A-014

グリッドコンピューティングを用いた連立代数方程式の解法 - 根の振る舞いの解析に向けて -

Numerical Calculation of Algebraic Equations with Grid Computing - For the Analysis of Behavior of a Root -

鈴木 秀男 † 小林 英恒 ‡ 三浦 正浩 §
Hideo Suzuki Hidetsune Kobayashi Masahiro Miura

1. まえがき

連立代数方程式の根の振る舞いを解析するための方法として，筆者らは文献 [1, 2, 3, 4] に示すような方法を考案している．これらの文献では，根の振る舞いとして，根の重複度を数値的に挟み込むための「挟み込み計算」と非常に近接度の強い近接根の「分離と高精度な計算法」を導出している.「挟み込み計算」も「分離と高精度な計算法」も理論的には，根の近傍での振る舞いを解析し，多変数・高次元へも適用可能な方法である．

現在，筆者らは，PC グリッドコンピューティング上で「挟み込み計算」を行うためにアルゴリズムの修正作業を行っている．本論文では,「挟み込み計算」のアルゴリズム実装の第1歩として，連立代数方程式の根を PC グリッドコンピューティングを利用することにより，効率的に計算できることが確認できたので，その結果を報告するとともに，アルゴリズムの修正作業の概略にも言及する．

2. グリッドコンピューティングの概略

グリッドコンピューティングとは，複数のコンピュータを結ぶことで仮想的に高性能コンピュータを作り，その資源を利用者に提供するシステムであり，大量の計算が必要な処理，大量のデータのやりとりが必要な処理を不可欠とする分野において，利用が進んでいる．

「挟み込み計算」は，処理をパスごとに独立して実行でき，それぞれの処理に依存関係はないため，処理 (タスク) を並列に実行することによって，効率化を見込むことができる．

グリッド上でタスクを並列に実行する仕組みとして GridRPC がある．本研究では，GridRPC の参照実装である Ninf-G[5, 6] を利用することによって，タスクを並列に実行するグリッドアプリケーションを作成した．

3. 「挟み込み計算」の概略

「挟み込み計算」とは，簡単に説明すると，与えられた連立代数方程式を特定の超平面で切断し，簡略化された連立代数方程式を導出し，対象とする根の近傍での数値解を計算することにより真の重複度をプラス方向とマイナス方向の両側から挟み込むものである．この手法の特徴は

1. 全ての根をあらかじめ求める必要はない
2. 与えられた点の近傍のみ考えればよい
3. 簡単な比の計算で重複度が求まる

などが挙げられる．

連立代数方程式のすべての根の近似値を統一的に計算する有力な方法にホモトピー法がある.「挟み込み計算」においても対象とする根の近傍に存在する根の近似値を計算するためにホモトピー法を採用している．

筆者らは，座標空間及びホモトピーパラメータへ代数的な処理を施し，近接根を分離するとともにパス追跡の精度を向上させる機能を取り入れたホモトピー法を考案している [1, 2, 3, 4]．特に，ホモトピーパラメータへ代数的な処理を施すことにより，近接根へ収束するパスを通常の数値精度以上の精度で計算することが可能となる．

以下に,「挟み込み計算」の計算手順の概略を述べる．

1. 与えられた連立代数方程式を超平面で切断し，新たな連立代数方程式を構成する．
2. 新たな連立代数方程式に対して座標空間の一次分数変換を適用する．
3. 改良されたホモトピー法で対象とする領域内に存在する根の近似値を計算する．
4. 得られた近似値を逆の一次分数変換を利用してもとの空間へ戻す．
5. もとの空間での近似値と切断に利用しなかった方程式から，代数方程式を導出する．
6. 必要であれば座標空間へ一次分数変換を適用する．
7. 代数方程式の対象とする領域内に存在する根の近似値を計算する．
8. 必要であれば，得られた近似値に逆の一次分数変換を適用してもとの空間へ戻す．
9. 根の近似値の座標成分から比の値を計算し，それらの総和を計算すれば，それが重複度となる．

超平面切断により新たに構成された連立代数方程式は，元の連立代数方程式に比べて，方程式の個数及び変数の個数が減少し，さらに個々の方程式の次数が低下することにより総次数も減少する．特に，切断に利用する超平面を適切に選べば効率的に次数低下が行える．

また，上記のアルゴリズムに基づき，切断する超平面を対称となるように選ぶことによって，それぞれ計算された重複度が真の重複度をプラス方向とマイナス方向とから挟み込むことが可能となる．これが,「挟み込み計算」と呼ばれるゆえんである．

実際の計算は，数式処理と数値計算のハイブリッド計算として実現している．

†能力開発総合大 東京校 情報技術科
‡日本大学 理工学部 数学科
§株式会社 創夢

4. 「挟み込み計算」のアルゴリズムの修正とその実装

ホモトピー法とは，ある初期値から出発するパスを追跡して，ホモトピーパラメータが終値へ達したときにたどり着く点を根の近似値とする方法である．この方法は，初期値から終値までのパス追跡が独立して行えるため，この部分を並列に計算することが可能である．

