2J-4

グリッドコンピューティングを用いた「挟み込み計算」の解法 - 根の振る舞いの解析に向けて -

Numerical Calculation of the Multiplicity on Grid Computing - For the Analysis of Behavior of a Root -

黒坂 翔一 * Shoichi Kurosaka　鈴木 秀男 † Hideo Suzuki　小林 英恒 ‡ Hidetsune Kobayashi　三浦 正浩 § Masahiro Miura

1. はじめに

連立代数方程式の重複度を求める最も素朴な方法は，代数的に因数分解を行い，そのべき数を調べる方法である．このような代数的解法では，主に数式処理が用いられるが，性質上次数が高くなると多くの時間と記憶領域が必要となる．

ここでは，数値解法的に連立代数方程式の重複度を求める計算手法として,「挟み込み計算」[1] を用いる．この計算手法は，与えられた連立代数方程式へ代数的処理を施し，対象とする根の近傍での数値解を計算することによって重複度をプラス方向とマイナス方向から挟み込む手法である．

文献 [3] では,「挟み込み計算」をグリッドコンピューティングへ実装することを目的に，アルゴリズムの修正作業を行っている．今回は「挟み込み計算」で並列化される部分を変更し，新たにグリッドコンピューティング上へ実装した．重複度を正しくかつ効率的に計算できることを確認したので，その変更について報告する．

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は，大規模な科学技術計算やデータを解析するために利用が広がっている．特に，タスク間に依存関係が少ない場合にグリッド化することで効率的に計算できることが分かっている．

グリッドコンピューティングを実現するミドルウェアとして，Globus Toolkit 上に構築されたミドルウェアである Ninf-G[4] を利用する．Ninf-G は，GridRPC を実現するシステムであり，非常に簡単なプログラミングインタフェースを通じて，ネットワーク上に配置された計算機を用い，効率的に処理を行えるように開発されている．今回構築したグリッド環境は以下の通りである．

表 1: 構築したグリッド環境

OS	Fedora Core 3
ミドルウェア	Globus Toolkit 4.0.1
	Ninf-G 2.4.0

「挟み込み計算」では，処理をパスごとに独立して実行できるため，それぞれの処理を並列に実行させることで処理の効率化を見込むことができる．

*能力開発総合大 東京校 情報技術科
†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

3. 「挟み込み計算」の概略

「挟み込み計算」[1] とは，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を考えればよい
3. 簡単な比の計算で重複度が求まる

などが挙げられる．

ここでは，以下の 2 変数の連立代数方程式を考える．

$$\begin{array}{rcl} f_1(x,y) & = & 0 \\ f_2(x,y) & = & 0 \end{array} \qquad (1)$$

式 (1) の連立代数方程式での「挟み込み計算」の概略を以下に述べる．

1. $h > 0$ を十分に小さい数とし，$f_1(x, y)$ の x に h を代入する．
2. $f_1(h, y)$ を y についての 1 変数の方程式とみなし，$f_1(h, y) = 0$ を満たす y を $y_1, \cdots, y_r$ とする．得られた $y_i(i = 1, \cdots, r)$ を用いて点 P_i を $P_i = (h, y_i)$ とする．(以下ステップ P と呼ぶ)
3. 点 P_i と外部に与えた任意の点 $O = (o_1, o_2)$ を通る直線を l_i とすれば l_i は,
$$y = \frac{o_2 - y_i}{o_1 - h}x + \frac{y_i o_1 - h o_2}{o_1 - h} \qquad (2)$$
となる．
4. 直線 l_i を用いて、$f_2(x, y)$ の y を消去し、得られた式を x についての 1 変数の方程式とみなし，これを満たす x を $x_i^{(1)}, \cdots, x_i^{(s_i)}$ とする．得られた $x_i^{(j)}(i = 1, \cdots, r,\ j = 1, \cdots, s_i)$ を用いて点 $Q_i^{(j)}$ を $Q_i^{(j)} = (x_i^{(j)}, ax_i^{(j)} + b)$ とする．ただし，$a = (o_2 - y_i)/(o_1 - h)$, $b = (y_i o_1 - h o_2)/(o_1 - h)$ とする．(以下ステップ Q と呼ぶ)

