

最適化算法と高速自動微分法について

慶応義塾大学理工学部 久保田 光一 (Koichi KUBOTA)
東京大学工学部 伊理 正夫 (Masao IRI)
東京大学工学部 室田 一雄 (Kazuo MUROTA)

1. はじめに

最適化問題の「目的関数」が、何らかの形で積分を含むことは少なくない。その積分が解析的に計算できるような幸運な場合は別として、一般には、何らかの数值積分手法を利用してその目的関数の“近似値”を計算することができるだけである。このように表式は得ることができてもその正確な値を計算することが困難であるような関数を目的関数に持つ最適化問題においては、その近似値を“十分に”小さくするようなパラメータの組を何らかの方法で見出し、それを「最適解」として採用せざるをえない。

ここでは、従来の解法と「高速自動微分法」に基づく解法とを比較し、それらの問題点について考察し、最適化問題の例として、「地理的最適化問題」を取り上げ、数值実験を行なう。

2. 最適化算法と微分

現在までに様々な最適化手法が開発され¹、プログラムパッケージの形で利用できるものも多くなっている。それらの最適化パッケージを利用する際には、通常、目的関数 $F(\mathbf{x})$ とその勾配 $\nabla F(\mathbf{x})$ (ときには Hesse 行列 $\nabla^2 F$ も) の計算手順をサブルーチンなどの形で記述して入力として与えることが必要とされる。しかし、利用者の立場からすると、目的関数 $F(\mathbf{x})$ の定義式だけでなく、その勾配 $\nabla F(\mathbf{x})$ の計算法まで逐一記述して入力するのは煩わしい。数学的には、 $F(\mathbf{x})$ が定義されれば $\nabla F(\mathbf{x})$ も一意的に定まるのであるから、 $(F(\mathbf{x}), \nabla F(\mathbf{x}))$ の組を入力データとして与えるのは何とも冗長である。

最適化の諸算法は、目的関数 $F(\mathbf{x})$ とその勾配 $\nabla F(\mathbf{x})$ とが正しく与えられていることを当然の前提として設計されているわけであるが、現状のように $F(\mathbf{x})$ と $\nabla F(\mathbf{x})$ とを独立に入力する方式だと、何らかの理由 (例えば単純な計算まちがい) で目的関数と勾配が正しく対応しないことも起こりうる。そのような不整合入力に対して最適化パッケージはどのように対処するのであろうか。冗長なデータを入力として与えてその整合性の心配をするよりは、目的関数 $F(\mathbf{x})$ の計算式だけを入力データとして与えて、その勾配などは最適化パッケージの内部で高速自動微分法 (§3 参照) によって自動生成する方が今や正統的な方法であろう。

さて、地理的最適化問題 (§4) など、積分の形で定義されてる目的関数 $F(\mathbf{x})$ を最小(大)化しようとする最適化問題について、従来の実際的な最適化手順を整理してみると、次のようになっている。

- (1) $F(\mathbf{x})$ の勾配 $\nabla F(\mathbf{x})$ などの表式を解析的に導く。これもまた積分の形で表される。
- (2) $F(\mathbf{x})$ の積分表式を適当な数值積分公式を用いて離散近似する。このようにして得られた近似目的関数を $\bar{F}(\mathbf{x})$ と書き表そう。
- (3) $\nabla F(\mathbf{x})$ の積分表式を適当な数值積分公式を用いて離散近似する。このようにして得られた近似勾配を $\bar{\nabla} F(\mathbf{x})$ と書き表そう。
- (4) $\bar{F}(\mathbf{x})$ と $\bar{\nabla} F(\mathbf{x})$ の計算手順を別々にサブルーチンの形に記述して、既存の最適化手法を適用する。

上の手順の中で、(1) の計算は相当に面倒であり、正しく計算できたという確信がなかなか得られないことも稀ではない。しかし、ここではこの計算は正しく行われているとしよう。

ここで議論したいのは、目的関数と勾配のそれぞれの表式を独立に離散近似して、その近似式

の組 $(\bar{F}(x), \nabla\bar{F}(x))$ を最適化パッケージの入力として与えるという計算法の妥当性である。すぐわかるように、真の勾配 $\nabla F(x)$ の離散近似関数 $\nabla\bar{F}(x)$ は、 $F(x)$ の離散近似関数 $\bar{F}(x)$ の勾配 $\nabla\bar{F}(x)$ とは異なるのが普通である。すなわち離散近似と微分演算は可換でない：

$$\nabla\bar{F}(x) \neq \nabla\bar{F}(x).$$

すると、上記の手順は不整合入力データ $(\bar{F}(x), \nabla\bar{F}(x))$ に対して最適化手法を適用していることになる。もちろん、

$$F(x) \approx \bar{F}(x), \quad \nabla F(x) \approx \nabla\bar{F}(x)$$

であるから、大雑把に言えば

$$\nabla\bar{F}(x) \approx \nabla\bar{F}(x)$$

が成立しているに違いない。しかし、目的関数 $F(x)$ の離散近似 $\bar{F}(x)$ には無限の可能性があり、勾配 $\nabla F(x)$ の離散近似 $\nabla\bar{F}(x)$ についても同様であるから、上記の計算手順が両者の離散近似の整合性という面倒な問題を抱えていることは確かである。

