JAPLA 研究会資料 2012/12/8

Ｊにより開立法(立方根を求める)を体験する

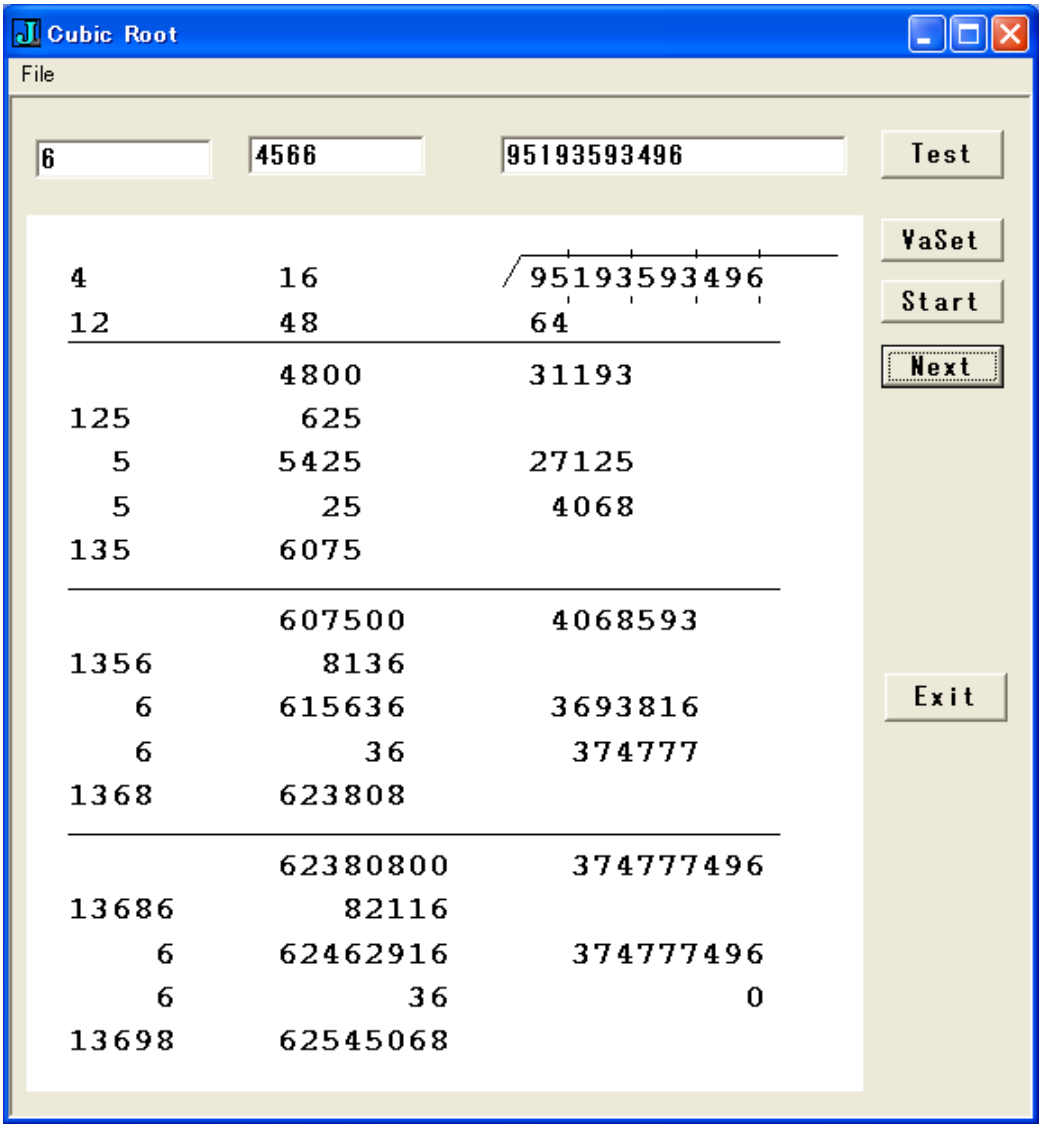
西川　利男

私と同じ世代の人なら、平方根を手計算で求める開平法を学校で習ったはずである。今では電卓、一発で求められる。同じように立方根を手計算で求める開立法がある。私自身、名前は聞いてはいたが、実際にやったことはなかった。インターネットなどで探し当てたが、その計算法はなかなかややこしい [1][2] 。

この開立法をＪの助けを借りて体験してみよう。

1．　開立法の計算とは

つぎのＪの実行画面を見ながら、計算法を説明する。



[1] http://www004.upp.so-net.jp/s_honma/root.htm

[2] 堀場芳数「対数ｅの不思議」p.64-67, 講談社ブルーバックス (1991).

