

グラフの変形を用いた偏導関数の計算過程の導出†

吉 田 利 信††

従来、偏導関数値を必要とする数値計算には数値微分法が用いられてきた。この方法では、一つの関数のすべての偏導関数値が、変数の数に比例する回数の関数の計算によって得られる。最近、計算グラフを用いた自動微分法が提案され、一つの関数のすべての偏導関数値が、変数の数によらずに、関数自身を計算する計算量のたかだか定数倍の計算量で求められることが示された。しかしこの自動微分法を用いても、複数の関数のすべての偏導関数値を計算する場合には、関数の数または変数の数に比例する計算量が必要になる。そこで本論文では、自動微分法よりも少ない計算量の計算過程が得られる方法、すなわち、計算グラフとは別に要素的偏導関数値を辺に持つグラフ（随伴グラフ）を導入し、その随伴グラフを変形する操作を繰り返すことによって偏導関数の計算過程を順次計算グラフに追加する方法を提案する。まず、従来の自動微分法に対応する随伴グラフ上の算法を基本算法として定義し、基本算法を施したときに計算グラフに追加される計算過程の計算量を評価する。次に、随伴グラフに対する基本変形を定義し、基本算法を直接施した場合と基本変形を施してから基本算法を施した場合について、追加される計算過程の計算量を比較する。この評価を用いて、一連の基本変形を施す方法が基本算法を直接施す方法よりも少ない計算量の計算過程を導出することを示す。

1. ま え が き

数値計算の算法には偏導関数値を必要とするものがあり、従来これらの偏導関数値の計算には数値微分法や数式処理が用いられてきた。しかし数値微分法には、一つの関数のすべての偏導関数値を求めるのに変数の数に比例する回数の関数の計算が必要であること、十分な精度が得られにくいことなどの欠点がある。一方、数式処理には精度の良い偏導関数値が求められるという長所があるが、関数が少し複雑になるとその偏導関数を表す数式が膨大なものとなるという欠点がある。

以上の数値微分法や数式微分法に対して、関数を四則演算や初等関数などの基本演算に分解し、その関数をこれらの合成関数とみなし、その全微分を合成関数の偏微分の規則によって計算するという方法がある^{1)~3)}。この方法によって、一つの関数の全微分が関数自身を計算する計算量のたかだか定数倍の計算量で求められる。しかし、すべての変数に関する偏導関数値を求めるには変数の数に比例する計算量が必要となる。

最近、Baur ら⁴⁾、Kim ら^{5),6)}、伊理⁷⁾、Sawyer⁸⁾によって、一つの関数のすべての変数に関する偏導関数値が関数自身を計算する計算量のたかだか定数倍の計

算量で求められることがそれぞれ独立に示された。伊理⁷⁾は、関数の計算過程を表す計算グラフの辺に要素的偏導関数値を置き、関数側から変数側へ、あるいは、変数側から関数側へ計算グラフをたどりながら要素的偏導関数値の経路に沿った積（複数の経路があるときはそれらの積の和）を節点に置き、最終的に変数あるいは関数を表すすべての節点上に偏導関数値を得る算法（以下、自動微分法と呼ぶ）を示した。また、計算グラフを関数側から変数側へ逆にたどることによって、一つの関数のすべての変数に関する偏導関数値が関数自身を計算する計算量のたかだか定数倍の計算量で求められることを示した。さらに伊理ら^{11),12)}、土谷ら¹³⁾は、この自動微分法によって、 m 変数の n 次元ベクトル関数のヤコビ行列が関数自身の $1+3m$ 倍以下または $1+3n$ 倍以下の計算量で、また、ヤコビ行列とベクトルとの積が関数自身の 4 倍以下の計算量で求められることを示した。山下ら¹⁴⁾は、計算グラフを分割し、分割された個々の部分の偏導関数値を求め、それらを合成することによって、単に自動微分法を用いるよりも計算量が少なくなる例を示した。Volin ら¹⁰⁾は、計算グラフを階層的に分割し、計算グラフを変数側から関数側へたどる方法と関数側から変数側へたどる方法とを融合した方法を示し、計算量と記憶領域の必要量を解析した。また、以上の偏導関数の自動計算法を実現するシステムも既にいくつか作られている^{8),9),14),15)}。

本論文では、山下ら、Volin らに関連して、複数の関数のすべての偏導関数値を計算する場合について、

† Derivation of a Computational Process for Partial Derivatives of Functions Using Transformations of a Graph by TOSHINOBU YOSHIDA (Department of Electrical Engineering, Faculty of Engineering, Chiba University).

†† 千葉大学工学部電気工学科

自動微分法よりも少ない計算量の計算過程を得る方法について考察する。この方法では計算グラフとは別に要素的偏導関数値を辺に持つグラフ（以下、このグラフを随伴グラフと呼ぶ）を導入し、その随伴グラフを変形する操作を繰返すことによって偏導関数の計算過程を順次計算グラフに追加する。計算グラフの定義を2.1節に、随伴グラフの定義を2.2節に示す。2.3節では計算グラフ上の自動微分法を随伴グラフ上の基本算法として定義する。2.4節では随伴グラフに対して基本算法を施したときに追加される計算グラフ上の計算過程の計算量を評価する。3.1節では随伴グラフに対する基本的な変形操作を定義し、それぞれの基本変形の前後において、基本算法によって追加される計算過程の計算量がどのように変化するか調べる。3.2節では随伴グラフに対して一連の基本変形を施すことによって、基本算法を直接施した場合よりも少ない計算量の偏導関数の計算過程が導出される例を示す。3.3節では随伴グラフを変形する方法によって導出される偏導関数の計算過程の計算量について考察する。

