解けない方程式を解いてみる

A trial for solving equations including complex numbers

作花 一志 （京都情報大学院大学）

Kazushi Sakka（The Kyoto College of Graduate Studies for Informatics）

Abstract

本稿は実数解をもたない複素数を未知数とする方程式を解く一つの試みである。与えられた方程式を漸化式の形に変形し 50 回の繰り返し演算の結果を複素平面上で表現した。

This article is an attempt to solve an equation that makes a complex number unknown without a real number solution. The given equation was transformed into a recurrence formula ,and the result of 50 iterative operations was expressed on the complex plane.

1. グラフ描画

方程式を解くための最初の段階はグラフを描くことであり，その最も簡単な方法はブラウザ（GoogleChrome など）で関数名を検索することである。Wolfram Alpha でも容易に描ける。R 言語で y ＝ sin(x) を－ 5 ≦ x ≦ 5 の範囲で描くには

x 軸 y 軸付きで

```
curve(sin(x), － 5,5)
abline(v ＝ 0,col ＝ 8)
abline(h ＝ 0,col ＝ 8)
```

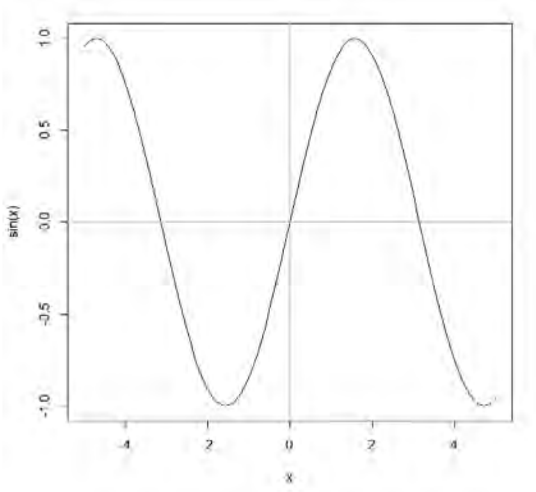

図１ y ＝ sinx

この 3 行だけで図 1 が得られる。col ＝の後の数はカラーコードで 1（黒：通常省略), 2（赤), 3（緑), 4（青)，5（シアン)，6（マゼンタ)，7（黄)，8（グレー）である。

一般に関数 f(x) を定義してグラフを描くには

```
f ＝ function(x) sin(x)
curve(f(x), － 5, 5)
```

とすればよい。第 1 行は関数定義である。＝の代わりに＜－を使うことも多い。

またここに別の関数 cos(x) のグラフを重ねて描く時には add ＝ T を付ける。これを付けないと前に描いたグラフは消えてしまう。

```
f ＝ function(x) sin(x)
g ＝ function(x) cos(x)
curve(f(x), － 5, 5)
curve(g(x), － 5, 5, lty ＝ 2,add ＝ T)
abline(v ＝ 0, col ＝ 8)
abline(h ＝ 0, col ＝ 8)
```

