

# 区分的線形近似の誤差推定

宮城 亮<sup>†</sup> 久保田 光一<sup>‡</sup>

中央大学大学院 理工学研究科 情報工学専攻<sup>†‡</sup>

概要: 区分的微分可能なモデルに対する最適化は、微分不可能点を含んでおり困難である。本研究では Griewank らにより提唱されている Abs-Normal Form (ANF) を利用して一般化勾配を導出し、区分的線形化で関数を近似する際の誤差推定について考える。

キーワード: 区分的微分可能, 区分的線形近似, アルゴリズム微分

## 1 はじめに

工学問題で用いられる区分的微分可能なモデルに対するアルゴリズム微分 [1] は、微分不可能点の処理は実装に依存してきた。しかし近年、微分不可能点の扱いに進展があり、ANF を用いて微分不可能点での微分計算が可能になってきている [2, 3, 4, 5, 6]。

本研究では、区分的微分可能なモデルの中で絶対値を用いたアルゴリズム (手続き) により計算される関数に関して、その計算値に含まれる誤差推定の方法について検討する。さらに、二階微分による誤差推定の精度と計算効率の観点から最適化計算 [6, 7, 8] などにおける本手法の有用性を検証する。

## 2 アルゴリズム微分 [1]

アルゴリズム微分とは、ある関数の値を求めるアルゴリズムが与えられたとき、その偏導関数値を求めるアルゴリズムを導出する手法である。

### 2.1 基本演算

プログラム中で使用できる演算のことを基本演算といい、本研究では以下の演算を対象とする。

- 四則演算 ( +, -, ×, ÷ )
- 開平 ( sqrt )
- 初等超越関数 ( log, exp, sin, ... )
- 絶対値演算 ( abs )

最大値、最小値関数 ( max, min ) は abs により次のように定義する：

$$\begin{aligned} \max(u, w) &= (u + w + \text{abs}(u - w))/2, \\ \min(u, w) &= (u + w - \text{abs}(u - w))/2. \end{aligned}$$

### 2.2 中間変数と計算過程

基本演算の実行と一対一に対応し、その結果を格納する変数を中間変数という。中間変数を逐次計算する過程を計算過程という。

$y = f(x)$  ( $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ) を計算する手続きを次の計算過程で表現する (式 (1))：

$$\begin{aligned} u_{i-n} &= x_i, & i &= 1, \dots, n, \\ u_i &= \varphi_i(u_j)_{j < i}, & i &= 1, \dots, \ell, \\ y_i &= u_{\ell-m+i}, & i &= 1, \dots, m. \end{aligned} \quad (1)$$

$\varphi_i$  は基本演算,  $u_i$  は中間変数である。中間変数には、その計算順序に半順序関係がある。

### 2.3 偏導関数値の導出

アルゴリズム微分による偏導関数値は、各基本演算から合成関数の微分則である連鎖律 (chain rule) に従い導出できる (式 (2))：

$$\begin{aligned} \frac{\partial u_{i-n}}{\partial x_j} &= i == j ? 1 : 0, & i &= 1, \dots, n, \quad j = 1, \dots, n, \\ \frac{\partial u_i}{\partial x_j} &= \sum_{k < i} \varphi'_i(u_k) \frac{\partial u_k}{\partial x_j}, & i &= 1, \dots, \ell, \quad j = 1, \dots, n, \\ \frac{\partial y_i}{\partial x_j} &= \frac{\partial u_{\ell-m+i}}{\partial x_j}, & i &= 1, \dots, m, \quad j = 1, \dots, n. \end{aligned} \quad (2)$$

ここで  $\varphi'_i(u_k)$  は  $u_i$  を計算する演算における  $u_k$  に関する要素的偏導関数を表す。また、アルゴリズム微分を繰り返し実行することで二階以上の微分も計算することができる。

### 2.4 計算グラフ

演算と被演算子の関係を木構造で表現したグラフを計算グラフという。各ノードに中間変数が対応しており、本研究のアルゴリズム微分に使用するデータ構造は、この計算グラフを実装している。

## 3 Abs-Normal Form [4, 5, 6]

絶対値を含む式  $y = f(x)$  ( $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ ) に対して、絶対値演算を境にわけて表現し、このような計算過程の表現を ANF という (式 (3))：

$$\begin{aligned} w &= g(x, z), \\ z &= |w|, \\ y &= h(x, z), \end{aligned} \quad (3)$$

ただし、各変数は  $x \in \mathbb{R}^n$ ,  $y \in \mathbb{R}^m$ ,  $w \in \mathbb{R}^s$ ,  $z \in \mathbb{R}^s$  であり、 $g: \mathbb{R}^{n+s} \rightarrow \mathbb{R}^s$ ,  $h: \mathbb{R}^{n+s} \rightarrow \mathbb{R}^m$  は連続微分可能関数、 $z = |w|$  は成分毎の絶対値を表す。