2. 随伴グラフと基本算法

2.1 計算グラフの定義

いくつかの関数を計算する計算過程に対して、閉路のない有向グラフ $CG=G(V, E)$ を次のように構成し、これを計算グラフと呼ぶ。

- (1) V および E をともに空集合とする。
- (2) 計算過程に現れる演算の順序ですべての演算について以下を行う。
 - (2.1) 演算を $v := \Psi(u_1, u_2, \dots, u_k)$ とする。
 - (2.2) 各 u_i について以下を行う。
 - (2.2.1) もし u_i が定数または入力変数であり、かつ、 u_i を表す節点 u_i が V 中に存在していないならば u_i を表す節点 u_i を生成し、 $V := V \cup \{u_i\}$ とする。
 - (2.2.2) v を表す節点 v を生成し、 $V := V \cup \{v\}$ とする。
 - (2.3) 節点 v に対して v を求めるのに用いられた演算子 Ψ を対応させ、 $\omega(v) := \Psi$ のように表す。
 - (2.4) 各 u_i について以下を行う。
 - (2.4.1) 節点 u_i から節点 v への有向辺 e_i を生成し、 $E := E \cup \{e_i\}$ とする。

このように構成されるグラフが閉路を持たないこと、一つの計算過程から計算グラフが一意に定まると、一つの計算グラフを与える計算過程が複数存在し

得ることに注意する。また、ある計算過程によって計算グラフが構成されているとき、さらに計算過程が与えられると上に定義した手順によって計算グラフが拡張されることに注意する（以下、このことを“計算過程が計算グラフに追加される”あるいは“計算過程を計算グラフに追加する”と表現する）。

たとえば、

$$\begin{aligned} f_1 &= (x_1 * x_2 + \sin(x_3)) * ((x_1 * x_2) * \sin(x_2 - x_1)), \\ f_2 &= (x_1 * x_2 + \sin(x_3)) * (\sin(x_2 - x_1) + \sin(x_3)), \\ f_3 &= (x_1 * x_2) * \sin(x_3) + \sin(x_2 - x_1), \end{aligned} \quad (1)$$

の計算過程が図1のように与えられているとき、その計算グラフ $CG=G(V, E)$ は図2のように表される。図2の各節点 v にその演算子 $\omega(v)$ を示す。

2.2 随伴グラフの定義

演算 $v := \Psi(u_1, u_2, \dots, u_k)$ に対して、 v の u_i に関する偏導関数 $\partial v / \partial u_i$ を要素的偏導関数と呼ぶ。計算グラフ $CG=G(V, E)$ に対して、 V の部分集合 V_s を独立変数、 V の部分集合 V_f を従属変数として、要素的偏導関数値を辺に持つ随伴グラフ $DG=D(CG, V_s, V_f)=G(V^*, E^*)$ を次のように構成する。

- (1) CG において独立変数 V_s 中の節点から出て従属変数 V_f 中の節点へ入るすべての経路上の節点全体を V' 、辺全体を E' とする。

$v_1 = x_1 * x_2$
 $v_2 = \sin(x_3)$
 $v_3 = v_1 + v_2$
 $v_4 = x_2 - x_1$
 $v_5 = \sin(v_4)$
 $v_6 = v_1 * v_5$
 $f_1 = v_3 * v_6$
 $v_7 = v_3 + v_5$
 $f_2 = v_3 * v_7$
 $v_8 = v_1 * v_2$
 $f_3 = v_8 + v_5$

図1 計算過程

Fig. 1 Computational process.

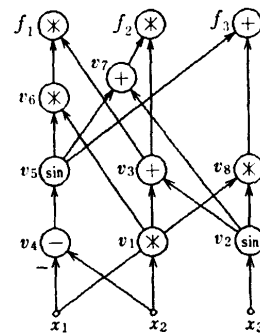


図2 計算グラフ CG

Fig. 2 Computational graph CG.

(2) V' のすべての節点 v に対して節点 v° を生成しその全体を V° とする.

(3) E° を空集合とする.

(4) E' のすべての辺 e に対して以下を行う.

(4.1) e の始点 $u \in V'$ に対応する V° の節点を u° , 終点 $v \in V'$ に対応する V° の節点を v° とする.

(4.2) u° を始点, v° を終点とする有向辺 e° を生成し, $E^\circ := E^\circ \cup \{e^\circ\}$ とする.

(4.3) もし $\partial v / \partial u$ および $-\partial v / \partial u$ を表す CG の節点が存在しないならば, そのどちらかを計算する計算過程を CG に追加する.

(4.4) w を $\partial v / \partial u$ あるいは $-\partial v / \partial u$ を表す CG の節点とする.

(4.5) もし w が $\partial v / \partial u$ を表す節点であるならば $sw := +$ とし, そうでないならば $sw := -$ とする.

(4.6) 辺 e° に対して符号 sw と CG の節点 w との組を対応させ, $d(e^\circ) := (sw, w)$ と表す.

たとえば, 図2の計算グラフ CG に対して, 独立変数を $V_x = \{x_1, x_2\}$, 従属変数を $V_f = \{f_1, f_2\}$ としたときの随伴グラフ DG は図3ようになる. このとき, 定数 $+1$ を表す節点1が CG に付け加えられる. 辺 e_3, e_5, e_{11} に対しては要素的偏導関数値が $d(e_3) = d(e_5) = d(e_{11}) = (+, 1)$, 辺 e_4 に対しては $d(e_4) = (-, 1)$ と表される. 辺 e_6 に関しては要素的偏導関数値 $\partial v_5 / \partial v_4 = \cos(v_4)$ を計算する計算過程が CG に追加される (図5参照). この演算結果を表す節点を v_9 とすると, 辺 e_6 に対する要素的偏導関数値は $d(e_6) = (+, v_9)$ と表される. ほかの辺については CG に存在して

いる節点を用いられ, たとえば, $d(e_1) = (+, x_2)$ などのように表される.

