数理解析研究所講究録
第648巻 1988年 245-261

非線形最適化問題の数値解法への偏導関数自動導出システムの応用

千葉大 工　山下　稔 (Minoru Yamashita)
吉田　利信 (Toshinobu Yoshida)

1．まえがき

n変数スカラー関数の最適化問題において、勾配ベクトルやヘッセ行列の計算が必要な場合、従来は偏導関数値を数値微分で算出するか、数式処理システムを用いて偏導関数を導出し偏導関数値を算出していた。これらの偏導関数計算法を用いるとき勾配ベクトルの算出に要する計算量は目的関数を算出する計算量のn倍程度、ヘッセ行列やヘッセ行列と探索方向ベクトルとの積を算出するのに要する計算量は目的関数を算出する計算量のn^2倍に比例した程度かかった。さらに数値微分法においてはケタ落ちが生じて正確なヘッセ行列が得られない。このような点から、ヘッセ行列を用いた算法は実用的でないとされてきた。

これらの方法に対して、関数を計算グラフとして表現し、その計算グラフをたどることで偏導関数値を計算する自動微分法が提案された[1)]。自動微分法ではスカラー関数の全ての変数についての正確な偏導関数値が、関数自体を計算するのに要する計算量の高々定数倍の計算量で求められる。また、勾配ベクトル、ヘッセ行列と探索方向ベクトルとの積を算出する計算量は目的関数を算出する計算量の高々定数倍程度、ヘッセ行列を算出する計算量は目的関数を算出する計算量の高々3n倍程度で計算できる。

当研究室では、自動微分法を実現する前処理システムとして、勾配ベクトル、ヘッセ行列、さらには高階の偏導関数を表す数式を導出する偏導関数自動導出システムを作成し、改良を重ねてきた[2) 3)]。

本研究では、非線形最適化問題に偏導関数自動導出システムを応用し、実用的でないとされてきた正確なヘッセ行列を用いたニュートン法や共役勾配法と、一般的に用いられている擬ニュートン法を比較する。また、ハーレー法とニュートン法を比較する。

2．各種最適化算法

R^n で定義された非線形の目的関数f(x)を最小にするxを探索する制約条件なしの非線形最適化問題を考え、これに用いられるいくつかの算法について述べ

る[4]。

以下では、x(k)を反復回数k回目の推定点を表すベクトル、 g(k)を x=x(k)における目的関数f の勾配ベクトル、H(k)を x=x(k)におけるヘッセ行列とする。また、初期点を x(0)とする。

2.1 ニュートン法

ニュートン法の算法を示す。

1. x(0)を与える。
2. k:=0
3. $x(k+1):=x(k)-H^{-1}(k)g(k)$;
4. 収束判定
   NO => k:=k+1; ステップ 3 へ
   YES => 推定極小点 x(k)

ニュートン法を用いたプログラムにおいては、偏導関数自動導出システムを用いて関数値、勾配ベクトル、ヘッセ行列の計算過程を導出した。

ここでは、H(k)、g(k) が与えられた際に、$H^{-1}(k)g(k)$ を算出する方法として、以下の2つを用意した。

(1)逆行列を求めてから、それと勾配ベクトルとの積を計算する方法

これを用いてのニュートン法を以下では NEWTON と記述する。

(2)方程式 g(k)=H(k)Y(k) を改訂コレスキー法を用いてY(k)について解くことにより、$H^{-1}(k)g(k)$ を計算する方法

これを用いてのニュートン法を以下では NEWTCL と記述する。改訂コレスキー法のプログラムは HITAC 数値計算副プログラムライブラリ中の MSL1.DLINLD を用いた。

ニュートン法の特徴を以下に述べる。

1．2次収束し、2次関数に対しては1回の反復で解を得る。

2．反復1回ごとにヘッセ行列の成分を計算しなければならない。もし、これを数値微分で計算するとそれだけで$O(n^2)$の関数計算が必要となる。

3．反復1回ごとにヘッセ行列の逆行列と勾配ベクトルとの積を求めるために連立1次方程式を解かなければならない。これには、$O(n^3)$の計算が必要である。

4．ヘッセ行列が正則でないとき、連立1次方程式の解は一意に定まらない。

5．この算法では極小点、極大点、鞍形点の判別はできず、ヘッセ行列が正定値でない場合、極大点または鞍形点を推定してしまうことがある。

2.2 共役勾配法

反復回数 q回ごとに再スタートを行う共役勾配法の算法を述べる。ここで pは探索方向ベクトルを表す。

1．x(0)を与える。
2．g(0)を求めて、p(0):=-g(0)とする。
3．k:=0;
4．$x(k+1):=x(k)+\alpha_k p(k)$;
5．kが qの定数倍ならばステップ 6 へ それ以外ならステップ 7 へ
6．p(k+1):=-g(k+1); ステップ 6 へ
7．$p(k+1):=-g(k+1)+\beta_k p(k)$;
8．収束判定
　NO => k:=k+1; ステップ 4 へ
　YES => stop ; 推定極小点 x(k)