第 4 行の後 lty ＝に続く数は線のスタイルで

1（実線 通常省略）2（破線）3（点線）

を表す。

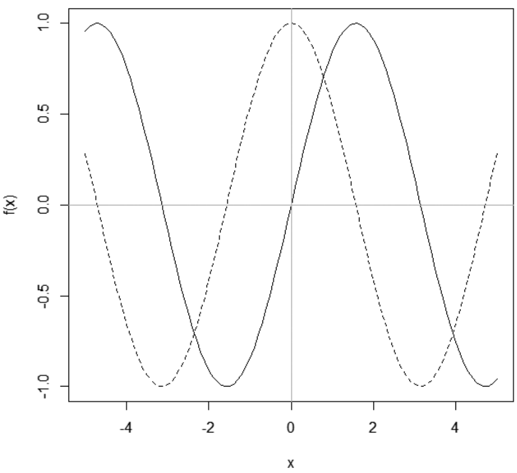

図２　y ＝ sinx　y ＝ cosx

方程式解法

方程式を解くという学習は中学校以来学んでいるが，公式によって解ける方程式は１次方程式と２次方程式くらいなもので，きわめて例外である。

整方程式をＲで解くには係数をベクトルで与えて polyroot 関数を使えばよい。$x^2 + 2x - 3 = 0$ の解は

polyroot(c(− 3,2,1)) より求まる。( ) 内は昇べき順の係数である。

1 + 0i　− 3 + 0i　　これは x ＝ 1, − 3 を表す。

また $2x^3 + x^2 - 2x + 3 = 0$ の解は polyroot(c(3, − 2,1,2)) より

[1] 0.5767283 + 0.7580072i　−1.6534566 + 0.0000000i 0.5767283 − 0.7580072i

と求まるが２番目の値は実数で他は複素数である。

分数方程式，無理方程式は整方程式に変形してから解けばよい

次に非線形方程式　$x^2 - 2^x = 0$ を解いてみよう。2^x を y と置き換えて…などとしてもうまくいかない。置換が悪いのではなく，そもそもいわゆる数学的解法が無理なのだ。受験技術に長けた大学生よりむしろ直観力のある小中学生のほうが早く解くことができる。x ＝ 2 としてみると $2^2 - 2^2 = 0$ だから解であることはすぐにわかり，また x ＝ 4 も明らかに解である。しかし解はこの２つだけだろうか。これ以上は直観でも解析でもわからない。

方程式 f(x) ＝ 0 の解とは y ＝ f(x) のグラフを描いて x 軸との交点の x 座標のことである。すなわち y ＝ f(x) の逆関数 $x = f^{-1}(y)$ において y ＝ 0 となる $x = f^{-1}(0)$ を求めることである。

```
# 方程式 f(x)=0　の f(x) を与える
f = function(x) x^2 − 2^x
curve(f(x), − 6,6)              # f(x) プロット
abline(h = 0, col = 4)          # 青で X 軸を描く
abline(v = 0, col = 4)          # 青で y 軸を描く
x1=    ;x2=
ans = uniroot(f, c(x1, x2))
text ( − 1, − 2, ans$root, col=2)
title(main="Equation    x^2=2^x")

# 整方程式 x² + 2x − 3=0 の解
polyroot(c( − 3, 2, 1))
```

グラフを描いてみるともう一つ負の解があるが，これは数値的にしか求まらない。その解法は昔から多数開発されているが，詳しくは数値計算の本を参照されたい。R では uniroot という便利な関数が装備されている。x 軸との交点の左右に適当な数 x1 と x2 を取り

ans ＝ uniroot(f, c(x1, x2)) とするだけでよい。

解は ans$root に収まっていて, 数値 − 0.7666825 はコンソールとグラフィック画面に表示される。なお四捨五入して小数点第３位まで表すには round(ans$root,3) とする。

これにより関数を変更すればどんな方程式でも解ける。ただしもちろん実根だけであるが。

でも x1, x2 にどんな値を与えるか？グラフを描いてみてから x1 ＝ − 2, x2 ＝ 0 と x 軸との交点の左右の点を決めてもいいが，もっと素晴らしい方法がある。locator を使えばグラフと x 軸との交点を挟む２点をクリックするとその２点の x 座標が x1, x2 となるのだ。

```
z = locator(2); # 曲線と x 軸との交点を挟む 2
点をクリック
points(z, col = 4)
x1 = z$x[1] ; x2 = z$x[2]      # クリックした
点の x 座標
ans = uniroot(f, c(x1, x2))     # その間の解
```

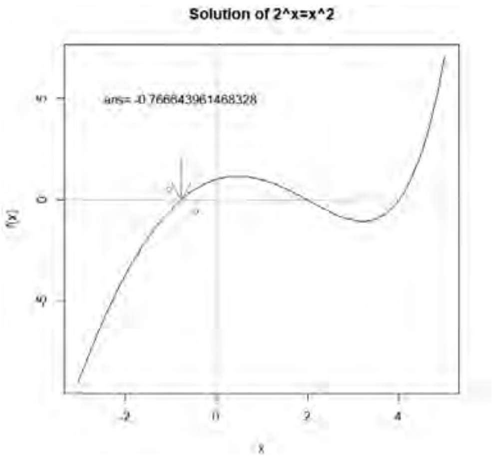

図３　$2^x = x^2$ の解

解けないものを解く

cosx ＝ 2

そんなもの解けるはずがない。－ 1 ≦ cosx ≦ 1 だから解なしだ。でも　$x^2 = -1$ は高校１年生でも ± i と解くのだから，この方程式も複素数の範囲で考えてみよう。