### 3.1 基本演算の増分

各基本演算の増分の近似を以下のように考える (式 (4))：

$$\begin{aligned} \Delta u_i &= \Delta u_j \pm \Delta u_k & \text{when } u_i &= u_j \pm u_k, \\ \Delta u_i &= u_j \Delta u_k + u_k \Delta u_j & \text{when } u_i &= u_j \cdot u_k, \\ \Delta u_i &= \varphi'_i(u_j) \Delta u_j & \text{when } u_i &= \varphi_i(u_j), \\ \Delta u_i &= |u_j + \Delta u_j| - u_i & \text{when } u_i &= |u_j|, \end{aligned} \quad (4)$$

最終的に近似されている増分  $\Delta y_i$  を  $\Delta f(x; \Delta x)$  と記す。  $\varphi_i$  は連続微分可能な関数である。

このとき、関数  $f$  の点  $x$  での  $\Delta x$  方向の区分的線形近似  $f_{PL,x}(\Delta x)$  は以下のように表現できる (式 (5))：

$$f_{PL,x}(\Delta x) = f(x) + \Delta f(x; \Delta x). \quad (5)$$

### 3.2 ANF の区分的線形化

ANF (式 (3)) に基づいて、点  $x_0$  からの  $\Delta x$  方向の増分は、 $w_0 = g(x_0, z_0)$ ,  $z_0 = |w_0|$ ,  $y_0 = h(x_0, z_0)$  を用いて、 $g(x_0 + \Delta x, z_0 + \Delta z) - w_0$ ,  $|w_0 + \Delta w| - z_0$ ,  $h(x_0 +$

Error estimation of piecewise linear approximation

<sup>†</sup> Ryo MIYAGI, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

<sup>‡</sup> Koichi KUBOTA, Information and System Engineering Course, Graduate School of Science and Engineering, CHUO University

$\Delta x, z_0 + \Delta z) - y_0$  となる。これらから,  $w, z, y$  の増分の区分的線形近似を以下のように表現する (式 (6)) :

$$\begin{aligned} \Delta w &= \frac{\partial g}{\partial x}(x_0, z_0)\Delta x + \frac{\partial g}{\partial z}(x_0, z_0)\Delta z, \\ \Delta z &= |w_0 + \Delta w| - z_0, \\ \Delta y &= \frac{\partial h}{\partial x}(x_0, z_0)\Delta x + \frac{\partial h}{\partial z}(x_0, z_0)\Delta z. \end{aligned} \quad (6)$$

ここで,  $\widetilde{\Delta w} = \Delta w + w_0$ , および,  $\widetilde{\Delta z} = \Delta z + z_0 = |\widetilde{\Delta w}|$  を導入すると,  $\Delta y$  は以下のように表現できる (式 (7)) :

$$\begin{aligned} \widetilde{\Delta w} &= w_0 + \frac{\partial g}{\partial x}\Delta x + \frac{\partial g}{\partial z}(\widetilde{\Delta z} - z_0), \\ \widetilde{\Delta z} &= \Sigma \widetilde{\Delta w}, \\ \Delta y &= \frac{\partial h}{\partial x}\Delta x + \frac{\partial h}{\partial z}(\widetilde{\Delta z} - z_0). \end{aligned} \quad (7)$$

$\Sigma$  は各絶対値演算の引数の符号を格納する行列である。具体的な定義を以下に示す。符号関数 (式 (8)) を用いて,

$$\text{sign}(a) = \begin{cases} 1 & (a > 0) \\ 0 & (a = 0) \\ -1 & (a < 0) \end{cases} \quad (8)$$

絶対値演算は  $|w| = \text{sign}(w) \cdot w$  と書き換えられるので, 次のような対角行列を  $\Sigma$  として定義する :

$$\Sigma = \begin{pmatrix} \text{sign}(w_0 + \Delta w_0) & & \\ & \ddots & \\ & & \text{sign}(w_s + \Delta w_s) \end{pmatrix}.$$

式 (7) を定数項をまとめて  $w_c = w_0 - \frac{\partial g}{\partial z}z_0$ ,  $y_c = -\frac{\partial h}{\partial z}z_0$  とおいて, 行列で表現すると以下ようになる (式 (9)) :

$$\begin{aligned} \begin{pmatrix} \widetilde{\Delta w} \\ \Delta y \end{pmatrix} &= \begin{pmatrix} w_c \\ y_c \end{pmatrix} + \begin{bmatrix} \frac{\partial g}{\partial x} & \frac{\partial g}{\partial z} \\ \frac{\partial h}{\partial x} & \frac{\partial h}{\partial z} \end{bmatrix} \begin{bmatrix} \Delta x \\ \widetilde{\Delta z} \end{bmatrix}, \\ \widetilde{\Delta z} &= \Sigma \widetilde{\Delta w}. \end{aligned} \quad (9)$$

これを  $\Delta x$  について解くと,  $(I - \frac{\partial g}{\partial z}\Sigma)^{-1}$  を  $A$  とおいて,  $y = y_0$  における区分的線形近似  $y_0 + \Delta y$  を得る :

$$\begin{aligned} y_0 + \Delta y &= c + \left( \frac{\partial h}{\partial x} + \frac{\partial h}{\partial z}\Sigma A \frac{\partial g}{\partial x} \right) \Delta x, \\ c &= y_0 + y_c + \frac{\partial h}{\partial z}\Sigma A w_c. \end{aligned} \quad (10)$$

