非線形連立方程式の解法

Masato Shimura

2006 年 11 月 7 日

目次

1 連立方程式とニュートン法

1.1 微分関数 D. を用いる

J は微分のプリミティブ (D.) を備えている。

N.Thomson が 1994 年にこの D. を用いる簡潔な多変数のニュートン法を発表している。

```
NB. N.Thomson
new_2=: 1 : ’] - x (%.|:) x D.1’(^:17)("1)
```

`(^:17)` は反復回数を指定する任意の整数で、例として 17 が与えられている。

このスクリプトについては以前にも紹介しているが、再度レビューしてみよう。ニュートン法で解く関数は動詞で定義する。

1.1.1　2 変数

$$\left(\begin{array}{rc} f_0 = & e^x + xy - 1 \\ g_0 = & sinxy + x + y + 2 \end{array} \right.$$

Tacit と Explicit の表現。どちらを用いても良い。

Tacit	Explicit

`f0=: -&1@(*/)+^@{.`
`g0=: +&2@(+/)+1&o.@*/`
 	
`%f00=: 3 : '(^{.y )+(*/y )-1'`
`%g00=: 3 :  '2+(+/ y )+1&o.*/ y '`
	`f0=: 3 : '(^{.y )+(*/y )-1'`
	`g0=: 3 :  '2+(+/ y )+1&o.*/ y '`

連立では `(f0,g0)` **とカンマで動詞を連結して用いる。**

```
   (f0,g0) new_2 1 1
_9.4112e_6 _2
```

1.1.2　3 変数の例

$$\left(\begin{array}{ll} h_1(x,y,z) & = 16x^4 + 16y^4 + z^4 - 16 \\ h_2(x,y,z) & = x^2 + y^2 + z^2 - 3 \\ h_3(x,y,z) & = x^3 - y \end{array} \right.$$

```
  h1=:3 : '  (16 16 1 +/ . *   y ^4)- 16'

  h2=:3 : '3- +/ y ^2 '

  h3=:3 : ' (1{y )-~(({. y )^3)'

 (h1,h2,h3) new_2 1 1 1
0.877966 0.676757 1.33086

 (h1,(h2,h3)) new_2 1 1 1
0.877966 0.676757 1.33086
```

(h1,(h2,h3)) としても結果は同じである。h1,h2,h3,h4... と続けていけば, 多変数に適用できる。

1.2 ニュートン法とヤコビアン

ニュートン法の反復公式

ニュートン法の反復式	マトリクスの場合 F is vector-valued
$x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}$	$x^{(n+1)} = x^{(n)} - \dfrac{F(x^{(n)})}{F'(x^{(n)})}$

$dF = F'(x^n)\Delta x$
$x + \Delta x$ は次により求める、

$$\text{df=} \frac{\partial f}{\partial x_1}\Delta x_1 + \frac{\partial f}{\partial x_2}\Delta x_2 + \frac{\partial f}{\partial x_3}\Delta x_3 + \cdots + + \frac{\partial f}{\partial x_m}\Delta x_m$$

$F(x)$ の偏微分 $F'(x)$ は *Jacobian* と呼ばれる。そのスカラ表現

$$
F' = \begin{bmatrix}
\frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \cdots & \frac{\partial f_1}{\partial x_m} \\
\frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \cdots & \frac{\partial f_2}{\partial x_m} \\
\frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \cdots & \frac{\partial f_3}{\partial x_m} \\
 & \vdots & & \vdots & \\
\frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \frac{\partial f_m}{\partial x_3} & \cdots & \frac{\partial f_m}{\partial x_m}
\end{bmatrix}
$$