α_k を決定するためには、$f(x(k)+\alpha_k p(k))$ が最小になるように直線探索を行わなければならない。 α_k の決定方法としては、以下の方法が知られている[5)]。ただし、

$$F(\alpha)=f(x(k)+\alpha p(k)) \qquad (1)$$

とする。

1．$F(\alpha)$を用いる方法（黄金分割法、2次補間法）
2．$F(\alpha)$、$F'(\alpha)$を用いる方法（3次補間法、セカント法）
3．$F(\alpha)$、$F'(\alpha)$、$F''(\alpha)$ を用いる方法（ニュートン法）

β_k の決定方法としては以下の3つの方法が知られている[5)]。

$$\beta_k=||g(k+1)||^2/||g(k)||^2 \qquad (2)$$

$$\beta_k=(g(k+1)-g(k),g(k+1))/(g(k+1)-g(k),p(k)) \qquad (3)$$

$$\beta_k=(g(k+1)-g(k),g(k+1))/||g(k)||^2 \qquad (4)$$

$$\beta_k=(g(k+1),H(k+1)p(k))/(p(k),H(k+1)p(k)) \qquad (5)$$

(2)式、(3)式、(4)式、(5)式はそれぞれ Fletcher-Reeves の公式、Sorenson-Wolfe の公式、 Polak-Ribière-Polyak の公式、Hestenes-Stiefel-Danielの公式と呼ばれている。このなかでよく用いられるのは (2)式である。

共役勾配法を用いたプログラムにおいては、関数値、勾配ベクトル、ヘッセ行列の計算過程を偏導関数自動導出システムを用いて導出した。

数値実験においては、α_kとして、

$$\alpha_k=-(g(k),p(k))/(p(k),H(k)p(k)) \qquad (6)$$

を採用した。このα_kは、ニュートン法の反復１回目のステップ幅にあたる。すなわち、ここでは$f(x(k)+\alpha_k p(k))$が最小になるような正確な直線探索を行ったのではなく、荒い直線探索を行ったことになる。この荒い直線探索を用いることで直線探索に要する計算時間が短かくなり、極小点探索に要する計算時間の高速化が期待できる。これを確かめるために荒い直線探索と正確な直線探索との比較を拡張ローゼンブロック関数について行った。

数値実験においてβ_kとして採用したのは(5)式の Hestenes-Stiefel-Daniel の公式である。(5)式はヘッセ行列の算出が実用的でないということから、従来は用いられなかった。(5)式を用いた共役勾配法により極小点探索が効率的に行われることを確認できれば、これは(5)式の有効性だけでなく、偏導関数自動導出システムを用いて導出したヘッセ行列の計算効率が高いことも確認できたことを意味する。

また、拡張ローゼンブロック関数においては、共役勾配法において一般によく用いられる(2)式のFletcher-Reeves の公式についても数値実験を行った。

さらに、拡張ローゼンブロック関数においてはβ_kを次式で決定する方法も採用し、数値実験を行った。

$$\beta_k=(g(k+1),H(k)p(k))/(p(k),H(k)p(k)) \qquad (7)$$

この式は(2)式において H(k)p(k)を導出しているので、H(k+1)p(k) を算出するかわりにその値を H(k)p(k) で置き換えたものである。

以下では、β_k の決定方法として (5)式、(2)式、(7)式を用いた算法をそれぞれCG1, CG2, CG3と記述する。

(1)CG1 (Hestenes-Stiefel-Daniel の公式)

ここで、q は再スタートを行う反復回数である。

1. x(0)を与える。
2. g(0)を求めて、p(0):=-g(0)とする。
3. k:=0;
4. α_k:=-(g(k),p(k))/(p(k),H(k)p(k));
5. x(k+1):=x(k)+α_kp(k);

6. kが qの定数倍ならばステップ 7へ それ以外ならステップ 8へ
7. p(k+1):=-g(k+1); ステップ 10へ;
8. β_k:=(g(k+1),H(k+1)p(k))/(p(k),H(k+1)p(k));
9. p(k+1):=-g(k+1)+β_kp(k);
10. 収束判定
    NO => k:=k+1; ステップ 4へ
    YES => 推定極小点 x(k)