計算グラフ $CG = G(V, E)$ の随伴グラフ $DG = D(CG, V_x, V_f)$ に対して, i 番目の独立変数 x_i のみを独立変数とする随伴グラフを $DG_i = D(CG, \{x_i\}, V_f)$ とする. このとき図3の DG は DG_1 と DG_2 を重ね合わせたものとみることができる. DG の節点 v° が k 個の DG_i に現れるとき v° を多重度 k の節点, また同様に, DG の辺 e° が k 個の DG_i に現れるとき e° を多重度 k の辺と呼ぶ.

2.3 随伴グラフ上の基本算法

偏導関数値の自動計算法として, 計算グラフを入力変数側から関数側へたどる算法と関数側から入力変数側へたどる算法とが既に示されているが^{7), 8)}, これらの算法は計算グラフ上の経路に沿った要素的偏導関数値の積の和を計算するという点で本質的には同じである. ここでは変数側から関数側へたどる自動微分法を, 随伴グラフ上を独立変数側から従属変数側へたどる基本算法として定義する. 従属変数側から独立変数側へたどる算法は以下に示す算法と全く同様に定式化され, 計算量の評価も同様に行うことができる.

変数 v が演算 $v = \Psi(u_1, u_2, \dots)$ によって定義されている場合, $\partial v / \partial x$ は合成関数の微分の規則を用いて変数 u_i の x に関する偏導関数値 $\partial u_i / \partial x$ と要素的偏導関数値 $\partial v / \partial u_i$ とから

$$\partial v / \partial x = \partial v / \partial u_1 * \partial u_1 / \partial x + \partial v / \partial u_2 * \partial u_2 / \partial x + \dots \quad (2)$$

のように計算される. 前節では符号と計算グラフ CG の節点との組を用いて要素的偏導関数値を表し, これを随伴グラフ DG の辺 e° に対して $d(e^\circ)$ と記述した. ここでは符号と $\partial v / \partial x$ または $-\partial v / \partial x$ を表す CG の節点との組で変数 v の独立変数 x に関する偏導関数値 $\partial v / \partial x$ を表す. この組を DG の節点 u_i° に $\partial u_i / \partial x$ が $s(u_i^\circ)$ として, 辺 e_i° に $\partial v / \partial u_i$ が $d(e_i^\circ)$ として表されているとき, 式(2)から $\partial v / \partial x$ は節点 v° に $s(v^\circ)$ として次のように記述される.

$$s(v^\circ) := d(e_1^\circ) * s(u_1^\circ) + d(e_2^\circ) * s(u_2^\circ) + \dots \quad (3)$$

ただし, 符号と CG の節点の組の間の和および積は次のように定義される.

$$(sv_1, v_1) + (sv_2, v_2) = \begin{cases} (sv_1, v_1 + v_2) & \text{if } sv_1 = sv_2, \\ (+, v_1 - v_2) & \text{if } sv_1 \neq sv_2, sv_1 = +, \\ (+, v_2 - v_1) & \text{if } sv_1 \neq sv_2, sv_1 = -. \end{cases} \quad (4)$$

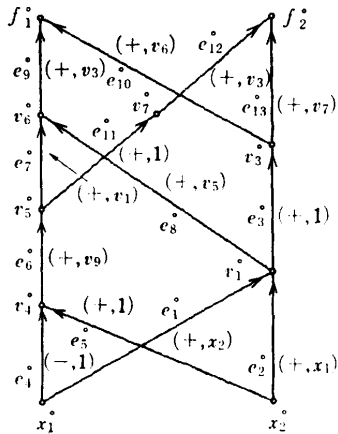


図3 随伴グラフ DG
Fig. 3 Associate graph DG .

$$(sv_1, v_1) * (sv_2, v_2) = (sv_1 * sv_2, v_1 * v_2). \quad (5)$$

ここで、これらの定義中の $v_1 + v_2$ などの演算が CG に追加されることに注意する。

一つの独立変数を持つ随伴グラフに対して、基本算法 1 を次のように定義する。

基本算法 1 (独立変数が一つの場合)

随伴グラフ $DG = D(CG, \{x\}, V_f) = G(V^\circ, E^\circ)$ に対して、次の操作を行う。

- (1) $s(x^\circ) := (+, 1)$ とする。
- (2) x° を除くすべての $v^\circ \in V^\circ$ について、計算過程の順序に従って式 (3) から $s(v^\circ)$ を求める。
- (3) すべての従属変数 f に対する DG の節点 f° について以下を行う。

- (3.1) $s(f^\circ) = (sv, v)$ とする。

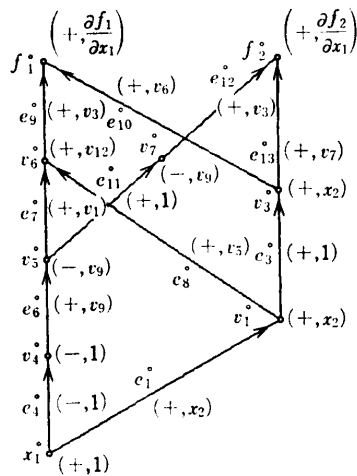


図 4 基本算法 1 の結果

Fig. 4 Result of the fundamental algorithm 1.