従来実施されてきた地理的最適化問題の諸解法ではこの問題に目をつぶってきたが、目的関数と勾配を独立に離散化して最適化パッケージに入力するという手順にはやはり無理があろう。それよりも、 $F(x)$ の離散近似 $\bar{F}(x)$ だけを最適化パッケージに入力として与えて、その勾配 $\nabla\bar{F}(x)$ は高速自動微分法 (§3) で自動生成するという方式の方があらゆる点で健全である。そうすれば、積分表示式の微分を解析的に求める必要もなくなるし、関数と勾配の整合性に気を使う必要もなくなる。

3. 高速自動微分法

多くの変数 $x \in \mathbf{R}^N$ の実数値関数 $\bar{F}(x)$ を計算する手続きが（通常のプログラム言語を用いて）与えられているとき、 $\bar{F}(x)$ の x に関する勾配ベクトル $\nabla\bar{F}(x)$ ($\nabla\bar{F}(x)$ ではないことに注意) や Hesse 行列 $\nabla^2\bar{F}(x)$ などを計算する手続きを自動的かつ高速に生成する実用的な算法“高速自動微分法”が最近開発されている^{2,6,8,9,11,12}。

次の第4節の地理的最適化問題に見られるような複雑な関数（近似目的関数） $\bar{F}(x)$ は、その計算手順を正しく記述するだけでも一苦勞であり、ましてやその微分を計算する式を手計算で誤りなく与えるのは大変な作業である。しかし、高速自動微分法を利用すれば、人間は関数 $\bar{F}(x)$ の定義を記述するだけでよく、勾配などの計算手順は自動的に生成されてしまう。しかも、関数 $\bar{F}(x)$ を定義するプログラム中には条件分岐や反復計算（if 文、while 文、do 文、for 文）などが含まれていても構わないので、応用上十分一般的な関数を扱うことができる。

高速自動微分法はその名の示す通り高速であり、実用的な規模の複雑な関数 $\bar{F}(x)$ を扱うことができる。例えば、 N 変数関数の勾配ベクトル $\nabla\bar{F}(x)$ の計算手続きを生成するための手間も、その手続きを実行して $\nabla\bar{F}(x)$ の値を得るための手間も、ともに、関数 $\bar{F}(x)$ そのものを計算するための手間の高々定数倍（その定数の値は変数の数 N にはもちろん関数の形にも依らない！）で済んでしまう。

通常のように、差分に基づく数値微分法によって、例えば、

$$\frac{\partial\bar{F}}{\partial x_i} \approx (\bar{F}(x + h_i e_i) - \bar{F}(x)) / h_i, \quad i = 1, \dots, N,$$

（ただし、 e_i は第 i 単位ベクトル）のように計算したとすると、 $\nabla\bar{F}(x)$ を計算する手間は $\bar{F}(x)$ の計算の手間の $N+1$ 倍程度になってしまう。しかも、このようにして得られる $\nabla\bar{F}(x)$ の値は微分を差分で代用したことによって生じる離散化誤差を含む近似値であり、刻み幅 h_i を適切に選ばないと桁落ちによって丸め誤差が拡大して精度が著しく低下するという別の問題点

ももっている。これに比して、高速自動微分法によれば、より速くより精度よく $\nabla \bar{F}(x)$ を計算できるのである。

高速自動微分法の算法は、関数計算の過程が（加減乗除算、指数・対数・三角関数などの）基本的な演算の列であること、これらの基本演算の微分が直ちに計算できること、そして合成関数の微分がいわゆる鎖律 (chain rule) によって逐次的に計算できることの三つの事実に基づいた自然な算法である。いわゆる計算グラフの概念を用いるとその算法の原理を直感的に理解しやすい。

高速に自動的に偏導関数を計算できるということの他に、もうひとつ大きな利点がある。それは、副産物として、計算値に含まれる計算誤差の大きさを評価できることである。その評価値は、数値計算の品質保証という面から重要であるばかりでなく、反復解法の妥当な反復停止規則を与えるという意味も持っている。さらに、誤差評価値はベクトルのノルムをどのように計算すべきかという問題で、「正規化ノルム」^{9,11} という形で本質的な役割を果たす。たとえば、最適化算法では、勾配 (ベクトル) が 0 ベクトルに等しいかどうかを判定する必要があることが多いが、このとき、勾配ベクトルの大きさを測るのに、正規化ノルムを利用できる。すると、近似解の精度を定める上で重要な量でありながら従来は「適当に」設定されていた「十分に小さな数」に代わって、妥当な値を設定できるのである^{9,11}。

4. 地理的最適化問題

地理的最適化問題という分類中にも色々な種類の問題があるが⁷、それらの目的関数の殆んどが積分の形で定義されている。ここでは、その中で、次に述べる仮定のもとで、ある地域に n 個の施設 (例えば図書館) P_1, \dots, P_n を利用者の費用が最小になるように「最適に」配置するという問題を取り上げる (図書館問題とも呼ばれる)。

仮定 1. 施設の個数 n は固定する;

2. 施設の利用者は、自分の住居から一番近い施設だけを利用する;

3. 施設を利用するのに必要な費用は、施設と利用者の住居との 2 乗距離に比例する ([10] では、2 乗距離の非負狭義単調増加関数という形で一般化されている);

4. 利用者の (人口) 分布は $\phi(x)$ で与えられる;