次の部分を反復させることが NEWTON 法のキーポイントである。

$[J(x^{(n)})]v^{(n)} = -F(x^{(n)})$

ニュートン法の反復は次による。

$x^{(n+1)} = x^{(n)} - [J(x^{(n)})]^{-1}F(x^{(n)})$

$v^{(n)} = -[J(x^{(n)})]^{-1}F(x^{(n)})$

$[J(x^{(n)})]v^{(n)} = -F(x^{(n)})$

$x^{(n+1)} = x^{(n)} + v^{(n)}$

1.2.1 例題

$$
\left(\begin{array}{ll}
x_1^3 - 2x_2 - 2 & = 0 \\
x_1^3 - 5x_3^2 - 7 & = 0 \\
x_2 x_3^2 - 1 & = 0
\end{array}\right.
$$

$f_1(x_1, x_2, x_3) = x_1^3 - 2x_2 - 2 = 0$ $f_1(x_1, x_2, x_3) = x_1^3 - 5x_3^2 - 7 = 0$ $f_1(x_1, x_2, x_3) = x_2 x_3^2 - 1 = 0$	F(x)= $\begin{bmatrix} f_1(x_1, x_2, x_3) \\ f_1(x_1, x_2, x_3) \\ f_1(x_1, x_2, x_3) \end{bmatrix}$ = $\begin{bmatrix} x_1^3 - 2x_2 - 2 \\ x_1^3 - 5x_3^2 - 7 \\ x_2 x_3^2 - 1 \end{bmatrix}$
Starting from initial vector $x^(0) = [\ 1\ 1\ 1\ ]$	
Jacobian Matrix J(x)= $\begin{bmatrix} 3x_1^2 & -2 & 0 \\ 3x_1^2 & 0 & -10x_3 \\ 0 & x_3^2 & 2x_2x_3 \end{bmatrix}$	

1.3 ヤコビアンを直接定義するニュートン法

N.Thomson の連立方程式の NEWTON 解法は J の微分関数 (D.) を用いたこのエレガントな方法で、解析関数には有効であるが頑強ではない。

```
new_2=: 1 : '] - x (%.|:) x D.1'(^:17)("1)
```

D. が通らない場合はヤコビアンを直接手で作って、反復する。この方法は手数はかかるが頑強である。

$F(x)$ を動詞で定義する。	`f0=: 3 : '_2 + 1 _2 +/ . *(^&3 1 )0 1 { y'`
	`f1=: 3 : '7 + 1 _5 +/ . *(^&3 2 )0 2{ y'`
	`f2=: 3 : '_1+ */(^& 1 2)1 2 { y'`
	`fnewx=: f0,f1,f2`

初期値 1 1 1 を与える	`   fnewx 1 1 1` `_3 3 0`

ヤコビアンを直接, 定義する	`NB. J(x)`
	`j0=: 3 : '(3 * (^&2)0{y ),_2,0'`
	`j1=: 3 : '(3 *(^&2 )0  {y ),0,_10*2{y '`
	`j2=: 3 : '0,((^&2)(2{y )),2 * */1 2 {y '`
	`jnewx=: 3 : ';("1) ,.(j0;j1;j2) L:0 y '`

初期値 1 1 1 を与える	`   jnewx 1 1 1`
	`3 _2   0`
	`3  0 _10`
	`0  1   2`

$v^{(0)} = \dfrac{-F(x^0)}{J(x^0)}$	`   (- fnewx 1 1 1 ) %. jnewx 1 1 1` `0.428571 _0.857143 0.428571`

$x^{(1)} = x^{(0)} + v^{(0)}$	`X=. 1 1 1 +  (- fnewx 1 1 1 ) %. jnewx 1 1 1` `1.42857 0.142857 1.42857`
$Fx^{(1)}$	`fnewx X` `0.629738 _0.28863 _0.708455`

$Jx(1)$	`jnewx X`
	`6.12245       _2        0`
	`6.12245        0 _14.2857`
	`      0 2.04082 0.408163`

$v^{(1)} = \frac{-F(x^1)}{J(x^1)}$	`(- fnewx X)%. jnewx X` `0.0115397 0.350195 _0.0152585`
$x^{(2)} = x^{(1)} + v^{(1)}$	`X=. X+  (- fnewx X)%. jnewx X` `1.44011 0.493052 1.41331`
$v^{(2)} = \frac{-F(x^1)}{J(x^1)}$	`(- fnewx X)%. jnewx X` `0.00214417 0.00695637 0.000902039`
$x^{(3)} = x^{(2)} + v^{(2)}$	`X+  (- fnewx X)%. jnewx X` `1.44226 0.500008 1.41421`