例「95193593496 の立方根を求める」

① 計算は第一副演算(左欄)、第二副演算(中央欄)、主演算(右欄)で行う。
② 値を下位から 3 桁ごとに区切る。
　95 193 593 496
③ まず、最上位の 95 を越えない最大の開立数を求め、立方根の首位とする。
　そのためには、以下の開立九々を用いる。
　　$1^3 = 1$, $2^3 = 8$, $3^3 = 27$, $4^3 = 64$, $5^3 = 125$
　$6^3 = 216$, $7^3 = 343$, $8^3 = 512$, $9^3 = 729$
　今の場合は立方根の首位として N = 4 が立てられる。
④ 最初は、第一副演算、第二副演算、主演算を次のように行う。

第一副演算	第二副演算	主演算
N: 4	N*N: 4 * 4	A0: 95
4	4 * 4	64
4	4 * 4	
I: 12	S0: 48	A0X: 31

⑤ 立方根の第2位の計算の準備を行う。
　第二副演算では S0 に 2 桁の 00 を後ろに付け、
　主演算では A0X に初期値の次の 3 桁を下して付け加えて、A1 とする。
　　P: 4800　A1: 31193

⑥ そして、立方根の第2位の数は、次のようして決められる。
　N = 5 としたとき

第一副演算	第二副演算	主演算
	P: 4800	A1: 31193
J: 125	Q: 625	
N: 5	R: 5425	R*N: 27125
N: 5	N*N: 25	A1X: 4068
K: 135	S: 6075	

上の計算は、以下のように行われた。
　J = 10*I + N
　K = J + N + N
　Q = J * N
　R = P + Q
　S = Q + R + (N*N)
　A1X = A1 − R*N

ここで、N = 5 としたときは、A1X は 4068 となる。 しかし、N = 6 として、上の計算をすると、A1X は _2143 と負になってしまう。

つまり、A1X が正になる最大の数を何らかの方法で求め、それを立方根の第2位の数として立てるのである。ここが開立計算のポイントである。

⑦ 立方根の第3位の計算は、次のようになる。

⑤と同様な準備をしてから

```
I:  135      P: 607500       A2:   4068593
```

⑥と同様に

N = 6 を立てて、同じ手順で計算を行う。

```
    第一副演算   第二副演算      主演算
-----------------------------------------------
            P:  607500     A2:     4068593
   J:  1356     Q:    8136
   N:     6     R:  615636     R*N:    3693816
   N:     6     N*N:    36     A2X:     374777
   K:  1368     S:  623808
-----------------------------------------
```

⑧ 立方根の第4位の以下の計算は、さらに次のように行う。

⑤と同様な準備をしてから

```
I:  1368     P: 62380800       A2:      374777496
```

⑥と同様に

ここでも N = 6 を立てて、同じ手順で計算を行う。

```
    第一副演算   第二副演算       主演算
-------------------------------------------------
            P:  62380800   A2:      374777496
   J:  13686    Q:     82116
   N:     6    R:  62462916   R*N:     374777496
   N:     6    N*N:     36   A2X:           0
   K:  13698    S:  62545068
-----------------------------------------
```