(2)CG2 (Fletcher-Reeves の公式)
CG1におけるステップ 7のβ_kを (2)式
β_k:=||g(k+1)||^2/||g(k)||^2;
で置き換え、CG1の手順で計算する方法。

(3)CG3
CG1におけるステップ 7のβ_kを(7)式
β_k:=(g(k+1),H(k)p(k))/(p(k),H(k)p(k));
で置き換え、CG1の手順で計算する方法。

共役勾配法の特徴を以下に述べる。

1．n回あたりの超1次収束をする。n変数の2次関数に対しては高々n回の反復で解を得る。

2．β_k を決定するのに(3)式、(4)式、(5)式を採用すれば、ヘッセ行列を用いなくても実現できる。また、その場合、ヘッセ行列の要素を保持する必要がないので、少ない記憶領域で実現できる。

2.3 ハーレー法

次にハーレー法の算法を示す。

1. x(0)を与えて、k:=0;とする。
2. x(k+1):=x(k)-(H(k)-1/2L(k)(H^{-1}(k)g(k))$^{-1}$g(k);
3. 収束判定
    NO => k:=k+1; ステップ 2へ
    YES => 推定極小点 x(k)

ただし、Lは、

$$L_{ijk} = \partial H_{ij} / \partial x_k \qquad (8)$$

である。また、$L(k)(H^{-1}(k)g(k))$は３階のテンソル $L(k)$ とベクトル $H^{-1}(k)g(k)$ との積を意味し、その積はヘッセ行列と同じ型の行列となる。その行列をＭとすると、その積の定義は

$$M_{ij}=\Sigma\Sigma(\partial H_{ij}/\partial x_k)\cdot H^{-1}{}_{kl}\cdot g_l=\Sigma\Sigma L_{ijk}\cdot H^{-1}{}_{kl}\cdot g_l \quad (9)$$

である。

ハーレー法を用いたプログラムにおいては、関数値、勾配ベクトル、ヘッセ行列を表す計算過程を偏導関数自動導出システムを用いて導出した。また、ヘッセ行列の各成分をベクトルの各成分で微分して得られる３階のテンソルとベクトルとの積$(\Sigma(\partial H_{ij}/\partial x_k)y_k)$を表す計算過程を偏導関数自動導出システムを用いて導出した。

また、反復１回あたりに２個所に現れる、ヘッセ行列の逆行列とベクトルの積を求めるために、連立方程式を改訂コレスキー法を用いて解いた。ハーレー法は、クラッグーレビ関数と関数Ｆ５について適用した。

この算法を以下では HALLEY と記述する。

ハーレー法の特徴を以下に述べる。

１．逆行列とベクトルとの積を求めるために連立１次方程式を反復１回あたり、２回解かなければならない。

２．３階のテンソルとベクトルとの積を導出する必要がある。

３．行列が非正則な時はこの算法は使えない。

４．３次収束する。

２.４ 擬ニュートン法

擬ニュートン法は、関数値と勾配ベクトルを用いて、ヘッセ行列の逆行列H'^{-1}を逐次近似で求めて、そのH'^{-1} を用いてニュートン法を行う算法である。

次に擬ニュートン法（ＤＦＰ公式）の算法を示す。

1. $x(0)$を与えて、$g(0)$を求めて
   $p(0):=-g(0)$; $H'^{-1}(0):=I$; $k:=0$;（Ｉは単位行列）
   とする。
2. 方向ベクトル $p(k)$ を
   $p(k):=-H'^{-1}(k)g(k)$;
   と定める。
3. $F(\alpha_k):=F(x(k)+\alpha_k p(k))$;
   を最小にするα_kを定める。

4. $x(k+1):=x(k)+\alpha_k p(k)$; $y(k):=g(k+1)-g(k)$;
5. 収束判定
   YES => stop ; 推定極小点 x(k)
   NO => ステップ 6へ
6. $H'^{-1}(k+1):=H'^{-1}(k)+\alpha_k^2 p(k)p^t(k)/(\alpha_k p(k),y(k))$
   $-H'^{-1}(k)y(k)y^t(k)H'^{-1}(k)/(y(k),H'^{-1}(k)y(k))$;
7. k:=k+1; ステップ 2へ

擬ニュートン法を用いたプログラムは、日立製作所作成のHITAC 数値計算副プログラム・ライブラリ中の MSL.DDAVDM を用いた。このプログラムと他のプログラムとを実行させて、その結果を比較することにより、偏導関数自動導出システムから導出されたヘッセ行列を用いた際の各種算法の評価が行える。また、その比較により偏導関数自動導出システムの有用性を確認できる。