5. 上記で求めた近似値から，

$$\alpha_{ij} = \frac{\log|x_i^{(j)} - h|}{\log|h|} \tag{3}$$

を計算する．

6. 計算された α_{ij} の総和を計算すれば，これが重複度となる．

$$m = \sum \alpha_{ij} \tag{4}$$

上記の基本アルゴリズムに基づき，計算された重複度を修正したものを m^+ とする．h の符号を変え，同様に計算された重複度を修正したものを m^- とすれば，m^+ と m^- で真の重複度を挟み込むことができる．これが，「挟み込み計算」と呼ばれるゆえんである．重複度の修正については、文献 [1] を参照されたい．

「挟み込み計算」では，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．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．本研究では，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改良ホモトピー法を用いる [2]．

解くべき方程式 $f(x) = 0$ が与えられたとき，次数が同じで根が簡単に求められる方程式を $g(x) = 0$ とし，次数が同じ重み多項式を $p(x)$ とすると，改良されたホモトピーは，

$$\begin{aligned} h(x,t,\delta) &= (t+\delta)(1+\delta)tf(x) \\ &\quad +\delta(t+\delta)(1-t)g(x) \\ &\quad +\delta(1+\delta)(1-t)tp(x) \end{aligned} \tag{5}$$

になり [2]，$h(x,0,\delta) = 0$ と $g(x) = 0$ の根が同じで，$h(x,1,\delta) = 0$ と $f(x) = 0$ の根も同じになる．$t = 0$ に対応する $g(x_0) = 0$ を満たす x_0 を用い，$h(x,t,\delta) = 0$ を満たすパスを追跡すれば，$t = 1$ に対応する x_1 が $f(x_1) = 0$ を満たす．ここで $\delta > 0$ は小さい数とし，δ を小さくすれば，パス追跡の分解能が向上する．

4. 「挟み込み計算」のアルゴリズムの修正とその実装

「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に工夫されている．しかし，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ほとんどの部分がパスの追跡に費されてしまう．ホモトピー法は，パス追跡がそれぞれ独立して行えるため，このパス追跡をグリッドコンピューティングを用い，効率化を図る．

「挟み込み計算」では，ステップ P で計算される $y_1, \cdots, y_r$ が計算されなければ，それに対応するステップ Q での $x_i^{(1)}, \cdots, x_i^{(s_i)}$ を計算することができない．そこで，ステップ P の計算処理を並列に割り振り，割り振られた先でステップ Q も計算するように並列化した．これが今回の修正箇所である．

5. 数値例

2 変数の連立代数方程式

$$\begin{aligned} f_1 &= 2x^6 - 3x^5y + x^4y^2 - 2x^3y^3 \\ &\quad +x^2y^4 + 5xy^5 + y^6 \\ f_2 &= x^6 + x^5y - 3x^4y^2 - x^3y^3 \\ &\quad +4x^2y^4 - xy^5 - y^6 \end{aligned} \tag{6}$$

の原点での重複度を計算する．この問題は，総数 36 本のパスを追跡する必要がある．すべてのパス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 2 のようになる．

表 2: PC の台数と計算時間の比較

PC の台数	1	2	3	6
計算時間 (秒)	105	54	38	21

計算には，Pentium4 1.5GHz（メモリ 512MB）の PC を 6 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，処理時間はほぼ台数分の一の割合で減少していることが確認できる．

6. おわりに

「挟み込み計算」をグリッドコンピューティングへ実装することを目的とし，文献 [3] を基に並列化に改良を施した．今回採用した並列化のアルゴリズムで連立代数方程式の重複度を正確に計算できることが確認できた．また，処理も効率化され，計算時間もほぼ台数分の一という理論値に近い結果を得ることができた．

ただし，文献 [1] で述べているすべての機能はまだ実装していないので，今後の検討課題とする．

参考文献

[1] 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).

[2] 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の解法について，情報処理学会論文誌，Vol.44, No.12, pp.3112–3122(2003).

[3] 鈴木秀男，小林英恒，三浦正浩: 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 - , FIT2004 講演論文集, pp.33–34(2004).

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