5. 施設配置に関する費用 — 目的関数 — は、各利用者に関する費用の総和 (積分) である。

施設 P_i の座標を $x_i = (x_i^1, x_i^2)$ ($i = 1, \dots, n$) と記せば、以上の仮定のもとで、目的関数は次のように定式化される:

$$F(x_1, \dots, x_n) = \frac{1}{2} \int \min_i \{ \|x - x_i\|^2 \} \phi(x) dx. \quad (1)$$

ここで、施設の位置 $X = (x_1, \dots, x_n)$ を定め、それらを母点として Voronoi 図 ([7] 参照) を構成すると、式 (1) の \min の代わりに施設 P_i の Voronoi 領域 V_i に関する定積分を用いて、式 (1) は

$$F(x_1, \dots, x_n) = \frac{1}{2} \sum_{i=1}^n \int_{V_i} \|x - x_i\|^2 \phi(x) dx \quad (2)$$

と表すことができる¹⁰。この目的関数の変数の個数を N とすると、 n 個の施設の各位置は 2 個の座標により表されるので、 $N = 2n$ である。この N 変数関数 $F(X)$ を最小にすることがここでの最適化問題を解くことになる。なお、 F の計算のためには、実際に Voronoi 図を構成する必要があるが、杉原・伊理により開発された Voronoi 図高速構成プログラムを利用することにより、大規模な問題 (施設の数が 100 万のものなど) の Voronoi 図も実用的な手間で構成できるようになっている¹⁴。

式(2)の1階偏導関数 ∇F および2階の偏導関数 $\nabla^2 F$ は、より一般的な形で、既に導出されている¹⁰。以下では上の仮定に基づく結果だけを示そう。記号は[10]に倣い、施設 P_i の座標の第 λ 成分を x_i^λ ($\lambda=1,2$) と記す。また、計量テンソル $g_{\lambda\kappa}$ を明示し、Einstein の総和記法を用いる。

まず、目的関数の x_i^λ に関する偏導関数(勾配の成分)は、

$$\frac{\partial F}{\partial x_i^\lambda} = \mu(V_i)g_{\lambda\kappa}(x_i^\kappa - \bar{x}_i^\kappa) \quad (3)$$

により与えられる。ここで、

$$\begin{aligned} \mu(V_i) &= \int_{V_i} \phi(\mathbf{x})d\mathbf{x} \\ \bar{x}_i^\kappa &= \int_{V_i} x^\kappa \phi(\mathbf{x})d\mathbf{x} / \mu(V_i) \end{aligned} \quad (4)$$

である。2階偏導関数は、(i)対角成分、(ii)Voronoi領域が隣接している施設 P_i, P_j に対応する (i,j) 成分、(iii)その他の成分に分けて記述される (W_{ij} は隣接する施設 P_i と P_j のVoronoi辺):

$$\frac{\partial^2 F}{\partial x_i^\kappa \partial x_j^\lambda} = \begin{cases} H_{\lambda\kappa}^i + G_{\lambda\kappa}^i & \text{for } j = i, \\ G_{\lambda\kappa}^{ji} & \text{for } j \neq i \text{ with } W_{ij} \neq \emptyset, \\ 0 & \text{for } j \neq i \text{ with } W_{ij} = \emptyset. \end{cases} \quad (5)$$

ここで、施設 P_i と P_j の距離を α_{ij} ($= \|\mathbf{x}_i - \mathbf{x}_j\|$) として、

$$\begin{aligned} H_{\lambda\kappa}^i &= \mu(V_i) \cdot g_{\lambda\kappa}, \\ G_{\lambda\kappa}^i &= - \sum_{j:j \neq i, W_{ij} \neq \emptyset} K_{\lambda\kappa}^{iji}, \\ G_{\lambda\kappa}^{ji} &= K_{\lambda\kappa}^{jji}, \\ K_{\lambda\kappa}^{kji} &= \frac{\mu(W_{ij})}{\alpha_{ij}} g_{\lambda\nu} g_{\kappa\tau} [x_i^\tau \cdot x_k^\nu - x_i^\tau \cdot \bar{x}_{ij}^\nu - \bar{x}_{ij}^\tau \cdot x_k^\nu + \bar{x}_{ij}^\tau \bar{x}_{ij}^{\nu\tau}]. \end{aligned} \quad (6)$$

式(6)の計算には、Voronoi領域 V_i に関する積分(2次元積分)の他に、次のVoronoi辺 W_{ij} に関する積分(線積分)も計算しなければならない:

$$\begin{aligned} \mu(W_{ij}) &= \int_{W_{ij}} \phi(\mathbf{x})ds, \\ \bar{x}_{ij}^\tau &= \int_{W_{ij}} x^\tau \phi(\mathbf{x})ds / \mu(W_{ij}), \\ \bar{x}_{ij}^{\kappa\lambda} &= \int_{W_{ij}} x^\kappa x^\lambda \phi(\mathbf{x})ds / \mu(W_{ij}). \end{aligned} \quad (7)$$

5. 数値実験

ここでは、地理的最適化問題(式(2)で定義される F の最小化問題)について、最適化算法としてニュートン法(ニュートン・ラフソン Newton-Raphson 法)を選び、従来の不整合入力データを与えた時の近似解と高速自動微分法による整合データを与えた時の近似解とを比較する。