この算法を以下では DV と記述する。

擬ニュートン法の特徴を以下に述べる。

1．逐次近似で求めている行列がヘッセ行列の逆行列にほぼ等しくなると、ニュートン法と同様の収束の速さが得られる。

2．ヘッセ行列の計算を行う必要がない。

3．ニュートン法にみられるようなヘッセ行列が正定値でないときの不安定性がない。

3．最適化問題の数値実験例

第2章で述べた算法を、偏導関数自動導出システムを用いていくつかのテスト関数[6]に適用し、数値実験を行った。

以下では Fmin は極小値、x は推定点を表すベクトル、x^* は極小点を表すベクトル、ERR は推定点と極小点とのユークリッド距離

$$ERR=|x-x^*| \qquad (10)$$

を表す。また、Tは極小値探索に要する全計算時間、tは反復1回あたりの計算時間、qは共役勾配法での再スタートを行う反復回数を表す。

収束判定は次式の丸め誤差評価関数 (ERF)を用いて行う[7]。

$$ERF=\varepsilon(1/3\Sigma((\partial f/\partial v_j)v_j)^2)^{1/2} \qquad (11)$$

ここで v_j は、関数を四則演算や初等関数などの基本演算列に分解した際、j番目のステップでの基本演算の値（中間変数）である。また、εはいわゆるマシンエプシロンで、浮動小数点演算で（1＋ε）≠1となるような最小の数であ

る。この評価が実際に丸め誤差の振舞をとらえていることは確かめられている[9)]。偏導関数自動導出システムによると、(11)式で与えられる関数計算時に生じる精密な丸め誤差評価関数 (ERF)が容易に導出できる。そこで、ニュートン法、共役勾配法、ハーレー法において次式を反復停止条件として収束判定をおこなった。

$$|F| \leq ERF \qquad (12)$$

擬ニュートン法においては次式を反復停止条件として収束判定をおこなった。

$$|x(k+1)-x(k)| \leq \varepsilon \quad \text{かつ} \quad |p(k)| \leq \varepsilon \qquad (13)$$

ここで p(k) は探索方向ベクトルを表す。

数値実験は千葉大学情報処理センター (HITAC M-260K) の VOS3. FORTRAN 77 OPT(3)を用いて倍精度計算で行った($\varepsilon = 2^{-56} \sim 1.4\times10^{-17}$)。

3.1 ビール関数 (Beale's function ; n=2)

$$F=\Sigma\{C_i-x_1(1-x_2^i)\}^2,\ C1=1.5, C2=2.25, C3=2.625 \qquad (14)$$

$$x(0)=(1.0,0.0), Fmin=0.0, x^*=(3.0,0.5)$$

表1　ビール関数の各方法による極小値

算法	反復回数	F	ERR	T [ms]	t [μs]
NEWTON	9	0.0	0.0	0.254	28.2
CG1	15	0.0	0.0	0.529	35.3
DV	12	0.0	0.2248D-15	1.00	83.3

ただし、q＝8

3.2 ローゼンブロック関数 (Rosenbrock's function ; n=2)

$$F=100*(x_2-x_1)^2+(1-x_1)^2 \qquad (15)$$

$$x(0)=(-1.2,1.0),\ Fmin=0.0\ ,x^*=(1.0,1.0)$$

表2　ローゼンブロック関数の各方法による極小値

算法	反復回数	F	ERR	T [ms]	t [μs]
NEWTON	9	0.0	0.0	0.119	13.2
CG1	31	0.1945D-31	0.4388D-16	0.498	16.1
DV	28	0.0	0.0	2.71	96.8

ただし、q＝8

共役勾配法では、ERR が 10^{-16}以下になっている。倍精度実数型においては、十進法で16けたの有効数字なので、この数字は推定点の位置ベクトルの成分が極小点と最終桁まで一致している事を示す。

3.3 クラッグ、レビ関数 (Cragg and Levy's function ; n=4)

$$F=(\exp(x_1)-x_2)^4+100(x_2-x_3)^6+\tan^4(x_3-x_4)+x_1^8+(x_4-1)^2 \quad (16)$$

$x(0)=(1.01,2.0,2.01,2.02)$, $Fmin=0.0$, $x^*=(0.0,1.0,1.0,1.0)$

表3 クラッグ、レビ関数の各方法による極小値

算法	反復回数	F	ERR	T [ms]	t [μs]
NEWTCL	54	0.2510D-28	0.1123D-4	7.71	142
CG1	249	0.8234D-20	0.2710D-3	15.6	62.7
DV	75	0.1542D-21	0.3043D-3	27.9	372
HALLEY	28	0.3318D-28	0.1179D-4	7.50	268

