総和・総乗演算式の自動微分手法の開発

3C−5　○野中　久典[1]　小林　康弘[1]　田村　正義[2]
(株)日立製作所　[1]エネルギー研究所，[2]ソフトウエア工場

1．はじめに

非線形最適化プログラムの入力データの作成を、記号処理技術を用いて支援する数式入力機能を開発した[1]。数式入力機能の処理の概要を図1に示す。本機能は、①高速自動微分法（FAD：Fast Automatic Differentiation）[2]を用いて最適化問題の目的関数・制約関数を自動微分する処理、および②最適化計算の実行時に必要とされる関数の定義領域に関する制約条件を自動抽出する処理を特徴とする。本報告では、上記①の機能に関し開発した、総和・総乗演算式を効率良く偏微分する手法について述べる。

2．総和・総乗演算式の自動微分

非線形最適化問題の目的関数・制約関数を記述する際に、総和・総乗演算式は頻繁に用いられる。数式入力機能には、これらの偏導関数を効率良く作成することが要求される。本機能において偏導関数作成手法として用いられている高速自動微分法の実行に際しては、偏微分すべき関数を、その計算過程を単項または二項演算からなる基本的な計算ステップの列として表現した計算グラフと呼ばれる記述形式に変換する必要がある。総和・総乗演算式は、計算グラフで直接表現することができないため、従来は、先ずこれらを和または積の形に展開して計算グラフを作成し、その後高速自動微分法を実行するという段階的な処理を行っていた。このため、特に総和・総乗演算式における繰返しの回数が大きくなった場合に、偏導関数作成に要する計算量およびメモリ量が極めて大きくなるという問題点があった。

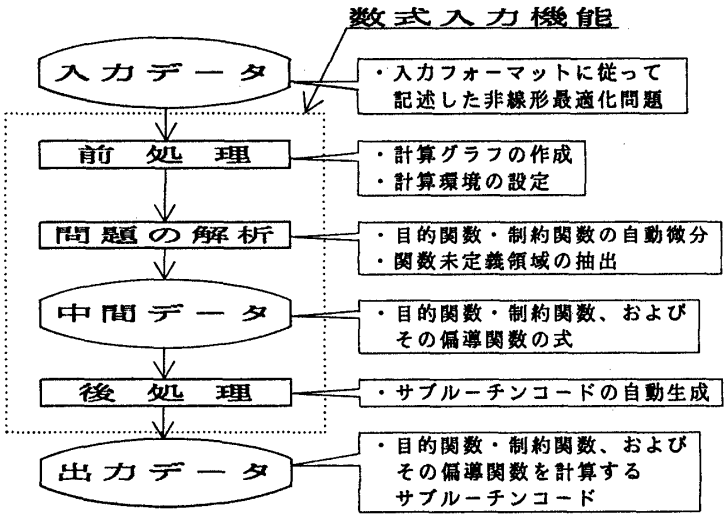

図1　数式入力機能の処理の概要

3．総和・総乗演算式を展開せずに微分する手法

上記の問題点を解決する目的で、総和・総乗演算式を展開せずに偏微分する手法を開発した。ここでは本手法を非展開微分法、従来法を展開微分法と呼称する。以下、非展開微分法を総和演算式の場合について説明する。総和演算式Fの一般形は次の通りである。

$$F = \sum_{i=imin}^{imax} f(i) \qquad (i：imin \leqq i \leqq imax なる整数)．$$

f(i)は連続微分可能な関数であり、

$$f(i) = f(Xi_1, Xi_2, \cdots, Xi_j, \cdots, Xi_{jmax}) \qquad (j：1 \leqq j \leqq jmax なる整数)．$$

である。ここでXi$_j$は、f(i)の式中に登場するj番目の入力変数を表す。

任意の入力変数Xkに関するFの偏導関数は次式で与えられる。

$$\frac{\partial F}{\partial Xk} = \sum_{i=imin}^{imax} \sum_{j=1}^{jmax} \frac{\partial Xi_j}{\partial Xk} \cdot \frac{\partial f(i)}{\partial Xi_j}$$

Development of Automatic Differentiation Method for Summation and Product Function.

Hisanori Nonaka, Yasuhiro Kobayashi, Masayoshi Tamura : Hitachi, Ltd.

$$= \sum_{i=imin}^{imax} \sum_{j=1}^{jmax} DELTA(k=i_j) \cdot \frac{\partial f(i)}{\partial Xi_j} .$$

ここで、DELTA(Cond) は、条件Condが成立した場合に１、しなかった場合に０となる関数であると定義する。本手法によると、ｆ(ｉ)のＸｉ$_j$に関する偏導関数をあらかじめ高速自動微分法を用いて求めておき、これを上式に代入することで、効率的にＦの偏導関数を作成することができる。