5.1. 離散化

上述した地理的最適化問題の目的関数とその偏導関数の値を計算するには、数値積分を実行しなければならない。Voronoi 領域に関する2次元積分(以降、面積分と呼ぶ)には、まず Voronoi 領域を三角形に分割して、各三角形領域で標本点が7個のガウス型積分公式(公式[H]と記す)³を用い、1次元積分(以降、線積分と呼ぶ)には、台形則と Gauss 積分とを用いることにする。

まず、目的関数 F とその勾配 ∇F には面積分しか現れないので、それらを公式[H]で離散近似し、それぞれ、 \overline{F} , $\overline{\nabla F}$ と記す。Hesse 行列 $\nabla^2 F$ には、面積分の他に、線積分も現れる(式(7))。そこで、面積分には公式[H]、線積分に台形則を用いて離散化したものを $\overline{\nabla^2 F}_T$ 、また、線積分にガウス積分を用いて離散化したものを $\overline{\nabla^2 F}_G$ と記す。

$(\overline{F}, \overline{\nabla F}, \overline{\nabla^2 F}_T)$ も $(\overline{F}, \overline{\nabla F}, \overline{\nabla^2 F}_G)$ も §2 で述べた不整合入力データである。なぜなら、 F , ∇F , $\nabla^2 F$ の離散近似の方法が互いに無関係だからである。特に、 $\overline{\nabla^2 F}_T$, $\overline{\nabla^2 F}_G$ では、面積分公式と、線積分公式とが無関係に使用されていることに注意されたい。

一方、高速自動微分法を用いると、 \overline{F} だけから整合入力データ $D_{FAD} \equiv (\overline{F}, \overline{\nabla F}, \nabla^2 \overline{F})$ が得られる。

なお、標本点7の台形則を用いた $\overline{\nabla^2 F}_T$ を $\overline{\nabla^2 F}_{T7}$ と記し、標本点が2, 4のガウス積分を用いた $\overline{\nabla^2 F}_G$ を、それぞれ、 $\overline{\nabla^2 F}_{G2}$, $\overline{\nabla^2 F}_{G4}$ と記すことにする。さらに、3種の不整合入力データを $D_{T7} \equiv (\overline{F}, \overline{\nabla F}, \overline{\nabla^2 F}_{T7})$, $D_{G2} \equiv (\overline{F}, \overline{\nabla F}, \overline{\nabla^2 F}_{G2})$, $D_{G4} \equiv (\overline{F}, \overline{\nabla F}, \overline{\nabla^2 F}_{G4})$ と記す。

5.2. 実験方法

施設は、その数を $n = 30$ に固定し、 $[-0.15, 0.15] \times [-0.15, 0.15]$ の正方形領域に配置することとする。すなわち、目的関数の変数は60個である。また、その領域内の人口分布は $\phi(x) = e^{-100\|x\|^2}$ であるとする。これは領域の中心部の密度が高いような人口分布のモデルである。Voronoi 図を構成するためには Voronoi2¹⁴、高速自動微分法のためには PADRE2¹³ というプログラムを使用する。実験には、サン・マイクロシステムズ社の SUN4-65 を用い、すべて倍精度で計算する。

局所最適解(以降、極小解)付近での最適化算法の振舞いを観察するために、予め(近似)極小解を計算しておき、それに十分近い点を初期値にとる。初期値では $\nabla^2 \overline{F}$, $\overline{\nabla^2 F}_T$, $\overline{\nabla^2 F}_G$ とともに正定値となり、初期値からニュートン法が遂行されるようにしておく。

停止条件の設定は重要な課題である。整合入力データ D_{FAD} に関しては、副産物として $\nabla \overline{F}$ の正規化ノルム $\|\nabla \overline{F}\|_N$ も計算されるので、停止条件として $\|\nabla \overline{F}\|_N \leq 1$ を採用できる。しかし、不整合入力データに関しては停止条件の設定方法の決め手が無い。ここでは特に停止条件を設定せずに最適化算法の振舞いを観察する。ただし、この実験では、不整合入力データについても $\nabla \overline{F}$ の計算誤差推定のためのプログラムを特別に作成し、 $\nabla \overline{F}$ の正規化ノルムも観察する。

また、最適化算法の大域的な振舞いについては §5.5 で簡単に触れる。

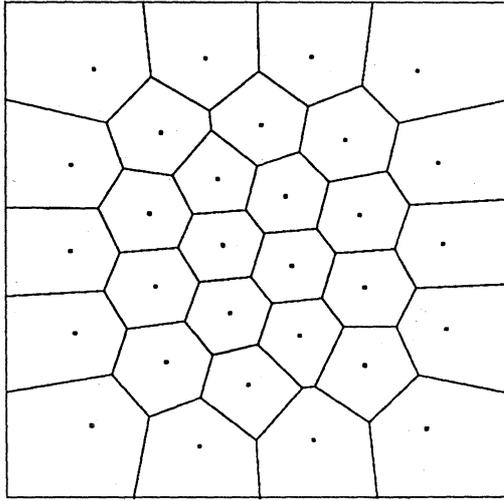


図1. 初期配置による Voronoi 図.

30 施設. 人口分布: $e^{-100\|x\|^2}$. 領域: $[-0.15, 0.15] \times [-0.15, 0.15]$.