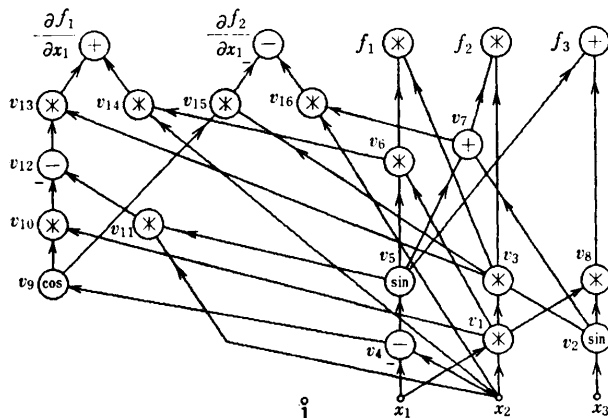


図 5 基本算法 1 によって拡張された計算グラフ CG

Fig. 5 Extended computational graph CG by the fundamental algorithm 1.

- (3.2) もし $sv = +$ ならば $\partial f / \partial x := v$, そうでなければ $\partial f / \partial x := -v$ とする。

ここで、基本算法 1 の (2) では計算過程の順序に従って $s(v^\circ)$ を求めているので、式 (3) によって $s(v^\circ)$ を求めているときにはすでに式 (3) 中の $s(u_i^\circ)$ は定義されていることに注意する。また、この過程で CG が順次拡張されることに注意する。

たとえば、図 2 の計算グラフ CG に対して、独立変数を x_1 としたときの随伴グラフ $DG_1 = D(CG, \{x_1\}, \{f_1, f_2\})$ を考える。この DG_1 に対して基本算法 1 を施すと図 4 のように各節点 v° に偏導関数値 $s(v^\circ)$ が得られ、図 5 に示すように CG が拡張される (図 4 の各辺 e° には要素的偏導関数値 $d(e^\circ)$ を示す)。

次に複数の独立変数を持つ随伴グラフに対して、基本算法 2 を以下のように定義する。

基本算法 2 (独立変数が複数の場合)

随伴グラフ $DG = D(CG, \{x_1, \dots, x_m\}, V_f)$ に対して、各 $DG_i = D(CG, \{x_i\}, V_f)$ ごとに基本算法 1 を行い、順次 CG を拡張する。

2.4 計算量の評価

計算グラフを用いた自動微分法の計算量について、伊理⁷⁾ は関数値および要素的偏導関数値を求める計算過程の計算を含め、その偏導関数を計算するために必要なすべての計算の計算量を評価している。ここでは基本変形 (第 3 章で導入する) による計算量の増減を評価するために、基本算法によって計算グラフに追加される計算過程の計算量のみについて評価する。

計算グラフ CG に対して、独立変数が一つの随伴グラフ $DG = D(CG, \{x\}, V_f)$ が与えられているとする。DG の節点の数を N_v 、辺の数を N_e 、要素的偏導関数値が $+1$ または -1 である辺の数を N_1 、要素的偏導関数値が $+1$ または -1 以外の辺で積の演算が不要な辺の数を N_* とする。このとき、基本算法 1 によって CG に追加される計算過程の計算量について以下の定理が成立する。

定理 基本算法 1 によって CG に追加される加減算の数 $N_+(DG)$ および乗算の数 $N_*(DG)$ は次のように表される。

$$N_+(DG) = N_e - (N_v - 1), \quad (6)$$

$$N_*(DG) = N_e - N_1 - N_*. \quad (7)$$

証明 独立変数に対応する節点 x° を除くすべての節点について (その節点に入る辺の数 -1) 回の加減算が必要であること、節点 x° を除くすべての節点に入る辺の数の和が N_e に等しいことから式

(6)が得られる. 要素的偏導関数値が +1 または -1 である辺では積の演算が不必要であり, また, 独立変数に対応する節点のように, $s(v^o)$ が (+,1) または (-,1) であることがあらかじめわかっている節点 v^o から出る辺に対しても, 積の演算は不必要である. したがって, N_+ と N_* の定義から式(7)が得られる.

たとえば, 図4のグラフ DG_1 では, $N_+=9$, $N_*=11$, $N_1=3$, $N_*=2$ であり, 式(6), (7)から $N_+(DG_1)=3$, $N_*(DG_1)=6$ が得られる (図5参照).

m 変数 n 次元ベクトル関数の計算グラフ CG およびその随伴グラフ DG に対して基本算法2を施したときのヤコビ行列の計算量 $N_+(DG)$, $N_*(DG)$ は, i 番目の独立変数 x_i に対応する DG_i に基本算法1を施したときの計算量を $N_+(DG_i)$, $N_*(DG_i)$ とすると, 次のように表される.

$$N_+(DG) = \sum_{i=1}^m N_+(DG_i), \quad N_*(DG) = \sum_{i=1}^m N_*(DG_i). \quad (8)$$

3. 随伴グラフの変形による偏導関数の計算過程の導出とその計算量の評価

3.1 基本変形と計算量

以下, m 変数 n 次元ベクトル関数の計算過程に対する計算グラフ CG とその随伴グラフ $DG=D(CG, V_*, V_f)$ を考える. ここで, V_* は m 個の独立変数を, V_f は n 個の関数 (従属変数) を表す. このベクトル関数のヤコビ行列を求める計算過程は基本算法1を m 回行うことによって CG に追加される. その計算量は, 式(8)の $N_+(DG)$, $N_*(DG)$ で表される. 以下に述べる DG の変形に伴って CG に追加される加減算の数を n_+ , 乗算の数を n_* とし, 変形後の随伴グラフを DG' とする. 変形による計算量の変化 δN_+ , δN_* を次のように定義する.

$$\begin{aligned} \delta N_+ &= N_+(DG') + n_+ - N_+(DG), \\ \delta N_* &= N_*(DG') + n_* - N_*(DG). \end{aligned} \quad (9)$$

以下に, 基本変形を示し, それぞれについて計算量の変化 δN_+ , δN_* を評価する.

