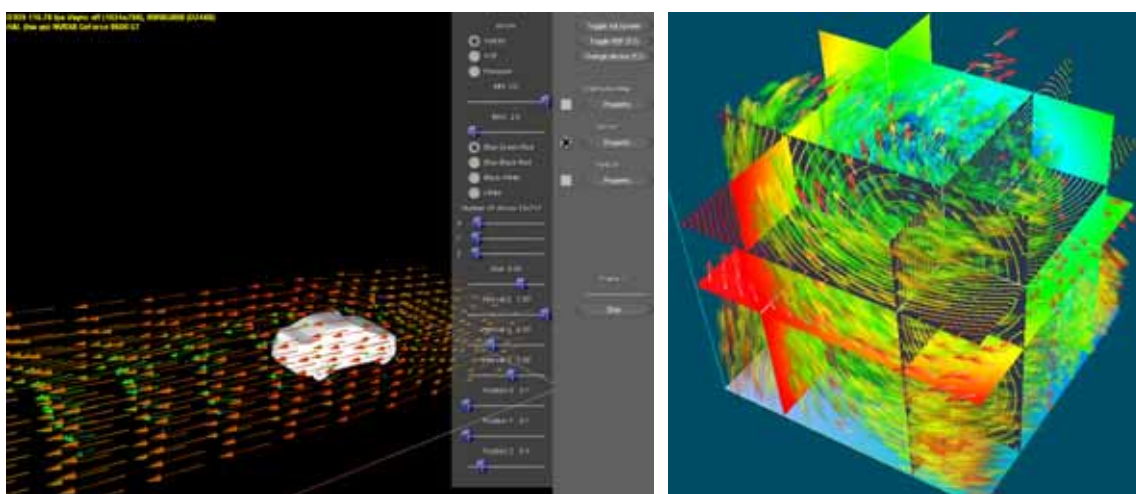


図 3-4: 3次元流体シミュレーションのリアルタイム可視化例

3. 環境設定・モデリングツール

数値シミュレーションを行うためには、まずシミュレーション環境の設定を行わなければならない。必要な物理パラメータの設定、障害物や壁の配置などを簡単に行えて、そのままシミュレーションが可能になるような環境設定ツールを開発した。このツールも通常的手法ではリアルタイムでの設定やモデリングが困難であるが、今回は GPU の利用を前提に対話的に環境設定・モデリングを可能とした。図 3-5 はそのツールの利用画面の例で、左側の図では 2 次元シミュレーションのための障害物の配置と編集を行っている。右側では 3 次元シミュレーションのための障害物のモデリング例であり、

この車のモデルに対してシミュレーションを行うことで車体の周りの空気の流れや空気抵抗を評価することが可能になる。この図の車体を用いたシミュレーション結果が図 3-4 の左図である。

またこのツールはシミュレーションを目的としない3次元モデルの構築に利用することもでき、物体の切削が容易に行えることから金型の切削シミュレーション等にも利用可能である。

12. プロジェクト評価

本プロジェクトでは、GPU をベクトル型並列演算装置として使用し、高性能な数値シミュレーション環境を実現した。応用範囲も広く、有用な技術であり、また学術的にも意義のある研究テーマであることから、今回の未踏プロジェクトで採択を行ったものである。地球シミュレータでも利用されている最新数値シミュレーション手法である CIP 法を GPU 上に実装し適用することで、市販の PC を用いて数値シミュレーションを高速・高精度・安定的に解くことを可能とするなど、開発された成果は非常に有用なものとなった。

結果は驚くべきもので、シミュレーション内容によっては、GPU を使用したベクトル演算の結果は CPU の数十倍に達している。また、シミュレーション結果の2次元・3次元表示を行うことのできる可視化プログラムについても、GPU の特徴を生かし、種々の表示形態に基づいて結果をリアルタイムに表示できる画期的なものとなった。ユーザインタフェースなど、プログラムの完成度も非常に高い。これらの成果は、容易に入手可能なハードウェアを用いることにより、誰もが高性能数値シミュレーションを利用できる環境の実現に道を拓くものである。

開発は、鳥山氏が CIP 法などによる数値計算プログラムを C/C++ で実装、さらにそれを安藤氏が GPU に適したプログラムに変換して実装するという形で進められた。GPU のシェーダプログラムで数値計算を行う場合、メモリアクセスに非常に制約があることから、GPU をベクトル型計算機として使用するのとは決して容易なことではない。しかしながら、開発者らは、GPU アーキテクチャに関する理解と、数値計算に関する深い見識と経験を駆使して、見事にハードウェアの性能を完全に引き出した。

また、本プロジェクトでは、GPU を用いた数値シミュレーション結果の高度な3次元可視化技術についても開発を行ったが、数値シミュレーション結果をただちに表示できる点で GPU のメリットを最大限に利用したものとなっている。なかでも注目すべきは数値シミュレーション結果を 2D/3D でリアルタイムに表示することのできる可視化プログラムである。この可視化プログラムだけを見ても、非常に有用性は高く、今回開発さ

れたプログラムについてはその完成度の高さに全く驚かされた。

近年の GPU の性能向上のスピードは、マイクロプロセッサのそれを凌駕するものであるが、今回開発された技術が普及すれば、容易に入手可能なハードウェアを用いた高性能数値シミュレーションを誰もが利用できる環境が実現する。ゲームや CG 製作のみならず、その他の分野においても数値シミュレーションの活用と普及が一気に加速される可能性があり、今回開発された成果のインパクトは極めて大きいと言える。今回の開発成果は無償公開される予定とのことでもあり、今後の応用の広がりについて期待している。

13. 今後の課題

今回の開発プロジェクトにおける最大の困難は、ハードウェアの不安定動作にあったと聞いている。最新の GPU を搭載したグラフィックス・ボードについては、ドライバ・ソフトウェアも含め、開発途上の部分もあり、完全に枯れているとは言い難い部分もある。実用化にむけて、安定動作は絶対的に重要と思われ、Linux/OpenGL 環境への対応も今後の重要な課題として提起されているが、これについてはぜひ対応をお願いしたい。

また、開発されたプログラムの完成度は高くほぼ実用的なものとなっているが、使い勝手や性能など、まだ改善の余地が残されている部分については、継続的な研究と開発を期待したい。今後、ユーザからのフィードバックを得ながら、実用化に向けてさらに改良を進めて欲しい。

最後に、開発者の安藤氏、鳥山氏の成果報告から、今後の課題について抜粋したものを列挙する。

- Linux/OpenGL への対応 (長時間稼働時の安定動作化)
- 計算精度の向上 (64 ビット化)
- 結果の分散可視化
- 外部レンダラによる超高品位な可視化
- 適応型格子 (CIP 自体の拡張、さらなる高速・高精度化)
- CG 製作分野、ゲーム等への応用と展開