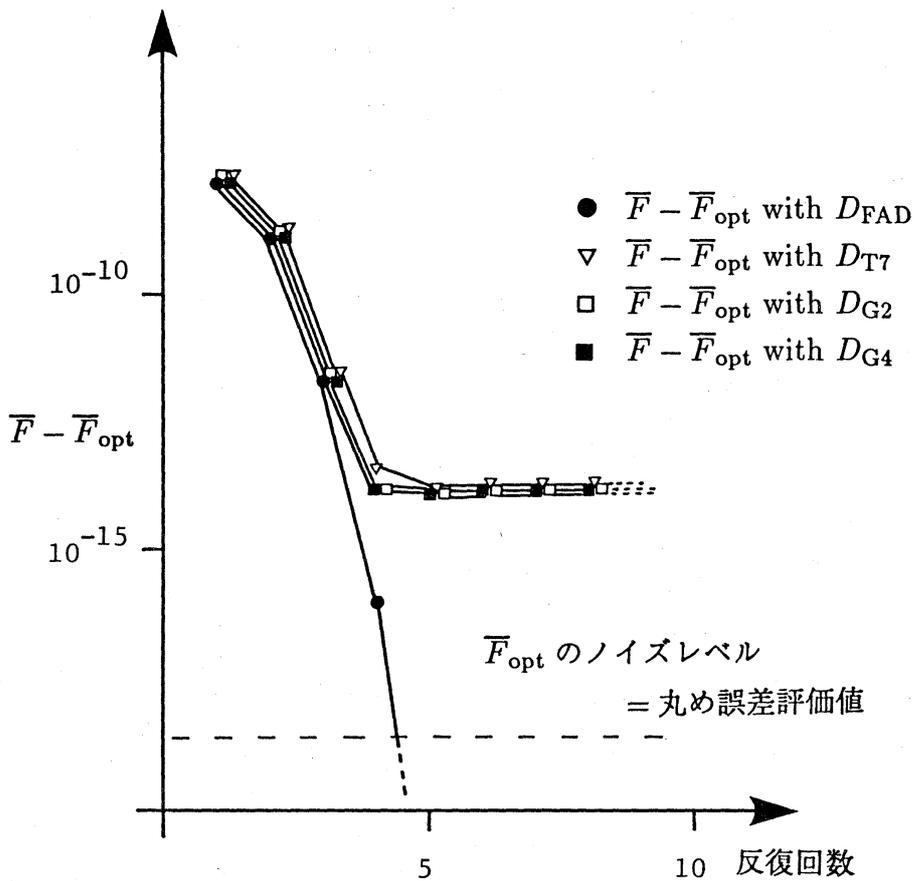


図2. 近似関数 \bar{F} の値と D_{FAD} により得られた近似極小値 \bar{F}_{opt} との差.

点線は \bar{F}_{opt} のノイズレベルを表す. 勾配を 0 ベクトルにすること, \bar{F} を小さくすることとが必ずしも一致しないことの実例. 不整合データに基づく解法では, 近似勾配と近似関数とが無関係なために, 近似関数値が下がりきらない.

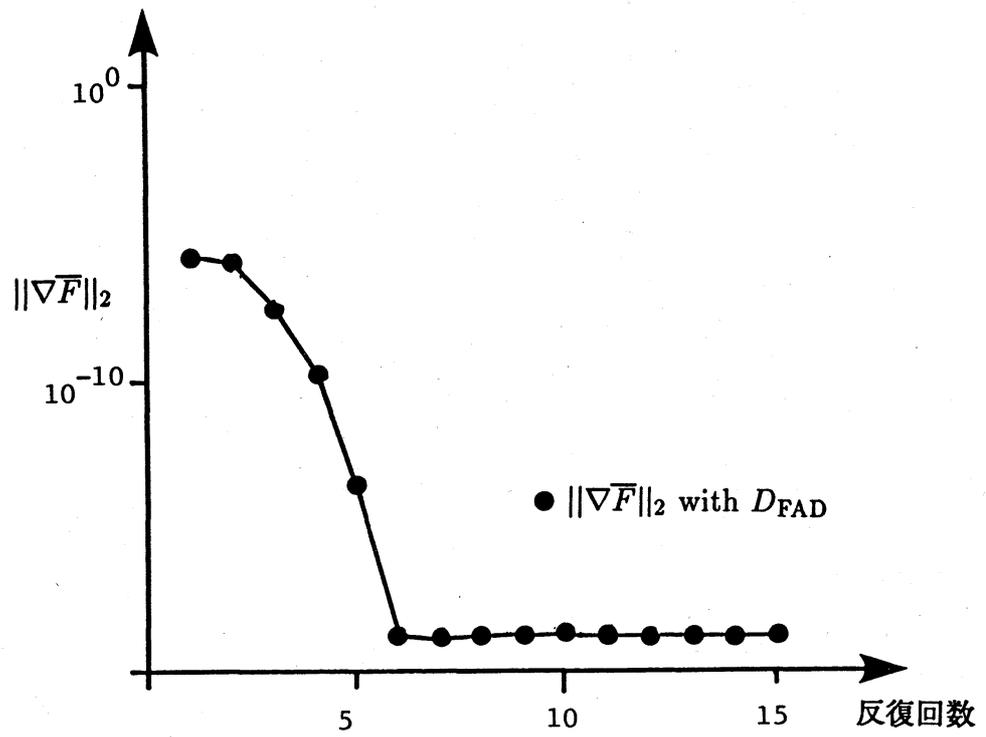


図 3.(a) 勾配 $\nabla \bar{F}$ の 2 ノルム.