⑨ 開立の剰余 A2X として 0 となり、開き切れた。

このように、計算操作⑤と⑥とを次々と繰り返せば、N として立てた数を並べることで、立方根の各桁が求められる。

立方根 = 4566

この計算アルゴリズムで注意すべきことは、立方根の首位を求める手順③と④と、第2位以下を求める手順⑤と⑥とは異なり、繰り返すのは手順⑤と⑥である。

2.J による直接開立法の計算

以上の各計算手順を J によりプログラム化した。第一副演算、第二副演算の部分はサブプログラム sub とし、主演算を含めた全体を main とした。

１ステップずつ、第一副演算、第二副演算の途中計算を表示する。立方根の各桁Ｎを立てて、１ステップずつ実行する。

また、立方根の各桁Ｎを立てる段階では、試行錯誤で入力できるようにした。失敗したときは、戻って別のＮを入力し、再実行を繰り返す。

もちろん、プログラムをちょっと修正すれば、自動的に行うことはごく容易である。

プログラムリストは最後にあげた。

ここでは、実行のようすを追っていただきたい。

```
  main 95193593496x
95 193 593 496
--------------
N: 4
I, S0: 12 48
95
64
--------------
31

A1: 31193
Input N:
6
J: 126
K: 138
P: 4800
Q: 756
R: 5556
S: 6348
============
31193
33336
-----------
_2143 negative !
Try again(y/n)?
y
Input N:
5
J: 125
K: 135
P: 4800
Q: 625
R: 5425
S: 6075
==============
31193
27125
-----------
4068
```

```
Try again(y/n)?

A2: 4068593
Input N:
6
J: 1356
K: 1368
P: 607500
Q: 8136
R: 615636
S: 623808
===============
4068593
3693816
-----------
374777
Try again(y/n)?

A2: 374777496
Input N:
6
J: 13686
K: 13698
P: 62380800
Q: 82116
R: 62462916
S: 62545068
==============
374777496
374777496
-----------
0
Try again(y/n)?
Cubic Root = 4566
```

