1D-3

Characteristic Set を求めるための Wu の方法の並列化について

白石 啓一 † 那須 英正 ‡ 野田 松太郎 ‡
詫間電波工業高等専門学校 † 愛媛大学 工学部 ‡

1 はじめに

連立代数方程式を正確に解くことは、制御・計算機援用設計 (CAD) 等多くの分野において重要である。その手法として、数式処理を用いた Gröbner 基底計算を基礎にした方法、消去法、 Characteristic set を計算する Wu の方法 [1] を利用する方法などが研究されている。

Gröbner 基底計算等が十分に研究され、その高速化がはかられているのに対し、Wu の方法に対する考察やインプリメントは十分ではない。そこで、我々は、並列処理の利用による高速化を考えた。 Wu の方法の基本演算は多項式の擬除算であるため、並列処理の利用による高速化が比較的容易であると考えられる。今回、我々は Wu の方法を並列計算機 AP3000 上で動く数式処理システム Risa/Asir 上に実現した。

2 Wu の方法

Wu の方法とは、多項式集合 PS の characteristic set CS を計算する方法である。CS は、 L-zero$(\cdot)$ で多項式集合の零点を表すと、

$$L\text{-zero}(CS/J) \subseteq L\text{-zero}(PS) \subseteq L\text{-zero}(CS)$$

を満たす。ここで、 $J=\prod_{f\in CS}\mathrm{ini}(f)$ 、$\mathrm{ini}(\cdot)$ は多項式集合の主係数を表す。

アルゴリズムを示す前に、まず、用語を簡単に説明する。 Wu の方法では、多変数多項式集合を扱う。この多変数多項式に含まれる変数には順序が決められている。以後、

$$x_1 \prec x_2 \prec \cdots \prec x_n$$

と考える。ある多項式に含まれる変数のうち、順序最大 (添数最大) の変数を主変数 (lvar) と呼ぶ。

非零多項式の有限個の列 $P_1, P_2, \ldots, P_r$ が、次の 2 つの条件のうち、どちらか一方を満たしているとき、この多項式集合を ascending set と呼ぶ。

1. $r=1$ かつ P_1 は定数
2. $\mathrm{lvar}(P_1) \prec \mathrm{lvar}(P_2) \prec \cdots \prec \mathrm{lvar}(P_r)$ 、かつ、整数の対 $(i,j)(0<i<j\leq r)$ それぞれについて、多項式 P_i,P_j が次の条件を満たす
   P_i の主変数を x_i とすると、

$$\deg_{x_i} P_i > \deg_{x_i} P_j$$

CS を求める Wu の方法のアルゴリズムは以下のように記述される。

1. PS より ascending set の条件を満たすように多項式集合 (PS の部分集合) を選択し、CS とする。
2. PS の CS に関する擬剰余を取り、 RS とする。
3. $RS=\emptyset$ ならば終了。$RS\neq\emptyset$ ならば、 $PS = PS\cup RS$ とし、 1. へ。

3 Wu の方法の並列化

2 節で示した Wu の方法のアルゴリズムの 2 を詳しく見ると、次のように書くことができる。

1. $RS=\emptyset$
2. $i=1,2,\ldots,n$ について
   $RS = RS \cup \text{pseudo-remainder}(PS_i, CS)$

ここで、 $PS=\{PS_1, PS_2, \ldots, PS_n\}$ である。

今回、我々は最も素朴な方法で、 Wu の方法を並列化した。即ち、上記 2 において、 PS_i の CS に関する擬剰余演算は、互いに独立に行うことができるので、これらを m 台のプロセッサに分散させて並列に処理

Implementation of parallel Wu's Method
K.Shiraishi, H.Nasu, M.T.Noda
Takuma National College of Tech., 551 Kohda, Takuma, Mitoyo, Kagawa 769–1192, Japan
Ehime Univ., 3 Bunkyocho, Matsuyama, Ehime 790–8577, Japan

させた。並列計算のモデルとしてマスタ–ワーカ・モデルを用い、データ交換にはメッセージ・パッシング方式を用いた。マスタでも計算させることができるが、今回、マスタでは計算しないこととした (図 1)。メッセージ送信、計算 (逐次処理を行った場合)、メッセージ受信の時間をそれぞれを T_S, T_C, T_R とすると、並列計算した場合の計算時間 T_p の上限値は

$$T_p \leq m \cdot T_S + T_C/m + m \cdot T_R$$

で表される。 T_p が T_C より小さければ、並列処理を行うことにより高速化される。

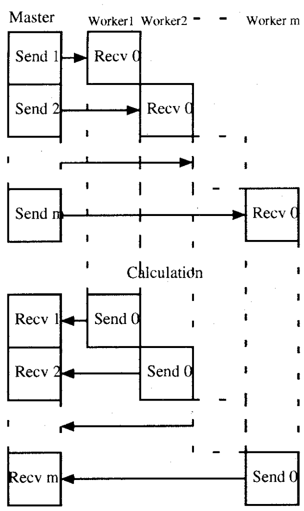

図 1: マスタ–ワーカ・モデル

4 実験と考察

例題として次の問題を挙げる。

$$\begin{aligned} PS \;=\; & \{y^2z + 2xyt - 2x - z, -x^3z + 4xy^2z + 4x^2yt \\ & +2y^3t + 4x^2 - 10y^2 + 4xz - 10yt + 2, \\ & 2yzt + xt^2 - x - 2z, -xz^3 + 4yz^2t + 4xzt^2 \\ & +2yt^3 + 4xz + 4z^2 - 10yt - 10t^2 + 2\} \end{aligned}$$

$$t \prec x \prec y \prec z$$

計算は AP3000(ノード: UltraSPARCII 360MHz×2, メモリ 640MB, 6 ノード/ 1 パーティション) 上で行い、並列計算ではプロセス数を 2,4,6 へ変化させた。計算時間は time コマンドで計り、実時間により比較した (表 1)。実験結果より、逐次処理よりも並列処理が高速であり、プロセス数を増やす程、高速になることが分かる。プロセス数が 2 の場合、逐次処理より並列処理が遅い。これは、ワーカが 1 であるため、擬剰余演算部が並列に計算されておらず、通信時間が余分にかかっているためである。

表 1: 計算時間 (単位: 秒)

逐次	並列 (プロセス数)		
	2	4	6
52.4	54.9	31.6	24.5

5 おわりに

Characteristic Set を求めるための Wu の方法を並列化した。実際の計算例を通しても、並列化による高速性を示すことが出来た。並列化における問題点は、分散させた計算の粒度が小さすぎる場合にある。この場合には十分な性能を期待できない。今後の課題として、

- 分散させた計算の粒度が大きくなるように、並列化部分を見直す
- データが小さい場合 (扱う多項式が少ない場合)、全てマスタで計算を行わせる
- AP3000 版 Risa/Asir での通信速度を調査し、効率の良い並列化を行う

が挙げられる。また、実際の工学問題に応用するには、浮動小数係数の多項式を扱えるようにする必要がある。数値数式融合アルゴリズムを取り入れていきたい。

参考文献

[1] Wu Wen-tsün: A Mechanization Method of Equations-solving and Theorem-proving, Advances in Computing Research, **6**, 1992, 103–138