(1) 1の辺の縮約a: 節点 v_1 から節点 v_2 への辺 e^o に要素的偏導関数値 +1 または -1 が対応し, e^o 以外の辺が v_1 から出ていないとき, 図6に示すように e^o を縮約し, v_1 に入るすべての辺が v_2 に入るようにする. この変形に伴って CG に追加される計算過程はなく, 変形後の計算量の変化もない. つまり, $\delta N_+=0$, $\delta N_*=0$ である.

(2) 1の辺の縮約b: 節点 v_1 から節点 v_2 への辺 e^o に要素的偏導関数値 +1 または -1 が対応し, e^o 以外の辺が v_2 へ入っていないとき, 図7に示すように e^o を縮約し, v_2 から出るすべての辺が v_1 から出るようにする. この変形の場合も $\delta N_+=0$, $\delta N_*=0$ である.

(3) 辺の統合: 図8に示すように節点 v_1 から節点 v_2 への辺 e_1 , e_2 があるとき, e_1 と e_2 の要素的偏導関数値の和を求める計算を CG に追加し, 辺 e_1 と辺 e_2 を統合し辺 e_1 とする. このとき, 辺 e_1 の要素的偏導関数値は,

$$d(e_1) := d(e_1) + d(e_2) \quad (10)$$

と表される. この変形に伴って CG に追加される計算過程の計算量は $n_+=1$, $n_*=0$ である. 節点 v_1 の多

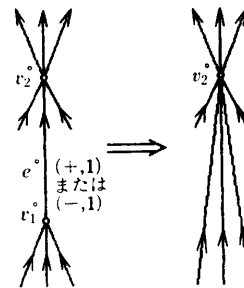


図6 1の辺の縮約a
Fig. 6 Reduction of 1's edge a.

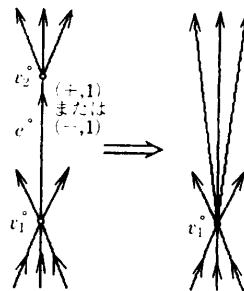


図7 1の辺の縮約b
Fig. 7 Reduction of 1's edge b.

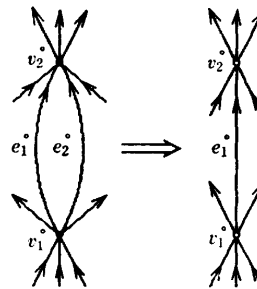


図8 辺の統合
Fig. 8 Unification of edges.

重度を k とすると, e_1, e_2 の多重度も k である. したがって, $\partial v_i / \partial x$ が $+1$ または -1 であることがあらかじめわかっている DG_i の数を l とすると,

$$\begin{aligned} \text{a) } d(e_1) \text{ と } d(e_2) \text{ のどちらも } (\pm, 1) \text{ でないとき,} \\ \delta N_+ = 1 - k, \delta N_* = l - k \end{aligned} \quad (11)$$

$$\text{b) } d(e_1) \text{ と } d(e_2) \text{ のどちらか一方が } (\pm, 1) \text{ であるとき,} \quad (12)$$

$$\delta N_+ = 1 - k, \delta N_* = 0$$

$$\text{c) } d(e_1) \text{ と } d(e_2) \text{ の両方が } (\pm, 1) \text{ であるとき,} \\ \delta N_+ = 1 - k, \delta N_* = k - l \quad (13)$$

が成立する. 特に v_i が独立変数であるときは $k=l=1$ となり上記のいずれの場合にも $\delta N_+ = \delta N_* = 0$ が成立することに注意する.

(4) 節点の削除: 図 9 に示すように節点 v_i から節点 v_2 への辺 e_i , 節点 v_2 から節点 v_3 への辺 e_2 があり, v_2 には e_1 と e_2 以外の辺が接続していないもの

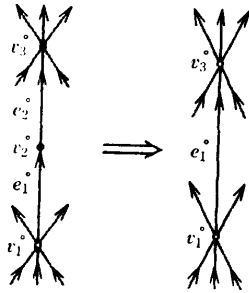


図 9 節点の削除

Fig. 9 Elimination of a node.

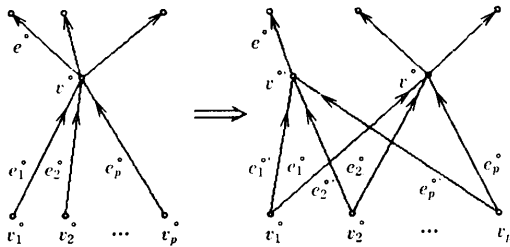


図 10 辺の分離 a

Fig. 10 Separation of an edge a.

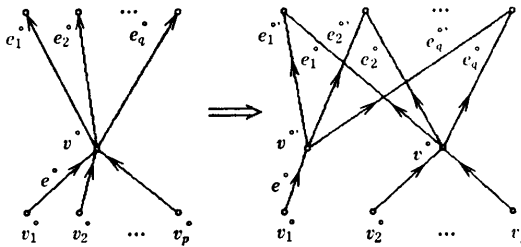


図 11 辺の分離 b

Fig. 11 Separation of an edge b.