#### 4 ANF による関数の計算値の誤差推定 [1]

スカラー値関数  $y = f(x)$  の  $x = x_0$  における計算値  $y_0 = f(x_0)$  に含まれる計算誤差  $\delta_y$  の推定 (線形近似) を考える。もし  $w_k \neq 0$  ( $k = 1, \dots, s$ ) なら通常の誤差推定の方法を用いることができる。  $\exists k, w_k = 0$  なら, 誤差推定を  $w_k = 0$  のところで分ける必要がある。  $w_k = 0$  の場合にはその  $w_k$  の大域誤差推定値を算出し, その値を  $z_k = |w_k|$  の入力誤差と考えて  $\delta_{z_k}$  を計算して誤差推定を行う (式 (11)) :

$$\delta_y = \sum_{j=1}^n \frac{\partial h}{\partial x_j} \delta_{x_j} + \sum_{k=1}^s \frac{\partial h}{\partial z_k} \delta_{z_k} + \sum_i \frac{\partial h}{\partial u_i} \delta_{u_i}. \quad (11)$$

#### 5 近似値の誤差推定

第3節で述べた手法をもとに, 区分的微分可能な関数に対する微分計算と線形近似を行う。この線形近似から絶対値演算の引数の符号が切り替わる点 (kink point) を探索する。kink point の探索は,  $\Delta x$  の値を更新していき, 式 (7) の  $\Sigma$  の対角成分のいずれかの値が 0 以外になるような  $\Delta x$  を見つければよい。本研究では, 線形近似された kink point の点の誤差を推定することを試みる。

ANF により目的関数を線形近似すると, 基準点のとり方次第では  $w_i$  と  $\Delta w_i$  の符号が同じになり, 実在する kink point が見つからないことがある。このような状況に対して, 基準点を勾配方向の逆方向に更新することで対応した。

#### 6 数値実験

C++ の演算子多重定義を用いた処理系を試作し, 文献 [7] のテスト問題の中から 2 変数の関数 11 種類と想定される近似値の総数が多くなる例として, 絶対値演算が入り子になった関数を対象に区分的線形化の数値実験を行った。具体的には, kink point に対して ANF による区分的線形化を計算し,  $\Sigma$  の値を意図的に変更し kink point 周辺での勾配の変化の推測による近似値を記録する。この kink point 周辺の推測と実測値の誤差を二階微分による二次近似で修正可能か検証する。ここでは,

$$\begin{aligned} f_1(x_1, x_2) &= -x_1 + 2(x_1^2 + x_2^2 - 1) + 1.75|x_1^2 + x_2^2 - 1|, \\ f_2(x) &= ||x^2 - 1| - 2|, \end{aligned}$$

に対して区分的線形近似を実行した結果を以下に示す (図 1, 2)。赤い線はそれぞれの基準点の位置 ( $f_1$  は  $(x_1, x_2) = (-1, -1)$ ,  $f_2$  は  $x = -2.8$ ) を示す。

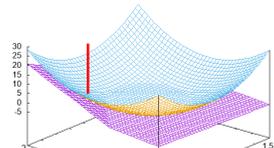


図1  $f_1, f_{1PL,(-1,-1)}$

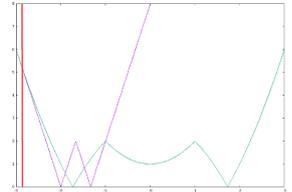


図2  $f_2, f_{2PL,(-2.8)}$

#### 参考文献

- [1] 久保田 光一, 伊理 正夫, “アルゴリズムの自動微分と応用”, コロナ社, 1998.
- [2] 吉田 北斗, “区分的微分可能関数と一般化自動微分”, 中央大学理工学研究科情報工学専攻, 2017.
- [3] 吉田 北斗, 久保田 光一, “区分的微分可能関数と一般化自動微分”, 情報処理学会 第 79 回全国大会予稿集, vol. I, p. 243, 2017.
- [4] A. Griewank, “On stable piecewise linearization and generalized algorithmic differentiation,” Optimization Methods and Software, vol. 28, pp. 1139–1178, Dec. 2013.
- [5] A. Griewank, Jens-Uwe Bernt, Manuel Radons and Tom Streubel, “Solving piecewise linear equations in absolute form”, Linear Algebra and its Application, vol. 471, pp. 500–530, 2015.
- [6] A. Griewank, A. Walther, S. Fiege, and T. Bosse, “On Lipschitz optimization based on gray-box piecewise linearization,” Mathematical Programming, vol. 158, no. 1, pp.383–415, 2016.
- [7] A. Bagirov, N. Karmitza, M. Mäkelä, “Introduction to Nonsmooth Optimization,” Springer, 2014.
- [8] 関口 良行, “はじめての最適化”, 近代科学社, 2014.