ただし、q＝8

反復１回あたりの計算時間がハーレー法ではニュートン法のほぼ２倍になったのは、反復１回あたりに逆行列とベクトルとの積を算出するために連立１次方程式を解く計算が、ハーレー法では２回必要であったのに対して、ニュートン法では１回でよいからである。

3.4 F5 (n=5)

$$F=(2x_1+x_2-3x_3+6x_4+5x_5-4)^4+(x_1-2x_2-6x_3+4x_4-5x_5+2)^2$$
$$+\{(x_1-1)(2x_2-1)(3x_3-1)(4x_4-1)(5x_5-1)\}^2 \quad (17)$$

$x(0)=(1.05,0.55,0.4,0.3,0.25)$, $Fmin=0.0$,

$x^*=(1,1/2,1/3,1/4,1/5)$

表4 F5の各方法による極小値

算法	反復回数	F	ERR	T [ms]	t [μs]
NEWTCL	25	0.8036D-18	0.9200D-2	3.75	150
CG1	300	0.6625D-26	0.4724D-1	18.7	62.3
DV	53	0.4193D-30	0.5628D-1	15.6	294
HALLEY	35	0.6375D-18	0.6749D-2	11.0	314

ただし、q＝15

ニュートン法及びハーレー法においては連立１次方程式を解くために改訂コレスキー法を用いたが、それぞれ反復25回、35回でその行列が非正則になってしまった。CG1 においては反復回数22回(計算時間 1.6ms)で関数値が10^{-10} になったが、それ以後の収束が悪かった。

３．５ 10変数の拡張ローゼンブロック関数(Extended Rosenbrock function;n=10)

以下の関数については、CG2、CG3 の実験も行った。

$F=\Sigma\{100(x_i-x_{i-1}^2)^2+(1-x_{i-1})^2\}$ (18)

$x(0)=(-1.2,1.0,-1.2,\cdots,1.0)$, $Fmin=0.0$,

$x^*=(1.0,1.0,1.0,\cdots,1.0)$

表５　１０変数の拡張ローゼンブロック関数の各方法による極小値

算法	反復回数	F	ＥＲＲ	T [ms]	t [μs]
NEWTCL	33	0.0	0.0	12.5	379
CG1	137	0.2836D-29	0.1628D-14	14.2	104
CG2	391	0.5177D-29	0.4965D-15	34.8	89.0
CG3	100	0.3988D+1	0.1994D+1	9.17	91.7
DV	116	0.0	0.0	71.5	616

ただし、ｑ＝40

収束の様子を付録１に示す。

CG1、CG2において関数値が10^{-20} になるのに要する計算時間は、それぞれ13.3ms、22.5ms となり、CG2では CG1の 1.7倍の計算時間を必要とした。またCG2 では関数値が10^{-27}以下になると関数値の減少の様子が急速に鈍った。 CG1ではそのような現象は起こらずに関数値が 0.2836･10-29 となる推定点に到達した。

３．６ 20変数の拡張ローゼンブロック関数(Extended Rosenbrock function;n=20)

$F=\Sigma\{100(x_i-x_{i-1}^2)^2+(1-x_{i-1})^2\}$ (19)

$x(0)=(-1.2,1.0,-1.2,\cdots,1.0)$, $Fmin=0.0$,

$x^*=(1.0,1.0,1.0,\cdots,1.0)$

収束の様子を付録２に示す。

CG1、CG3 において関数値が10^{-20} になるのに要する反復回数はそれぞれ258回、256回、計算時間は、それぞれ 54.4ms、46.0msとなった。これは H(k)p(k)を算出せ

ずに、その値として H(k+1)p(k)を用いることによっても極小値が探索でき、その計算時間だけ高速化できる場合のあることを示している。

表6　２０変数の拡張ローゼンブロック関数の各方法による極小値

算法	反復回数	F	ＥＲＲ	T [ms]	t [μs]
NEWTCL	45	0.3896D-31	0.1963D-15	73.1	1620
CG1	292	0.3496D-30	0.5375D-15	61.4	210
CG2	651	0.1049D-28	0.3167D-14	114	175
CG3	309	0.4459D-30	0.5375D-16	55.4	179
DV	233	0.0	0.0	438	1880