ニュートンの反復法のスクリプト。*fnewx,jnewx* の名称は固定している。反復回数は 20 回としている。(適宜変更する)

```
newton_iteration=: 3 : 0
X=: y
COUNTER=: 0
ANS=: <''
while. COUNTER < 20 do.
V=: (- fnewx X) %. jnewx X
X=: X + V
ANS=: ANS,<X
COUNTER=. >:COUNTER
end.
;("1)}.,.ANS
)
```

4 回の反復で収束している。

```
   newton_iteration 1 1 1
1.42857 0.142857 1.42857
1.44011 0.493052 1.41331
1.44226 0.500008 1.41421
1.44225      0.5 1.41421
1.44225      0.5 1.41421
1.44225      0.5 1.41421
1.44225      0.5 1.41421
1.44225      0.5 1.41421
(20 回まで反復)
```

1.4　Broyden 法

Brain Bradie [A friendly Introduction to Numeric Analsis] Pearson 2006 は Newton 法の別の解法として Broyden 法を紹介している。Broyden 法の原典として次のような文献が紹介されている。

C.G.Broyden [Quasi Newton Method and their application to function minimization] ’Mathematics of Computing 19’ 1967

Broyden 法はヤコびアンを用いるのは最初の一回のみで、次からは $F(x)$ の差分から求める反復解法である。

$$\mathrm{A}_k(x^{(k)} - x^{(k-1)}) = F(x^{(k)}) - F(x^{(k-1)})$$

$$\mathrm{f}\,{}'(\mathrm{x}_k) = \frac{f(x_x) - f(x_{(k-1)})}{x_k - x_{k-1}}$$

$$\mathrm{A}_k = A_{k-1} - \frac{y - A_{k-1}\Delta}{\Delta^T\Delta}\Delta^T$$

$$\mathrm{y{=}F(x}^k) - F(x^{k-1})$$

$$\Delta = x^k - x^{k-1}$$

$$\mathrm{A}_k^{-1} = \left(A_{k-1}\frac{y - A_{k-1}\Delta}{\Delta^T\Delta}\Delta^T\right)^{-1}$$

$$=\mathrm{A}_{k-1}^{-1} + \frac{(\Delta - A_{k-1}^{-1}y)\Delta^T A_{k-1}^{-1}}{\Delta^T A_{k-1}^{-1}y}$$

$F(x^{(0)})$	`fnewx 1 1 1` `_3 3 0`

$A_0 = J(x^{(0)})$	`jnewx 1 1 1`
	`3 _2   0`
	`3  0 _10`
	`0  1   2`

A_0^{-1}	`%.  jnewx 1 1 1`
	`0.238095  0.0952381  0.47619`
	`_0.142857   0.142857 0.714286`
	`0.0714286 _0.0714286 0.142857`

$v^{(0)} = -A_0^{-1}F(x^{(0)})$	`(%.  jnewx 1 1 1) +/ . * -fnewx 1 1 1` `0.428571 _0.857143 0.428571`
$x^{(1)} = x^{(0)} + v^{(0)}$	`1 1 1 + (%.  jnewx 1 1 1) +/ . * -fnewx 1 1 1` `1.42857 0.142857 1.42857`
$F(x^{(1)})$	`fnewx x0` `0.629738 _0.28863 _0.708455`
$y = F(x^{(1)}) - F(x^{(0)})$	`(fnewx x0) - fnewx 1 1 1` `3.62974 _3.28863 _0.708455`

$A_0^{-1}y$	`(%. jnewx 1 1 1) +/ . * y` `0.213661 _1.49438 0.392961`
$\Delta = v^{(0)}$ $\Delta^T A_{(0)} y$	`v0 +/ . * (%. jnewx 1 1 1) +/ . * y` `1.54088`