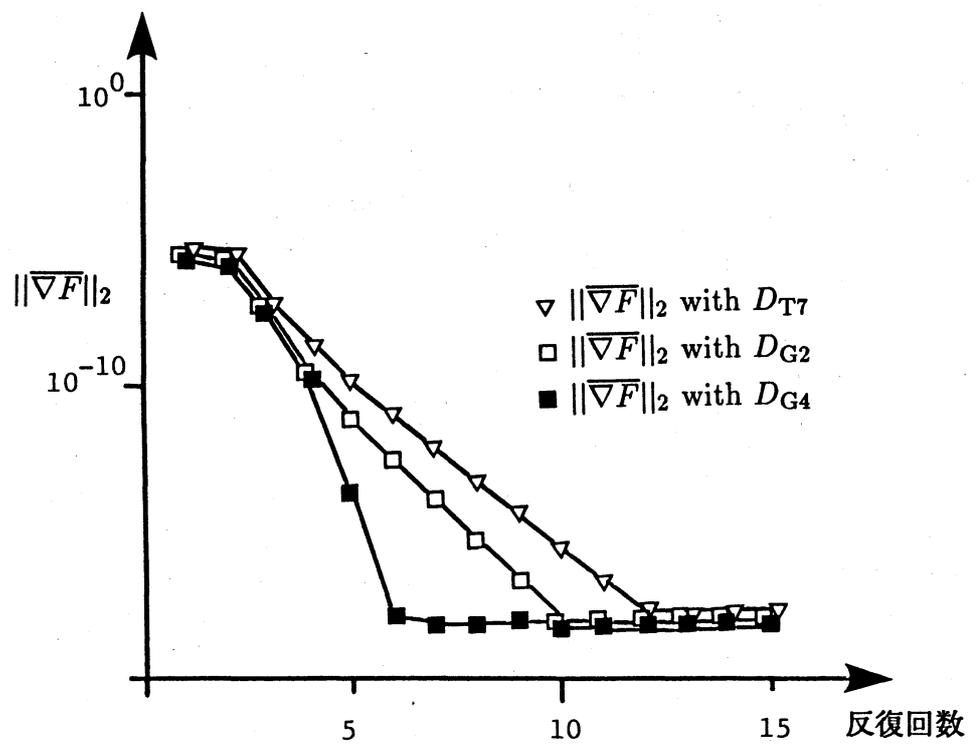


図 3.(b) 勾配 $\nabla \bar{F}$ の 2 ノルム.

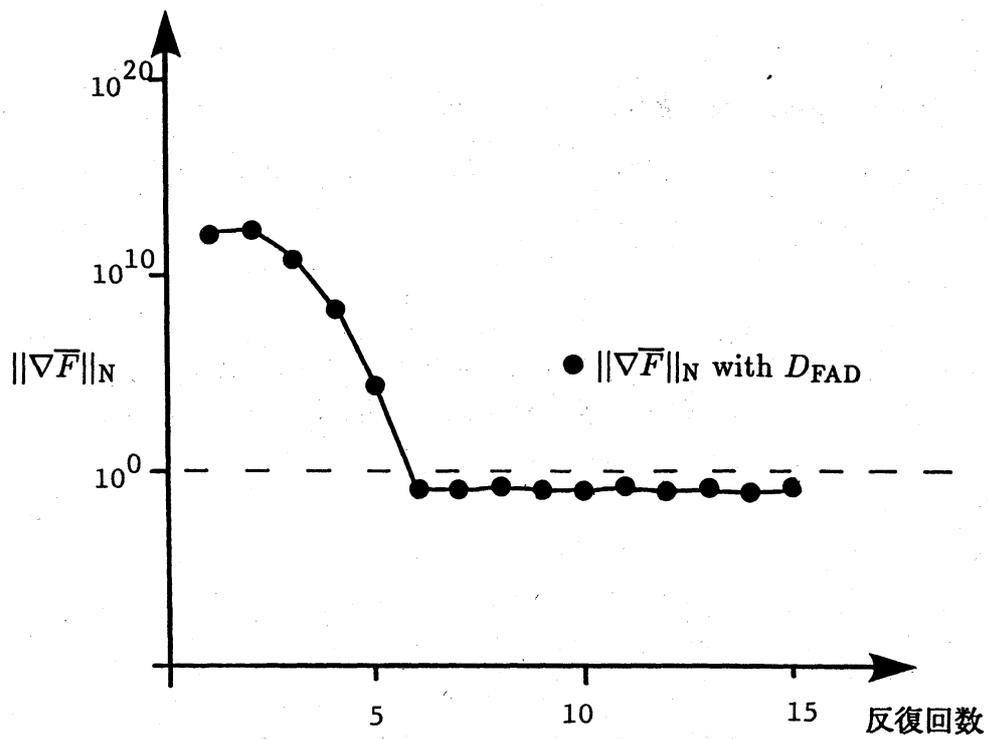


図 4.(a) 勾配 $\nabla \bar{F}$ の正規化ノルム.