・Euler の公式より

$\cos(x) = (\exp(ix) + \exp(-ix))/2$

$\exp(ix) = y$ とすると次の２次方程式が得られ

$y^2 - 4y + 1 = 0$

よって　$y = 2 \pm \sqrt{3}$

$\exp(ix) = 2 \pm \sqrt{3}$ だから　$x = i\log(2 \pm \sqrt{3})$

・Wolfram Alpha を使って

逆関数を使えば x ＝ arccos(2) であるから

x ＝ acos(2) と入力すれば　図４

$x = i\log(2 + \sqrt{3}) = 1.31696i - 1.31696i$

解は純虚数である

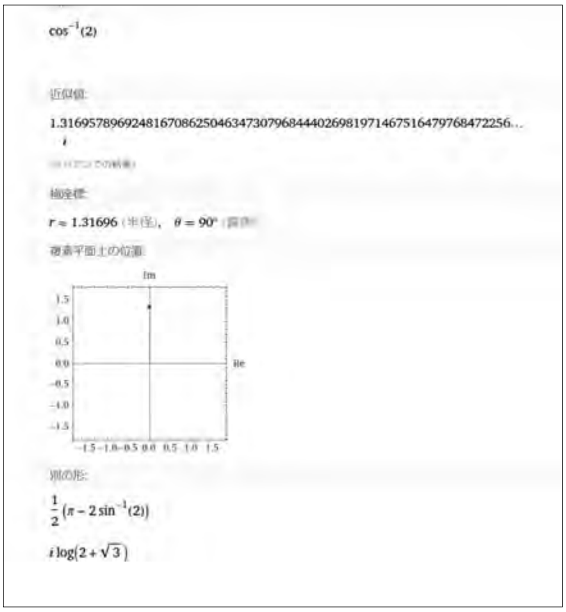

図４　cosx ＝ 2 の解

・gnuplot にて

print acos(2)

{0.0, － 1.31695789692482}　が得られる。

gnuplot では複素数を　{}　内のコンマの前が実部でコンマの後が虚部で表す。

∴　x ＝ － 1.31695789692482i

x ＝ log(x)

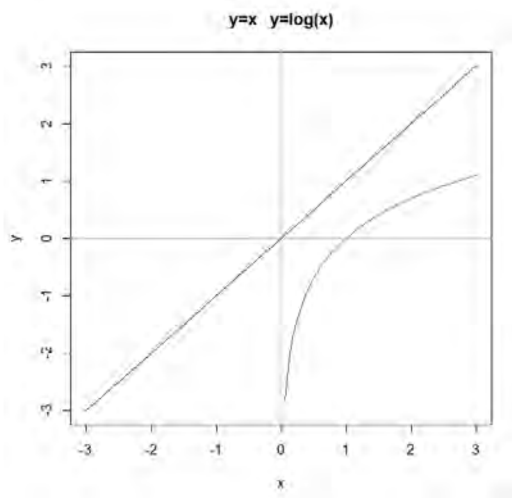

図５　y ＝ x　y ＝ log(x)

y = x と y = log(x) は交点を持たないからこの方程式は実解なしであるが

・Wolfram Alpha で y = x − log(x) のグラフを描く。ただし log(|x|) としてある

虚部では x < 0 で − π，x > 0 では 0 であり

実部では　x = − 0.6 あたりで x 軸と交点がありそうである。

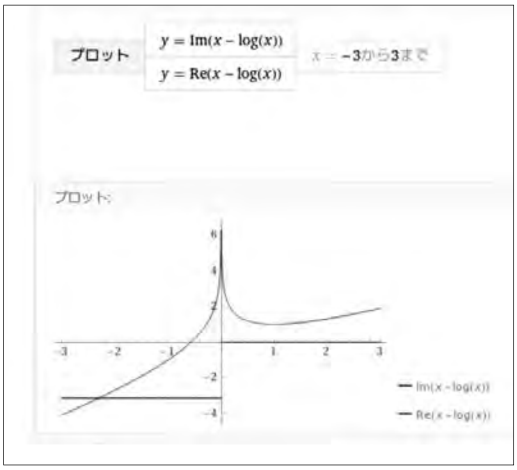

図 6　y = x-log(x)