ただし、ｑ＝80

３．７ 30変数の拡張ローゼンブロック関数(Extended Rosenbrock function;n=30)

$$F=\Sigma\{100(x_i-x_{i-1}^2)^2+(1-x_{i-1})^2\} \qquad (20)$$

$x(0)=(-1.2,1.0,-1.2,\cdots,1.0)$, Fmin=0.0,

$x^*=(1.0,1.0,1.0,\cdots,1.0)$

表7　３０変数の拡張ローゼンブロック関数の各方法による極小値

算法	反復回数	F	ＥＲＲ	T [ms]	t [μs]
NEWTCL	58	0.0	0.0	235	4050
CG1	301	0.2561D-29	0.1442D-15	93.7	311
CG2	633	0.2473D-29	0.1251D-14	162	256
CG3	508	0.1159D-30	0.1963D-16	135	266
DV	907	4.769D0	4.769D0	4080	4500

ただし、ｑ＝60

収束の様子を付録３に示す。

この関数に擬ニュートン法を用いた場合、反復回数 907回で近似ヘッセ行列が正定値でなくなり、極小点探索に失敗した。

CG1、CG3 で関数値が10^{-20} になるのに要する反復回数はそれぞれ264回、370回、計算時間は、それぞれ 82.2ms、98.3msとなった。これはH(k+1)p(k)を H(k)p(k)のかわりに用いたことにより、極小点から離れた地点では H(k)と H(k+1)とに差異が生じ、それだけ推定点が超１次収束する領域に到達するまでの反復回数が増加したためだと思われる。

3.8　拡張ローゼンブロック関数にニュートン法を用いた場合の改訂コレスキー法に要する計算時間の測定結果

次に、拡張ローゼンブロック関数にニュートン法を用いた際、反復１回あたりに改訂コレスキー法に要する計算時間(T1)とその他の部分の計算に要する計算時間(T2)を測定した。測定結果を表８に示す。

表８　改訂コレスキー法に要する計算時間

変数の個数	T1[ms]	T2[ms]	T1:T2
10	0.240	0.138	1.75:1
20	1.08	0.544	1.98:1
30	3.12	0.931	3.35:1

3.9　拡張ローゼンブロック関数に共役勾配法の正確な直線探索を用いた場合と荒い直線探索を用いた場合の比較[8)]

次に拡張ローゼンブロック関数に CG2を用いた際、α_kを定めるのに正確な直線探索を行った場合と荒い直線探索を行った場合の反復回数と計算時間を比較する。ここで用いた正確な直線探索とは、探索方向ベクトルに沿ってニュートン法を行い、推定点での方向微分係数(探索方向ベクトルと勾配ベクトルとの内積)が10^{-6}以下になったら、直線探索から抜け出すというものである。荒い直線探索とは、前述したように(6)式を１度計算するだけでα_k を決定するものである。

関数値が 10^{-20}以下になるのに要した計算時間と反復回数を表９に示す。表中の括弧内の数字は反復回数を示す。

表９　正確な直線探索を用いた場合と荒い直線探索を用いた場合の計算時間と反復回数

変数の個数	正確な直線探索	荒い直線探索
10	36.5 ms (236)	22.3 ms (254)
20	171 ms (514)	98.5 ms (568)
30	289 ms (576)	157 ms (625)

4．あとがき

本報告では、偏導関数自動導出システムにより導出された正確なヘッセ行列を用いてニュートン法や共役勾配法の数値実験を行った。また、実用的な算法とし

て推奨されてきた擬ニュートン法の数値実験を行った。また、ハーレー法を取り上げ、実験結果の報告を行なった。これより以下のことが確認できた。

（1）擬ニュートン法と比較して、ニュートン法は少ない反復回数で収束した。また、ニュートン法における反復１回あたりの計算時間も擬ニュートン法より短かった。

（2）ニュートン法では変数の個数が多くなると、連立１次方程式を解く計算が大きな比重を占めるようになった。

（3）正確なヘッセ行列を用いて探索方向を決定する共役勾配法は、勾配ベクトルで探索方向を決定する共役勾配法より、少ない反復回数で収束した。

（4）従来は３階の偏導関数を表すテンソルの導出が困難であったため用いられなかったハーレー法は、ニュートン法と比較して少ない反復回数で収束したが、収束までに要した計算時間はほぼ同じであった。

従来、正確なヘッセ行列を用いたニュートン法、共役勾配法は、正確なヘッセ行列を計算する効率がよくないという理由で実用的でないとされてきた。しかし、本システムを用いることにより、これらの算法は十分、実用的であることが確認された。以上より、ヘッセ行列を用いる算法、さらに高階の偏導関数を用いる算法に偏導関数自動導出システムを利用することの価値は大きいといえる。

今後の課題としては、システムを改良することによって、ヘッセ行列を用いたニュートン法あるいは共役勾配法の副プログラムを自動的に導出することがあげられる。また、ヘッセ行列やさらに高階の偏導関数を用いた新たな算法を考えることも重要である。

参考文献

1) Iri, M. : Simultaneous Computation of Functions Partial Derivatives and Estimates of Rouding Errors - Complexity and Practicality, Jpn. J. Appl. Math., Vol. 1, No. 2, pp. 223-252 (1984).