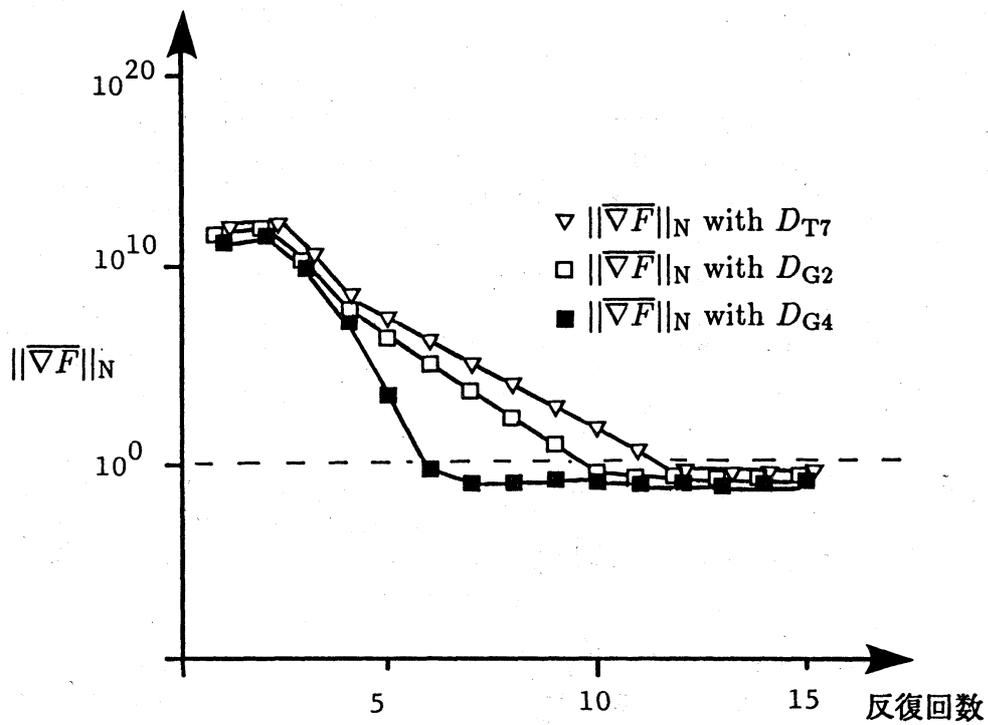


図 4.(b) 勾配 $\nabla \bar{F}$ の正規化ノルム.

5.3. 実験結果

施設の初期配置による Voronoi 図を図 1 に示す. まず D_{FAD} から, \bar{F} の最適解 \bar{F}_{opt} (の一つの上界) を計算する (ニュートン法の反復 5 回目以降 \bar{F} の値は一定になったのでそれを \bar{F}_{opt} とした). §5.1 の 4 種のデータから, ニュートン法の反復により生成される点列の各点での, 関数値 \bar{F} と \bar{F}_{opt} との差を図 2 に示す. 反復回数は 8 の所まで図示したが, それ以降は関数値の変化はみられなかった. また, その点列の各点での勾配の 2 ノルム $\|\nabla\bar{F}\|_2$, $\|\overline{\nabla\bar{F}}\|_2$ を図 3 に示す. さらに, $\nabla\bar{F}$ の正規化ノルム (これは $\nabla^2\bar{F}$ の計算の副産物として得られる) と, $\overline{\nabla\bar{F}}$ の正規化ノルム (これは別途作成したものから得られる) とを図 4 に示す.

5.4. 考察

図 2,3(a),4(a) から, 整合入力データ D_{FAD} により生成される点列が 2 次収束していることが読みとれる. これは, 単にニュートン法を実行しているのであるから, 当然である. 一方, 従来の不整合データ D_{T7} , D_{G2} を入力した場合のニュートン法では, ニュートン法であるにも拘らず, 1 次収束であることが読みとれる (図 3(b),4(b)). それどころか, 最終的に近似最適値として得られる関数値は, D_{FAD} により得られる関数値 \bar{F}_{opt} よりも「有意」に大きくなっている (図 2). 「有意」というのは, \bar{F}_{opt} の丸め誤差評価値 (= ノイズレベル) を考慮すると, まだ \bar{F} の値は小さくなりうるのに, 小さくならずにあるところで留まっているということである. これは, \bar{F} , $\nabla\bar{F}$, $\nabla^2\bar{F}$ の間の不整合のためである. 線積分の精度を高めた D_{G4} という入力データにより生成される点列の振舞いからは, 2 次収束の様子が見える. しかし, 収束先は D_{T7} , D_{G2} のものと同じであり, $\nabla\bar{F}$ を 0 にするという形で計算される最適解は必ずしも \bar{F} を極小にするわけではないことの例になっている. そのうえ, 一般には, 線積分の精度をどの程度に設定すべきかという基準も無い. それに比べて, D_{FAD} をデータとして与えたニュートン法は, $\nabla\bar{F}$, $\nabla^2\bar{F}$ の計算が自動的であるばかりでなく, 近似方法の選定法の問題もなく, ニュートン法により 2 次収束点列を生成することができるのである.

また, 正規化ノルムについては, $\|\nabla\bar{F}\|_N \leq 1$, $\|\overline{\nabla\bar{F}}\|_N \leq 1$ という停止条件が, $\nabla\bar{F}$, $\overline{\nabla\bar{F}}$ の収束の瞬間 — 2 次収束過程が終了しベクトルの大きさが減少しなくなるころ — を確実に捉えていることがわかる (ただし, 最適化問題解法として意味があるのは前者だけである).