とする. また, 1 の辺の縮約 a または b の変形が可能なのは, 先にその変形を行うことにし, e_1 と e_2 の要素的偏導関数値はどちらも ± 1 でないと仮定する. これらの要素的偏導関数値の積を求める計算を CG に追加し, 節点 v_2 を削除し, 新たに辺 e_1 を節点 v_1 から節点 v_3 への辺とする. このとき, 辺 e_1 の要素的偏導関数値は,

$$d(e_1) := d(e_1) * d(e_2) \quad (14)$$

と表される. この変形では $n_+ = 0, n_* = 1$ である. 節点 v_1 の多重度を k とすると,

$$\delta N_+ = 0, \delta N_* = 1 - k \quad (15)$$

が成立する. 特に v_i が独立変数であるときは $k=1$ となり $\delta N_+ = \delta N_* = 0$ となることに注意する.

(5) 辺の分離 a: 図 10 に示すように節点 v^o から複数の辺が出ているとき, そのうちの一边 e^o を他の辺から次のように分離する. すなわち, 節点 v^o と辺 e_i から辺 e_p までを複製し $v^o, e_1', \dots, e_p'$ とし, 辺 e^o が節点 v^o から出るようにする. この変形に伴って CG に追加される計算過程はない. つまり, $n_+ = 0, n_* = 0$ である. DG_i に対する基本算法 1 において, 式 (3) を用いて $s(v^o)$ を求めるときに必要な加減算の数を A_i , 乗算の数を M_i とするとき,

$$\delta N_+ = \sum_{i=1}^m A_i, \delta N_* = \sum_{i=1}^m M_i \quad (16)$$

が成立する. 特に $p=1$ で v_i が独立変数であるときは, すべての i について $A_i = M_i = 0$ となり $\delta N_+ = \delta N_* = 0$ が成立することに注意する.

(6) 辺の分離 b: 図 11 に示すように節点 v^o へ複数の辺が入っているとき, そのうちの一边 e^o を他の辺から次のように分離する. すなわち, 節点 v^o と辺 e_i から辺 e_p までを複製し $v^o, e_1', \dots, e_p'$ とし, 辺 e^o が節点 v^o へ入るようにする. この変形も $n_+ = 0, n_* = 0$ である. 辺 e^o の多重度を k とし, 辺 e_i から辺 e_p までの中で要素的偏導関数値が $+1$ または -1 となる辺の数を q_1 とする. DG_i が v_i を含み, かつ, v_2 から v_p までのいずれかを含むような DG_i の数を k' とする. また, 辺の分離を行った後, これらの DG_i の中で $s(v^o) = (\pm, 1)$ となる DG_i の数と $s(v^o) = (\pm, 1)$ となる DG_i の数の和を l とする. このとき,

$$\delta N_+ = (q-1) * k', \delta N_* = (q-q_1) * (k'-l) \quad (17)$$

が成立する. 特に v_i が独立変数であるときは $k'=0$ となり $\delta N_+ = \delta N_* = 0$ が成立することに注意する.

以上の基本変形のうち辺の統合と節点の削除だけが

計算量を減少させることが可能であること、多重度が高いほど大きく計算量を減少させることが可能なこと、および、独立変数に対応する節点を含む基本変形では計算量が変化しないことに注意する。

3.2 偏導関数の計算過程の導出例とその計算量

図3に示した随伴グラフ DG に対して一連の基本変形を施したときの計算量の変化を調べる。

(1) 辺 e_3 と e_{11} について1の縮約 b を行う。計算量の変化はない。

(2) 辺 e_7 について辺の分離 b を行う。すなわち、節点 v_6 と辺 e_9 を複製しそれぞれ v_6' , e_9' とし、 e_7 が v_6' に入るようにする。計算量の変化は $\delta N_+ = 0$, $\delta N_* = +2$ である。

(3) 節点 v_6 と v_6' について節点の削除を行う。このとき、 CG に $v_{10} = v_1 * v_3$ と $v_{11} = v_3 * v_5$ を追加し、辺 e_7 と e_8 がともに節点 f_1 に入るようにする。また、新たにそれぞれの要素的導関数値を $d(e_7) := (+, v_{10})$, $d(e_8) := (+, v_{11})$ と表す。これらの変形では計算量の変化はそれぞれ $\delta N_+ = 0$, $\delta N_* = -1$ である。

(4) 辺 e_8 と e_{10} について辺の統合を行い、それを e_8 とし e_{10} は削除する。このとき、 CG に $v_{12} := v_6 + v_{11}$ を追加し、 e_8 に対する要素的偏導関数値を新たに $d(e_8) := (+, v_{12})$ と表す。計算量の変化は $\delta N_+ = -1$, $\delta N_* = -2$ である。

(5) 辺 e_7 について辺の分離 a を行う。すなわち、節点 v_5 と辺 e_6 を複製しそれぞれ v_5' , e_6' とし、 e_7 が v_5' から出るようにする。このとき、 x_1 のみを独立変数とする随伴グラフ DG_1 において $s(v_4) = (-, 1)$, x_2 のみを独立変数とする随伴グラフ DG_2 において $s(v_4) = (+, 1)$ より、 $d(e_6) = (+, v_9)$ は $s(v_4)$ と演算する必要がない。つまり、式(16)における A_i と M_i はすべて0であり、 $\delta N_+ = 0$, $\delta N_* = 0$ が成立する。

(6) 節点 v_5 と v_5' について節点の削除を行う。このとき、 CG に $v_{13} = v_9 * v_{10}$ と $v_{14} = v_9 * v_9$ を追加し、 e_6' が f_1 に、 e_6 が f_2 に入るようにする。また、新たにそれぞれの要素的導関数値を $d(e_6') := (+, v_{13})$, $d(e_6) := (+, v_{14})$ と表す。これらの変形では計算量の変化はそれぞれ $\delta N_+ = 0$, $\delta N_* = -1$ である。