４．非展開微分法の評価

総和演算式の非展開微分法を、（ｉ）偏導関数の作成に要する時間、および（ii）偏導関数の作成に要するメモリ量の２項目に関して展開微分法と比較し評価した。評価のために用いた例題は次の３関数である。

$$F1 = \sum_{i=1}^{imax} X(i).$$

$$F2 = \sum_{i=2}^{imax} X(3)\cdot X(i)\cdot X(i+1)\cdot X(i-1).$$

$$F3 = \sum_{i=1}^{imax} [100\cdot (X(i+1)-X(i)^2)^2+(1-X(i))^2].$$

評価のためのパラメータは、繰返しの回数imax、すなわち入力変数の数である。なお、比較に用いたプログラムは、約３ＭＩＰＳの処理能力を持つワークステーション上に、Common-Lispを記述言語として実現している。比較の結果を図２から図４に示す。これらより、関数の形が特別に単純であるＦ１を除いて、非展開微分法が展開微分法より高速に、かつ少ないメモリ量で総和演算式の偏導関数を求めていることが分かる。特に、Rosenbrock関数の総和演算式であるＦ３においては、その差は２倍以上となる。また、入力変数の数が増えるに従って非展開微分法が有利になる傾向が認められる。

５．まとめ

非線形最適化問題の中に頻繁に登場する総和・総乗演算式の偏導関数の式を効率良く作成する、非展開微分法を開発した。本手法を（ｉ）偏導関数の作成に要する時間、および（ii）偏導関数の作成に要するメモリ量の２項目に関して従来の微分法と比較して定量的に評価し、本手法の有効性を確認した。

参考文献

［１］野中，小林，田村：高速自動微分法を用いた最適化プログラム用数式入力機能の開発：情報処理学会第39回全国大会講演論文集，5L-9 (1989)

［２］伊理，土屋，昼：偏導関数計算と丸め誤差推定の自動化の大規模非線形方程式系への応用：情報処理，Vol.26，No.11 (1985)

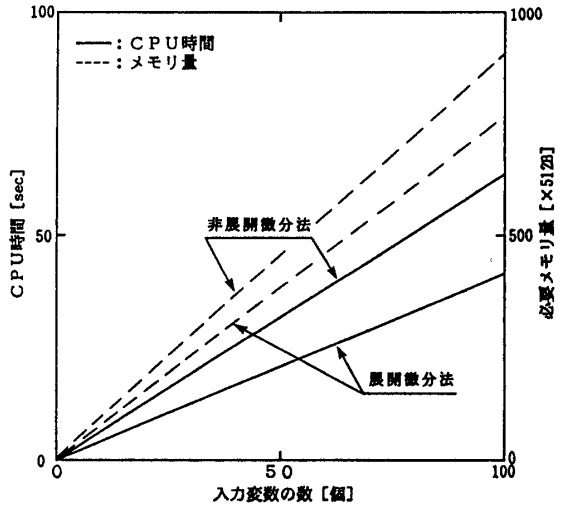

図２　Ｆ１の偏微分の要するＣＰＵ時間とメモリ量

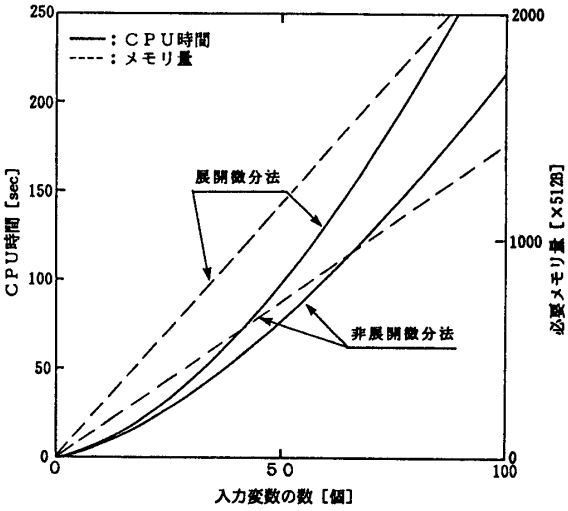

図３　Ｆ２の偏微分の要するＣＰＵ時間とメモリ量

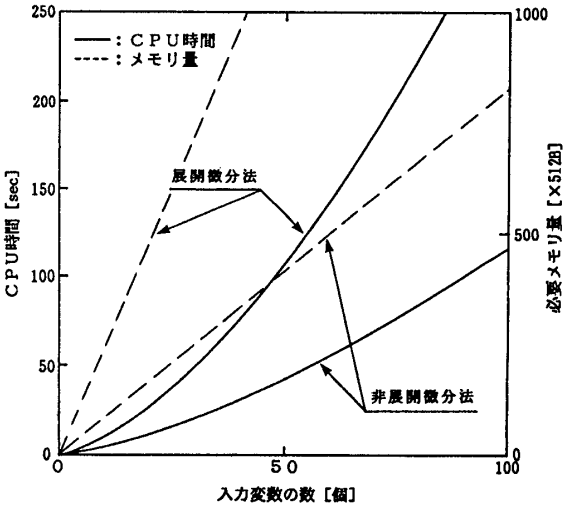

図４　Ｆ３の偏微分の要するＣＰＵ時間とメモリ量