

# GPU を用いた疎な多項式補間の高速処理法

宮下 宏樹<sup>†</sup>

村尾 裕一<sup>†</sup>

## 1 はじめに

近年,グラフィックス処理用のプロセッサである GPU を汎用計算に応用する GPGPU が幅広い分野で盛んに研究されている.本研究では,応用範囲として数式処理分野を対象とし,疎な多項式補間の処理に GPU を使用し高速化することで数式処理分野での GPU の応用の可能性を検討する.研究対象の多項式補間は,評価可能であるが未知の多変数多項式を入力とし,入力多項式を評価した値の列からその多項式表現を求めるもので,行列式を展開しその多項式表現を求める際に用いられる.本研究においては,疎な多項式の補間を行う際に有用な Ben-Or & Tiwari のアルゴリズムと,モジュラー算術および中国剰余定理 (CRT) を組み合わせた手法により多項式補間を行う [2].

GPU 上の実装には NVIDIA による開発環境 CUDA を使用し [1], GeForce GTX570 を用いて実験した結果, Core i7-920 での逐次処理に比べ最大 45.3 倍の速度向上を達成した.

## 2 並列化方法

本研究における補間手法は, Ben-Or & Tiwari のアルゴリズムの問題点である数の膨張を改善した方法で,複数の法を使用して計算が行われる.本手法は多項式の指数の決定と係数の決定の 2 つのステップに分けられ,この内指数の決定部分が全体に対し高い割合を占める.よって,指数決定部分の高速化が目的となる.以下に指数決定部分の処理の概要を示す.

1. 入力された多項式を評価した値の列に対して Berlekamp/Massey アルゴリズムなどを適用し値間の関係を表す多項式を求める.
2. 処理 1 で求めた多項式を構成因子に分解する.主な計算は, 1 で求めた多項式と複数の多項式 (本実装では 200 ~ 400 個程) との GCD 計算であり,共通因子を見つけ部分因子へ分解する.
3. 処理 2 の多項式の分解と同時に求まる値を組み合わせたベクトルを基底変換し指数を求める.

ただし,処理 1 および 2 は全ての法でそれぞれ同様の計算を行う.

上記の処理の内,処理 2 の多項式を部分因子へと分解する計算のコストが高いため,この部分を並列化できれば高速化が達成される.ここで,多項式の GCD 計算は互いに独立であるため並列化が可能である.以上より,多項式を因子に分解する処理は,まず各法ごとに独立であり,さらに多項式 GCD の計算も独立である.よって,アルゴリズムを CUDA の並列機構に対応させると, CUDA での並列化の基本方針としては,法ごとの処理をブロックに, GCD 計算をスレッドに割り当てることにする.実際には以下の 3 段階に分けて並列化を行う.

1. 1 つの法の処理を 1 つのブロックに全て担当させる.
2. 1 つの法の処理を複数のブロックに分割する.
3. 方法 1, 2 の場合で 1 つの多項式の計算を複数スレッドで並列に実行する.

GPU 上へ実装を行い並列化部分の実行時間を測定したところ, CPU での逐次実行の結果に比べ,方法 1 で最大 14.5 倍,方法 2 で最大 15.5 倍,方法 3 で最大 45.3 倍の速度向上が達成された.

## 3 まとめ

現時点の実装で, CPU の逐次処理と比較して最大 45.3 倍の速度向上が達成されていることから,多項式補間, ひいては数式処理においても GPU は十分有用であると考えられる.

今後の課題としては,用いる法の組み合わせやスレッド数などの最適値の検討や,既存の他の補間手法との比較などが挙げられる.

## 参考文献

- [1] NVIDIA Corporation. *NVIDIA CUDA C Programming Guide*, Version 4.0, 2011.
- [2] Hirokazu Murao and Tetsuro Fujise. Modular algorithm for sparse multivariate polynomial interpolation and its parallel implementation. *J. Symbolic Computation*, 1999.

<sup>†</sup> 電気通信大学大学院情報理工学研究所