「挟み込み計算」では，根の近傍に領域を限定することにより，パスの追跡本数を減少させ効率的に計算できるよう工夫されている．しかし，より多変数・高次元の問題に対しては，どうしても追跡するパスの本数が多くなりアルゴリズム中の処理のうちほとんどの部分がパス追跡に費やされてしまう．そこで，パス追跡の効率化のためにPCグリッドコンピュータを利用することにした．そのために，全体の処理過程をPCグリッドコンピュータの利用を考慮して変更を行う．具体的には，前述の「挟み込み計算」の手順中の

1. パス追跡に使用する初期値の計算
2. 初期値を一つ利用してパスを追跡
3. パス追跡の打ち切り条件を考慮しつつ終値までパスを追跡
4. 根の近似値から重複度計算のための比を計算

の処理を並列で計算することを考える．そして，この計算から得られた比の総和を計算すれば根の重複度を得ることができる．

5. 数値例

2変数の連立代数方程式

$$
\begin{aligned}
f_1 &= 7x^{25}+7x^{24}y-9x^{23}y^2+3x^{22}y^3+2x^{21}y^4\\
&\quad -8x^{20}y^5-3x^{19}y^6+10x^{18}y^7+10x^{17}y^8\\
&\quad -8x^{16}y^9+10x^{15}y^{10}-6x^{14}y^{11}+10x^{13}y^{12}\\
&\quad +10x^{12}y^{13}-8x^{11}y^{14}-9x^{10}y^{15}-5x^9y^{16}\\
&\quad -8x^8y^{17}+3x^7y^{18}+3x^5y^{20}-5x^4y^{21}\\
&\quad -5x^3y^{22}+2x^2y^{23}-6xy^{24}-2y^{25}\\
f_2 &= 4x^{40}+5x^{39}y-2x^{38}y^2-7x^{37}y^3+2x^{36}y^4\\
&\quad -4x^{35}y^5+x^{34}y^6+2x^{33}y^7-2x^{32}y^8\\
&\quad +x^{31}y^9-2x^{30}y^{10}+10x^{29}y^{11}+7x^{27}y^{13}\\
&\quad +10x^{26}y^{14}-6x^{25}y^{15}+10x^{24}y^{16}+x^{23}y^{17}\\
&\quad -9x^{22}y^{18}-3x^{21}y^{19}+3x^{20}y^{20}-7x^{18}y^{22}\\
&\quad -5x^{17}y^{23}+6x^{16}y^{24}-8x^{15}y^{25}+4x^{14}y^{26}\\
&\quad +6x^{13}y^{27}+9x^{12}y^{28}-6x^{11}y^{29}-9x^{10}y^{30}\\
&\quad -6x^9y^{31}+8x^8y^{32}-x^7y^{33}-3x^6y^{34}\\
&\quad -4x^4y^{36}-x^3y^{37}-9x^2y^{38}-5xy^{39}+9y^{40}
\end{aligned}
$$

の原点での重複度を計算する．この問題は，総数1000本のパスを追跡する必要がある．全てのパスの追跡を終了するまでの処理時間を，使用したPCの台数ごとに計測すると表1のようになった．

表1. PCの台数と計算時間の比較

PCの台数	1	4	8	10
計算時間(秒)	25316	6361	3223	2594

計算は，Pentium4 1.5GHz (メモリ512MB) のPCを10台利用し，100Mbpsのネットワークで接続した環境を利用している．

PCを複数台利用すると全体の処理時間はほぼ台数分の一の割合で減少しているが，厳密に台数分の一にはならず，若干大きな値になっている．これは，データ転送時のネットワーク通信におけるオーバヘッドが影響しているものと考えられる．

6. おわりに

「挟み込み計算」をPCグリッドコンピュータへ実装することを目的に，現在アルゴリズムの修正作業を行っている．現時点では，連立代数方程式の根の近似値を計算する部分のみグリッドコンピュータを利用しているが，最終的には，全ての処理をグリッドコンピュータで実現できるようにしたい．

計算例からも分かるように，近似値計算の部分では，確実に処理時間が短縮されているので，グリッド化が実現されればより多変数・高次元の問題への適用が有用であると考えられる．

アルゴリズムの改良においては，特に，初期値の計算・与え方・振り分け方と計算結果の集約の効率的な方法が今後の課題となる．

参考文献

[1] 鈴木秀男, 小林英恒: 一次分数変換を利用した近接根の分離方法とその誤差について, 情報処理学会論文誌, Vol.38, No.2, pp.180–191 (1997).

[2] Hidetsune Kobayashi, Hideo Suzuki, Yoshihiko Sakai: Numerical calculation of the multiplicity of a solution to algebraic equations, Mathematics of Computation, Vol.67 No.221, American Mathematical Society, pp. 257-270 (1998).

[3] 鈴木秀男, 小林英恒: 一次分数変換を利用した連立代数方程式の近接根の分離と擬局所化における誤差について, 情報処理学会論文誌, Vol.40, No.12, pp.4178–4192 (1999).

[4] 鈴木秀男, 小林英恒: ホモトピー法のパラメータに一次分数変換を適用した近接根問題の解法について, 情報処理学会論文誌, Vol.44, No.12, pp.3112–3122 (2003).

[5] Ninf-G , http://ninf.apgrid.org/

[6] Y. Tanaka, H. Nakada, S. Sekiguchi, T. Suzumura, and S. Matsuoka, "Ninf-G: A Reference Implementation of RPC-based Programming Middleware for Grid Computing", Journal of Grid Computing, Vol. 1, No. 1, pp. 41-51, Kluwer Academic Publishers, (June 2003).