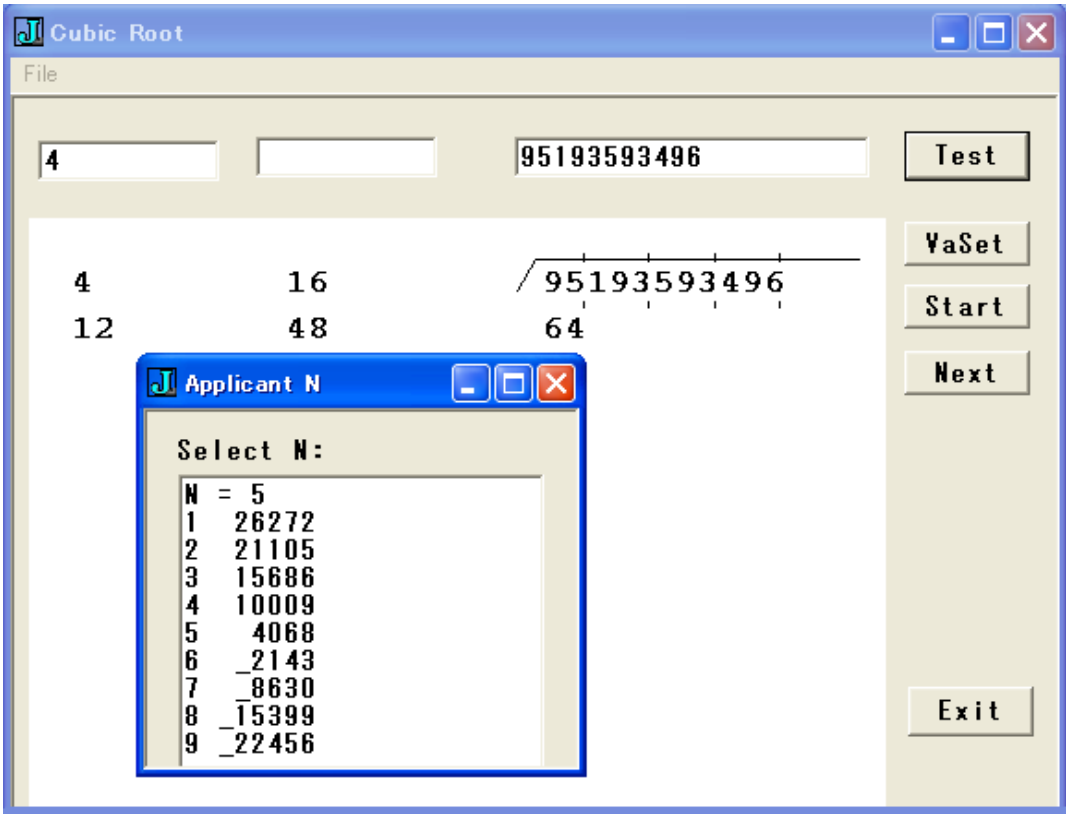
3.J-Windows 画面による開立法の計算

紙の上での実際の手計算のようすに合わせるには Windows 表示で行うことになる。実行の画面は本稿の最初のページにあげた。

第一副演算、第二副演算、主演算を横に並べて表示するには、Windows のグラフィックス画面に数値表示を貼り付けて行う。このためには gl2 グラフィックス・プログラミングで行った。

[Input EditBox]に値を入力し、[VaSet]ボタンを押すと 3 桁で区切って表示する。左端の[Input EditBox]に立方根の首位の数を入力し、[Start]ボタンを押すとその計算処理が表示される。第2位からは[Next]ボタンで計算処理が行われ、表示される。

ここで、立方根の各桁の数 N を立てるための便利なツールを作った。[Test]ボタンを押すと、次のようなポップアップ・ウィンドウが開き、候補の数値を示してくれる。
これをもとに1段階ずつ開立計算を進めていくのである。



なお、実は N を入力しないでも、[Test]-[Start]、[Test]-[Next]と次々に打ち込めば、自動的に N を選んで、最後まで開平計算を行い結果を出すようにもしてある。

プログラム・リスト1ー直接表示プログラム

```
NB. Cubic Root / Manual Calc. simulated in J Window
NB.   programmed by T.N. 2012/11/11
NB.   revised 2012/11/14, 11/16
wr =: 1!:2&2
rd =: 1!:1

VA =: 95193593496x
VB =: 48627125
VC =: 28372625
VD =: 225866529x
VE =: 248858189x

sub =: 3 : 0
'I P N' =. y.
wr 'J: ', ": J =. N + (10*I)
wr 'K: ', ": K =. N + N + J
wr 'P: ', ": P =. 100 * P
wr 'Q: ', ": Q =. x: N * J
wr 'R: ', ": R =. x: P + Q
wr 'S: ', ": S =. x: Q + R + (N*N)
K, S, (R*N)
)

CU =: 0, 1, 8, 27, 64, 125, 216, 343, 512, 729  NB. revised 11/16
NB. Select N version == 2012/11/12
main =: 3 : 0
wr A_init =. 1000 #.^:(_1) y.
wr '----------'
A0  =. {. A_init
A01 =. }. A_init
N0 =. _1 i.~ * A0 - CU
wr 'N: ', ": N0 =. <: N0
A02 =. A0 - (N0) { CU

A10 =. {. A01
A11 =. }. A01

II =. (3*N0), (3*N0*N0)
wr 'I, S0: ', ": II
wr A0
```

```
wr (N0*N0*N0)
wr '-----------'
wr A1 =. A0 - (N0*N0*N0)

A1 =. (A1 * 1000) + {. A01
wr ' '
wr 'A1: ', ": A1


NN =. ''
label_loop.
wr 'Input N:'
N =. ". rd 1
if. N = 0 do. '*** quit ***' return. end.
SUB =. sub II, N
RN =. {: SUB

wr '================'
wr A1
wr RN
wr '-----------'
A20 =. x: A1 - RN
if. 0 > A20 do. PN =. ' negative !' else. PN =. '' end.
wr (": A20), PN
wr 'Try again(y/n)?'
YN =. rd 1
if. 0 = #YN do. goto_ok. end.
if. 'y' = YN do. goto_loop. else. goto_ok. end.
label_ok. NB. N is OK!
II =. }: SUB
NN =. NN, N
wr ' '
if. 0 = # A11 do. goto_fin. end.
wr 'A2: ', ": A2 =. x: (A20 * 1000x) + {. A11
A1 =. A2
A11 =. }. A11
goto_loop.
label_fin.
'Cubic Root = ', ((":N0), (":NN)) -. ' '
)
```