以上の変形の結果、図12に示す随伴グラフが得られる。この過程で1個の加算と4個の乗算が CG に追加されている。さらにこの随伴グラフに対して基本算法2を施すと4個の加減算と4個の乗算が CG に追加され、合計、5個の加減算と8個の乗算からなる計算過程(図13)が得られる。図3に示した DG に対

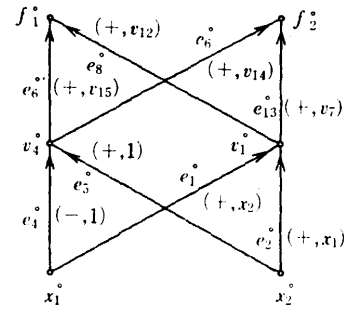


図12 一連の基本変形によって得られた随伴グラフ
Fig. 12 Associate graph derived by fundamental transformations.

$$\begin{aligned} v_{10} &= v_1 * v_3 \\ v_{11} &= v_3 * v_5 \\ v_{12} &= v_6 + v_{11} \\ v_{13} &= v_9 * v_{10} \\ v_{14} &= v_9 * v_9 \\ v_{15} &= x_2 * v_{13} \\ \partial f_1 / \partial x_1 &= v_{15} - v_{13} \\ v_{16} &= x_2 * v_7 \\ \partial f_1 / \partial x_1 &= v_{16} - v_{14} \\ v_{17} &= x_1 * v_{12} \\ \partial f_1 / \partial x_2 &= v_{15} + v_{17} \\ v_{18} &= x_1 * v_7 \\ \partial f_2 / \partial x_2 &= v_{14} + v_{18} \end{aligned}$$

図13 一連の基本変形と基本算法2によって得られた計算過程
Fig. 13 Computational process derived by fundamental transformations and the fundamental algorithm 2.

$$\begin{aligned} v_{10} &= v_1 * v_3 & v_{17} &= v_1 * v_9 \\ v_{11} &= x_2 * v_3 & v_{18} &= x_1 * v_9 \\ v_{12} &= v_{11} - v_{10} & v_{19} &= v_{17} + v_{18} \\ v_{13} &= v_9 * v_{12} & v_{20} &= v_9 * v_{19} \\ v_{14} &= x_2 * v_9 & v_{21} &= x_1 * v_9 \\ \partial f_1 / \partial x_1 &= v_{13} + v_{14} & \partial f_1 / \partial x_2 &= v_{20} + v_{21} \\ v_{15} &= v_9 * v_9 & v_{22} &= v_9 * v_9 \\ v_{16} &= x_1 * v_7 & v_{23} &= x_1 * v_7 \\ \partial f_2 / \partial x_1 &= v_{16} - v_{15} & \partial f_2 / \partial x_2 &= v_{22} + v_{23} \end{aligned}$$

図14 基本算法2のみによって得られた計算過程
Fig. 14 Computational process derived by the fundamental algorithm 2.

して基本算法2を直接施した場合、計算量の評価値は $N_+(DG) = 6$, $N_*(DG) = 12$ であり、図14に示す計算過程が CG に追加される。つまり、随伴グラフを変形する方法によって基本算法2を直接施すよりも加減算が1個、乗算が4個削減されており、計算量の変化の評価値の合計 $\delta N_+ = -1$, $\delta N_* = -4$ に一致している。

3.3 随伴グラフを変形する方法の評価

基本算法によって計算グラフ上に生成される偏導関数の計算過程と同じ計算過程を、随伴グラフに対して

基本変形のみを施すことによって得ることができる。これは独立変数に対応する節点を含む基本変形だけを用いることによって、基本算法1の(2)における符号と計算グラフの節点との組の間の演算を同じ順序で構成できることから容易に示される。したがってこのことは、随伴グラフを変形する方法が従来の自動微分法の導出する計算過程の計算量以下の計算量を持つ計算過程を必ず導出できることを示している。

次に随伴グラフを変形する方法によって大幅に計算量を削減できる極端な場合を示す。

$$t = \sum_{j=1}^m x_j, \quad (18)$$

$$f_i(t), i=1, 2, \dots, n. \quad (19)$$

これらの m 変数 n 次元ベクトル関数の計算過程を表す計算グラフを CG とする。独立変数を $x_1, x_2, \dots, x_m$ とする CG の随伴グラフを DG , 独立変数を t とする CG の随伴グラフを DG' とする。 DG に基本算法2を施したときの CG に追加される計算量を $N_+(DG)$, $N_*(DG)$ とし, DG' に基本算法1を施したときに CG に追加される計算量を $N_+(DG')$, $N_*(DG')$ とすると $\partial t / \partial x_j = 1$ であることから,

$$\begin{aligned} N_+(DG) &= m * N_+(DG'), \\ N_*(DG) &= m * N_*(DG') \end{aligned} \quad (20)$$

が成立する。一方, DG' に基本算法1を直接施したときに得られる計算過程と同じ計算過程を生成する基本変形の列が存在するので, その基本変形の列を用いて DG を変形する。変形された結果の随伴グラフを DG'' とする。このとき, 変形に伴って

$$n_+ = N_+(DG'), n_- = N_*(DG') \quad (21)$$

の計算量の計算過程が CG に追加されていること, および, DG'' の節点 t から出る辺はすべて従属変数に直接入り, それらの要素的偏導関数値が $\partial f_i / \partial t$ を表していることに注意する。したがって, DG'' に基本算法2を施したときの計算量は $\partial t / \partial x_j = 1$ であることから,

$$N_+(DG'') = 0, N_*(DG'') = 0 \quad (22)$$

となることが示される。

以上の式(21), 式(22)より, この特殊な m 変数 n 次元ベクトル関数のヤコビ行列の計算過程の計算量が, 随伴グラフの大きさのみに比例し変数の数にも関数の数にも依存しないことが示せた。