2) 山下稔：関数およびその偏導関数の計算過程を表す計算グラフの自動作成，千葉大学工学部電気工学科卒業論文(1986).

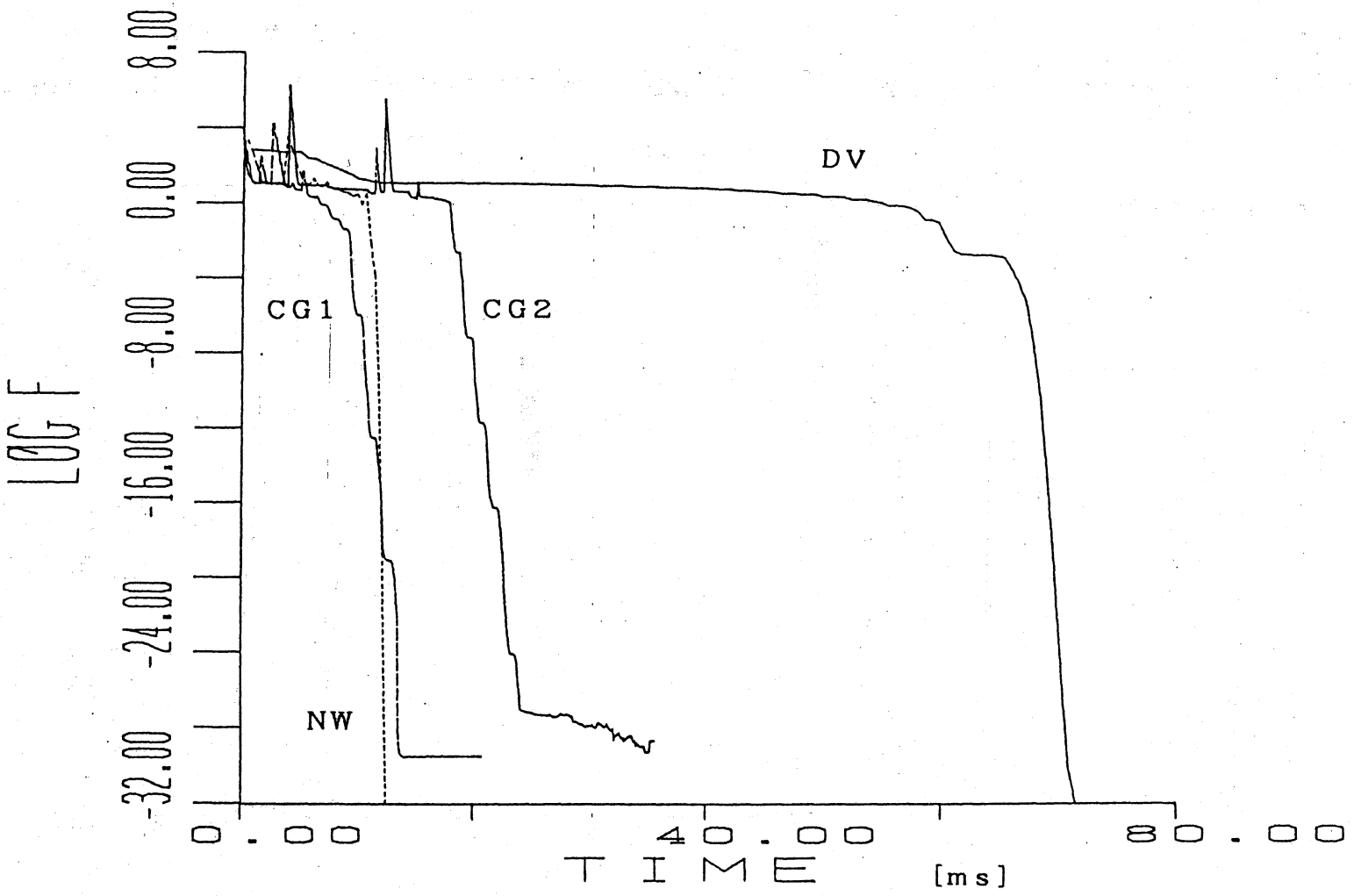
3) 山下稔，吉田利信：計算グラフを用いた数値計算のための数式処理システム，統計数理研究所研究報告「グラフ理論の数値計算への応用」(1987).

4) L. C. W. ディクソン（松原正一訳）：非線形最適化計算法，培風館(1974).

5) 今野浩，山下浩：非線形計画法，日科技連，東京（1978).