プログラム・リスト2−Windows 表示プログラム

```
NB. ===============================================
NB. Windows Version
NB. usage:
NB. (1) Input Value in [Value] EditBox: e.g VA, VB, VC, 28372625, ..
NB. (2) Push [Va_Set] Button => display Value
NB. (3) Push [Test] Button => Display First Applicant & stored
NB. (4) Push [Start] Button => Calc. & Display First calc.
NB. (5) Push [Test] Button => Display Second Applcant & stored
NB. (6) Push [Next] Button => Calc. & Display Second calc.
NB. Repeat (5) and (6), then go on Calc.
NB. Finally, Calc. will reach Cubic Root

require 'gl2'

ROOT=: 0 : 0
pc root;pn "Cubic Root";
menupop "File";
menu new "&New" "" "" "";
menu open "&Open" "" "" "";
menusep ;
menu exit "&Exit" "" "" "";
menupopz;
xywh 241 8 34 12;cc Test button;
xywh 242 142 34 12;cc cancel button;cn "Exit";
xywh 4 29 232 216;cc CubicRoot isigraph;
xywh 241 30 34 11;cc VaSet button;
xywh 6 10 50 11;cc InN edit ws_border es_autohscroll;
xywh 241 61 34 11;cc Next button;
xywh 65 9 50 11;cc CbRt edit ws_border es_autohscroll;
xywh 135 9 97 11;cc InValue edit ws_border es_autohscroll;
xywh 241 45 34 11;cc Start button;
pas 6 6;pcenter;
rem form end;
)

run =: root_run
root_run=: 3 : 0
wd ROOT
NB. initialize form here
wd 'pshow;'
```

```
wd 'setfocus InValue;'
NN =: ''
)


root_close=: 3 : 0
wd'pclose'
)

root_cancel_button=: 3 : 0
root_close''
)

NB. test applicant display ==============
NB.   required tsub0, tsub, tdisplay
root_Test_button=: 3 : 0
TD =: ,. ''
select. TN
  case. 0 do. AA =. A0
  case. 1 do. AA =. A1
end.
NI =. 1
while. NI < 10
  do.
    select. TN
      case. 0 do.
         Res =. tsub0 NI
         RR =. A0 - Res
      case. 1 do.
         Res =. (II, A1) tsub NI
         RR =. A1 - ({: Res)
    end.
NB.    wr NI, RR
    TD =: TD, (NI, RR)
    NI =. NI + 1
  end.
TF =.  (* {:"(1) TD) i. _1
TEXT =: , (": TD),"(1) LF
TTEXT =: 'N = ',(":TF), LF, TEXT
wd 'setfocus InN;'
tdisplay ''
N =: TF
```

```
)

tdisplay =: 3 : 0
wd 'pc testapl;pn "Applicant N";'
wd 'xywh 8 6 80 8;cc s0 static;cn "Select N:";'
wd 'xywh 8 15 100 100;cc e0 editm ws_border;'
wd 'set e0 *',TTEXT
wd 'pas 8 8;pcenter;pshow;wait;'
wd 'pclose;'
)
tsub0 =: 3 : 0
N =. y.
NB. ((<:N){CU)
N{CU
)

tsub =: 3 : 0
:
N =. y.
'I P A' =. x.
NB. wr I, P, A
J =. N + (10*I)
K =. N + N + J
P =. 100 * P
Q =. N * J
R =. P + Q
S =. Q + R + (N*N)
K, S, (R*N)
)
NB. ===========================================

root_VaSet_button=: 3 : 0
glfont '"ISIJ" 42 bold'
GN =: 0
gmain Value
NB. gmain VA
TN =: 0
NB. auto first digit =====
NB. gmain Value
NB. gmain VA
NB. gmain VB
)
```

```
root_Start_button=: 3 : 0
wd 'set InN *', (": N)
gstart ''
TN =: 1
)


root_Next_button=: 3 : 0
wd 'set InN *', (": N)
GN =: GN + 1
gnext ''
)


gsub =: 3 : 0