5.5. その他

初期値をランダムに選んだ場合, すなわち, 初期値が極小解から離れている場合の収束の様子は [10], [15] で調べられている. ただし, 極小解から離れると, Hesse 行列が正定値にならないので, (1) 最急降下法, (2) 対角行列 (式 (5) で $G_{\lambda\kappa}^i$ と $G_{\lambda\kappa}^{ji}$ とを零とおいたもの) によりヘッセ行列を近似する (修正) ニュートン法, (3) 共役勾配法などを用いる必要がある. 最急降下法は反復回数が多過ぎて非実用的であるが, (2) は速く収束するという事¹⁰, また, (3) も少ない反復で (近似) 極小解を得られるということ¹⁵ などが報告されている. 高速自動微分法を用いると, 共役勾配法の Hestenes-Stiefel-Daniel の公式が簡単に使用できるだけでなく, 直線探索をニュートン法で実行し, その停止条件にも丸め誤差評価値を利用できる.

また, 二つの点における関数の計算値の差が丸め誤差評価値よりも大きければ, その二つの点での関数値の大小関係を確実に判定できる. したがって, Goldstein の規則などの, 粗い直線探索についても, 規則中に丸め誤差評価値を取入れることにより, 関数値が確実に減少することを保証したステップ幅などを計算したり, ステップ幅が異常に小さくなって一歩も進めないという状況を打開したりすることもできる.

6. おわりに

本報告では, 最適化算法に高速自動微分法がどう係わっているのかについて述べ, 実験に

よりそれを確認した。モデルの離散近似が必須な場合だけでなく、目的関数 F が複雑に込み入ってくると、偏導関数の計算に高速自動微分法は不可欠となるであろう。今後は、誤差評価値を積極的に利用して、計算値の品質を実用的な手間で保証していくなどの重要な課題に取り組む必要があるであろう(最適化算法に関する区間演算を利用した最適解の存在範囲の保証は、E.R.Hansen により発表^{4,5}されているが、未だ実用的なものとは言い難い)。

参考文献

- [1] P.E.Gill, W.Murray, and M.H.Wright : *Practical Optimization*. Academic Press, Inc., London, 1981.
- [2] A. Griewank: On Automatic Differentiation. *Mathematical Programming — Recent Developments and Applications*, M. Iri and K. Tanabe eds., Kluwer Academic Publishers, 1989, pp.83–107.
- [3] P. C. Hammer, O. J. Marlowe, and A. H. Stroud: Numerical Integration over Simplexes and Cones. *Mathematical Tables and other Aids to Computation*, Vol. 10 (1956), pp.130–137.
- [4] E. R. Hansen: Global Optimization Using Interval Analysis: The One-Dimensional Case. *Journal of Optimization Theory and Applications*, Vol. 29 (1979), pp. 331–344.
- [5] E. R. Hansen: Global Optimization Using Interval Analysis: The Multi-Dimensional Case. *Numerische Mathematik*, Vol. 34 (1980), pp. 247–270.
- [6] M. Iri: Simultaneous Computation of Functions, Partial Derivatives and Estimates of Rounding Errors — Complexity and Practicality. *Japan Journal of Applied Mathematics*, Vol.1, No.2 (1984), pp.223–252.
- [7] 伊理正夫監修, 腰塚武志編集: 計算幾何学と地理情報処理. 共立出版, 1986.
- [8] 伊理正夫, 久保田光一: 高速自動微分法 — グラフ, 丸め誤差, ノルム. 数理科学, No.285 (1987), pp.41–48.
- [9] 伊理正夫, 久保田光一: 数値計算の基礎技術の最近の話題から — ノルム, 丸め誤差, 偏微分と高速自動微分法を中心として. 電子情報通信学会誌, Vol. 72, No. 10 (1989), pp. 1044–1052.
- [10] M.Iri, K.Murota, and T.Ohya: A Fast Voronoi Diagram Algorithm with Applications to Geographical Optimization Problems. *Proceedings of the 11th IFIP Conference on System Modelling and Optimization*, Copenhagen, 1983, Lecture Notes in Control and Information Science 59, “System Modelling and Optimization” (P.Thoft-Christensen, ed.), Springer-Verlag, Berlin, 1984, pp.273–288.
- [11] M. Iri, T. Tsuchiya, and M. Hoshi: Automatic Computation of Partial Derivatives and Rounding Error Estimates with Applications to Large-scale Systems of Nonlinear Equations. *Journal of Computational and Applied Mathematics*, Vol.24 (1988), pp.365–392.
- [12] 久保田光一: 高速自動微分法と応用. 東京大学大学院工学系研究科博士論文, 1989.
- [13] K. Kubota and M. Iri: PADRE2, version 1 — User’s Manual. *RMI 90-01*, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo, 1990.
- [14] K. Sugihara and M. Iri: VORONOI2 Reference Manual. *RMI 89-04*, Department of Mathematical Engineering and Information Physics, Faculty of Engineering, University of Tokyo, 1989.
- [15] 山本有作: 地理的最適化と高速微分法. 東京大学工学部計数工学科卒業論文, 1990.