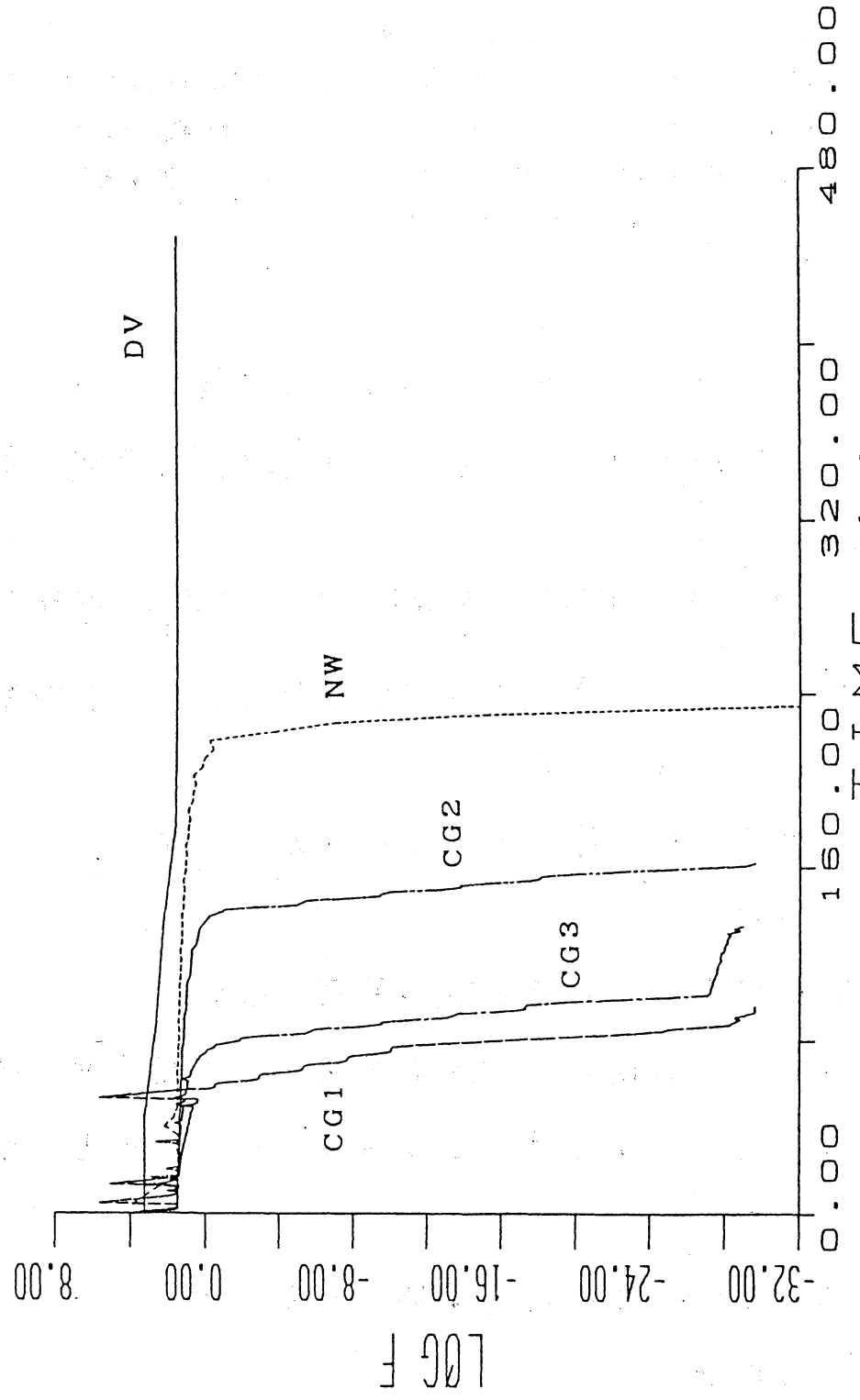
6) 西原薫，星守，戸田英雄：計算グラフによる２階偏導関数の自動計算とその応用，統計数理研究所研究報告「グラフ理論の数値計算への応用」(1987).

7) 伊理正夫，土谷隆，星守：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. 12，pp. 119-137.
8) 土谷隆：高速微分法および丸め誤差推定法とその応用，東京大学大学院工学系計数工学専門課程修士論文(1986)



付録１　拡張　Ｒｏｓｅｎｂｒｏｃｋ　関数　（　Ｎ　＝　１０　）
の収束の様子

付録2　拡張 Rosenbrock 関数 (N = 20) の収束の様子



付録3 拡張 Rosenbrock 関数 (N = 30) の収束の様子