'I P N' =. y.
J =. N + (10*I)
K =. N + N + J
P =. 100 * P
Q =. N * J
R =. P + Q
S =. Q + R + (N*N)
(50, (1070-GN*280)) gwr , J, N, N, K
(300, (1120-GN*280)) gwr P, Q, R, (N*N), S
K, S, (R*N)
)

gmain =: 3 : 0
V0 =: x: y.
A_init =. 1000 #.^:(_1) V0
A0  =: {. A_init
NB. initial value display
gllines 590 955 970 955
gllines 570 920 590 955

NB. gllines 877 960 877 930
NB. gllines 807 960 807 930
NB. gllines 732 960 732 930
NB. gllines 650 960 650 930

NB. 3-keta gotoni kugiru ==== 11/17 ===========
```

```
NB. From X = Kmx upto Left, Set Kugiri Lines
Km =. # ": V0
Kmm =. >. Km % 3
Kmx =. 600 + 25 * Km
NB. Kmx =. 880
Kx =: ,. Kmx - 76 * (i. Kmm)
KK =. Kx,"(1) 960,"(1) Kx,"(1) 900
KKK =. <"(1) KK
gllines L:0 KKK

NB. Display Initial Value
600 950 gwr0 V0
)

gstart =: 3 : 0
A_init =. 1000 #.^:(_1) V0
A0  =. {. A_init
A01 =: }. A_init
NB. first digit applicant ==================
NB. N0 =: _1 i.~ * A0 - CU  NB. calc. from CU table
N0 =: N NB. input from editbox InN
50 950 gwr0 N0
50 900 gwr0 (3*N0)
300 950 gwr0 (N0*N0)
300 900 gwr0 (3*N0*N0)

NB. N =: N0

RN0 =. N0 { CU         NB. revised
A02 =: A0 - (N0 { CU)  NB. revised
Ketma =: #": V0
Ketm =: 3 | Ketma
A10 =. {. A01
A11 =: }. A01

A1 =: (A02 * 1000) + A10
II =: (3*N0), (3*N0*N0)

600 900 gwr0 RN0

)
```

```
gnext =: 3 : 0  NB. revised 11/17 ================
NB. wr 'GN = ', ": GN
gllines 50, (855-(GN-1)*280), 900, (855-(GN-1)*280)
Ket0 =. (# ": A0)
Ket =. (3*GN) + Ket0
NN =: NN, N  NB. from editdox InN
SUB =. gsub II, N
RN =. {: SUB
II =: }: SUB

A20 =. x: A1 - RN

(600, (1120-GN*280)) gwr0k A1, Ket
(600, (1020-GN*280)) gwr0k RN, Ket
(600,  (970-GN*280)) gwr0k A20, Ket

if. 0 = # A11 do. goto_fin. end.

A1 =: x: (A20 * 1000x) + {. A11
A11 =: }. A11
label_fin.
NB. 'Cubic Root = ', ((":N0), (":NN)) -. ' '
CuR =. ((":N0), (":NN)) -. ' '
wd 'set CbRt *', CuR
)
gwr =: 3 : 0
:
DA =. y.
'GX GY' =. x.
I =. 0
while. I < #DA
  do.
    gltextxy GX, (GY-I*50)
    Keta =. >./ >. 10 ^. DA
    gltext  Keta ": I{DA
    I =. I + 1
  end.
glshow ''
)

gwr0 =: 3 : 0
:
```

```
DA =. y.
'GX GY' =. x.
I =. 0
while. I < #DA
  do.
    gltextxy GX, (GY-I*50)
    if. 0 > I{DA
      do.
        glrgb 255 0 0
        gltextcolor ''
      else.
        glrgb 0 0 0
        gltextcolor ''
      end.
    gltext  '': I{DA
    I =. I + 1
  end.
glshow ''
)

gwr0k =: 3 : 0
:
'DA Keta' =. y.
'GX GY' =. x.
I =. 0
while. I < #DA
  do.
    gltextxy GX, (GY-I*50)
    if. 0 > I{DA
      do.
        glrgb 255 0 0
        gltextcolor ''
      else.
        glrgb 0 0 0
        gltextcolor ''
      end.
    gltext Keta '': I{DA
    I =. I + 1
  end.
glshow ''
)
```

```
root_InN_button=: 3 : 0
N =: ". InN
)

root_InValue_button=: 3 : 0
Value =: ". InValue
)
```