$\mathrm{A}_k = A_{k-1} - \frac{y - A_{k-1}\Delta}{\Delta^T \Delta}\Delta^T$	`%. DELTA broyden_sub0 Y;jnewx 1 1 1`
	` 0.276741   0.133884 0.785355`
	`_0.108758   0.176956 0.987078`
	`0.0782484 _0.0646088 0.197416`

$v^{(1)} = -A_1^{-1} F(x^{(1)})$	
$x^{(2)} = x^{(1)} + v^{(1)}$	

```
   broyden 1 1 1
       1        1        1

 1.42857 0.142857 1.42857
 1.84933 0.961721 1.50051
 1.28191 0.844231 1.44359
 1.66612 0.478836 1.39738
 1.31599 0.667999 1.45464
 1.56872 0.472869 1.39122
 1.34922 0.557366 1.43495
  1.4907  0.45568 1.39655
 1.40136 0.518196 1.42501
 1.46108 0.478744 1.40543
 1.42285 0.503729 1.41856
 1.44784  0.48762 1.40993
 1.43177 0.497942 1.41557
  1.4422 0.491276 1.41191
 1.43548 0.495565 1.41428
 1.43982 0.492796 1.41275
 1.43702 0.494581 1.41374
 1.43883 0.493428  1.4131
 1.43766 0.494171 1.41351
 1.43842 0.493692 1.41324
 1.43793 0.494001 1.41342
 1.43824 0.493802  1.4133
 1.43804  0.49393 1.41338
 1.43817 0.493847 1.41333
 1.43809 0.493901 1.41336
 1.43814 0.493866 1.41334
 1.43811 0.493889 1.41335
 1.43813 0.493874 1.41334
 1.43812 0.493884 1.41335
 1.43812 0.493878 1.41335
 1.43812 0.493881 1.41335
 1.43812 0.493879 1.41335
 1.43812 0.493881 1.41335
 1.43812 0.493879 1.41335
 1.43812  0.49388 1.41335
 1.43812  0.49388 1.41335
 1.43812  0.49388 1.41335
```

1.5 Script

```
NB. ----NEWTON ---------------------------------
NB. F(x)
NB. OK fixed _2 7 _1
f0=: 3 : '_2 + 1 _2 +/ . *(^&3 1 )0 1 { y'
f1=: 3 : '7 + 1 _5 +/ . *(^&3 2 )0 2{ y'
f2=: 3 : '_1+ */(^& 1 2)1 2 { y'
fnewx=: f0,f1,f2
NB. ------------------------------------
```

```
NB. J(x)
j0=: 3 : '(3 * (^&2)0{y ),_2,0'
j1=: 3 : '(3 *(^&2 )0  {y ),0,_10*2{y '
j2=: 3 : '0,((^&2)(2{y )),2 * */1 2 {y '
jnewx=: 3 : ';("1) ,.(j0;j1;j2) L:0 y '
NB. ------------------------------------
newton_iteration=: 3 : 0
X=: y
COUNTER=: 0
ANS=: <''
while. COUNTER < 20 do.
V=: (- fnewx X) %. jnewx X
X=: X + V
ANS=: ANS,<X
COUNTER=. >:COUNTER
end.
;("1)}.,.ANS
)

NB. ------------------------------------
broyden=: 3 : 0
NB. Broyden method
NB. init calc
JN=: jnewx  y
DELTA=: V=. - (%.JN) +/ . * FN=: fnewx y
ANS=:(<y ), < X=.V+y
Y=:( fnewx X)-FN
NB. --------------------
COUNTER=. 0
while. COUNTER < 50 do.
JN=. DELTA broyden_sub0 Y;JN
DELTA=. -(fnewx X) %. JN
X=. X+DELTA
```

```
ANS=. ANS,<X
COUNTER=. >: COUNTER
end.
;("1),.ANS
)

broyden_sub0=: 4 : 0
NB. x  is delta
'Y JN'=: y
 JN-((Y - JN +/ . * x ) % x  +/ . * x )+/ . * x
)
```