4. む す び

計算グラフに対して要素的偏導関数値を辺に持つ随

伴グラフを導入し, そのグラフを変形することによって偏導関数の計算過程を導出する方法を示した。多変数のベクトル関数のヤコビ行列を計算するとき, 従来の自動微分法では計算グラフ上の経路に沿った要素的偏導関数値の積と和の計算を変数の数あるいは関数の数の回数行う必要があった。一方, 随伴グラフを変形する方法では, 従来の自動微分法で得られる計算量以下の計算過程を得ることが必ずでき, 最良の場合, 随伴グラフの大きさに比例し, 変数の数や関数の数に依存しない計算量の計算過程が得られることを示した。

偏導関数値を用いる計算が多数回行われるとき, あるいは, 高速に計算されなければならないとき, その計算過程を導くために多くの時間が必要であっても計算量の少ない計算過程が得られることは重要である。

随伴グラフを変形する方法では多くの場合, 計算量を減少させる基本変形をいくつも続けて実行することはできない。途中で計算量を増加させる基本変形を行うことによって基本変形を続行することが可能になる。したがって計算量を最小にする一連の基本変形の列を決定することは容易ではない。計算量を最小にするような基本変形の列を自動的に生成する算法を定式化し, その算法を実現するシステムを作成すること, および, その算法自身の計算量の評価とその算法によって得られた偏導関数の計算過程の計算量の評価をすることが今後の課題である。

謝辞 有益な御助言をいただいた千葉大学星守助教に深謝いたします。日頃御指導賜わる同大学吉江清教授, 島倉信講師, 戸田英雄教授, 小野令美講師に感謝いたします。

参 考 文 献

- 1) Wengert, R. E.: A Simple Automatic Derivative Evaluation Program, *Comm. ACM*, Vol. 7, No. 8, pp. 463-464 (1964).
- 2) Kedem, G.: Automatic Differentiation of Computer Programs, *ACM Trans. Math. Softw.*, Vol. 6, No. 2, pp. 150-165 (1980).
- 3) Rall, L. B.: *Automatic Differentiation: Techniques and Applications*, Lecture Notes in Computer Science, Vol. 120, Springer-Verlag, Berlin (1981).
- 4) Baur, W. and Strassen, V.: The Complexity of Partial Derivatives, *Theor. Comput. Sci.*, Vol. 22, pp. 317-330 (1983).
- 5) Kim, K. V., Nesterov, Yu. E., Skokov, V. A. and Cherkasskii, B. V.: An Efficient Algorithm for Computing Derivatives and Extremal

- Problems, *Ekonomika i matematicheskie metody*, Vol. 20, No. 2, pp. 309-318 (1984).
- 6) Kim, K. V., Nesterov, Yu. E. and Cherkasskii, B. V.: An Estimate of the Effort in Computing the Gradient, *Soviet Math. Dokl.*, Vol. 29, No. 2, pp. 384-387 (1984).
- 7) Iri, M.: Simultaneous Computation of Functions Partial Derivatives and Estimates of Rounding Errors—Complexity and Practicality, *Jpn. J. Appl. Math.*, Vol. 1, No. 2, pp. 223-252 (1984).
- 8) Sawyer, J. W.: First Partial Differentiation by Computer with an Application to Categorical Data Analysis, *The American Statistician*, Vol. 38, No. 4, pp. 300-308 (1984).
- 9) 岩田憲和: 偏導関数計算の自動化, 東京大学大学院工学系研究科情報工学専門課程修士論文 (1984).
- 10) Volin, Y. M. and Ostrovskii, G. M.: Automatic Computation of Derivatives with the Use of the Multilevel Differentiating Technique—1. Algorithmic Basis, *Comp. Math. Appl.*, Vol. 11, No. 11, pp. 1099-1114 (1985).
- 11) 伊理正夫, 小野令美, 戸田英雄: 合成関数の高速微分法とその導関数を含む Runge-Kutta 系の常微分方程式数値解法公式への応用, 情報処理学会論文誌, Vol. 27, No. 4, pp. 389-396 (1987).
- 12) 伊理正夫, 久保田光一: 高速自動微分法とその周辺, 統計数理研究所研究報告「グラフ理論の数値計算への応用」(1987).
- 13) 土谷 隆, 笹山晋一: 非線形方程式系の解法に対する高速微分法の応用, 統計数理研究所研究報告「グラフ理論の数値計算への応用」(1987).
- 14) 山下 稔, 吉田利信: 計算グラフを用いた数値計算のための数式処理システムの設計, 統計数理研究所研究報告「グラフ理論の数値計算への応用」(1987).
- 15) 久保田光一, 伊理正夫: 高速自動微分法の定式化の試みと利用のためのシステム, 統計数理研究所研究報告「グラフ理論の数値計算への応用」(1987).

(昭和 61 年 12 月 3 日受付)

(昭和 62 年 9 月 9 日採録)



吉田 利信 (正会員)

1951 年生. 1978 年東京大学大学院博士課程修了 (計数工学). 工学博士. 千葉大学工学部電気工学科助手. 言語理解システム, 神経細胞の応答特性, 自動微分システムの研究に従事. 電子情報通信学会会員.