log(abs(x)) = 0」と入力すると下図が得られ x = − 0.567 で x − log(x) の値は 0 − i π になる。

∴　x = − 0.567

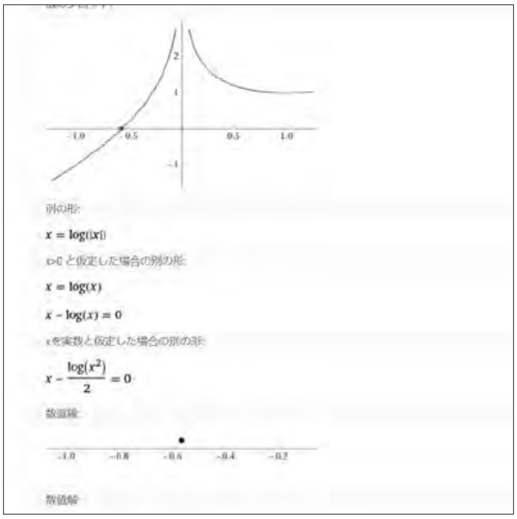

図 7　x-log(x) = 0 の解

では実部も虚部も 0 になるような複素数はどうやって求めるか？

複素平面上でそのような数を試行錯誤で探していくしかない。それにはある適当な初期値 z[1] を与えて漸化式 z[k + 1] = log(z[k]) を何回か繰り返し計算する。Z[1] = 1 + 2i として Excel で計算したところ 25 回で収束した。結果は表 1 に載せた。

下記プログラムでは z[1] としてマウスクリックした点の x 座標，y 座標をそれぞれ実部，虚部とする。

```
########## z=log(z) ###########
plot(0,0,xlim=c(-5,5), ylim=c(-5,5),type="n",
xlab="real",ylab="imag")            # 座標軸取り
abline(h=0,col=8);abline(v=0,col=8)
w=locator(1);points(w,col=2)
 (u=w$x) ;(v=w$y)
 z[1]=complex(re=u,im=v)            # 複素数形成
for (k in 1:50)
z[k + 1]=log(z[k])                  # z=log(z) 計算
paste(k+1," ",z[k+1]," ",z[k],"****")# z(k) 出力
k=1:50
 points(z[k],col=6,type="l")        # z(k) プロット
px=Re(z[50]);rx=round(px,3)
py=Im(z[50]);ry=round(py,3)
text(-1,-4,paste("ans = ",rx," + ",ry," i"),col=4)
title("x=log(x)")
```

繰り返す回数を 50 とし，z[k] を順に線で結んでいった結果が図 8 で巻き込んだ点に収束した。

このようにして

z = 0.318 + 1.337i

という解が得られた。なおこれと共役な複素数も解となる事は z = 0.318 − 1.337i において z − log(z) の実部も虚部 0 であることで確かめられる。

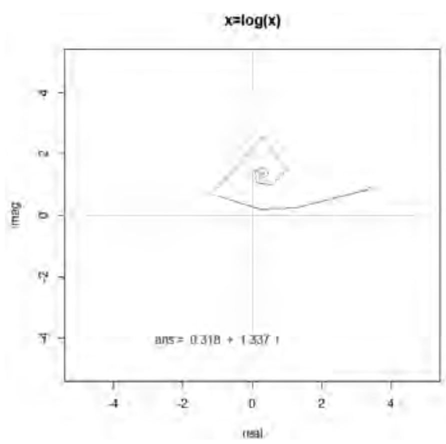

図 8　z[k + 1] = log(z[k])

ところが与方程式は z[k + 1] = exp(z[k]) という漸化式でも変形できる。この場合は z は発散する。

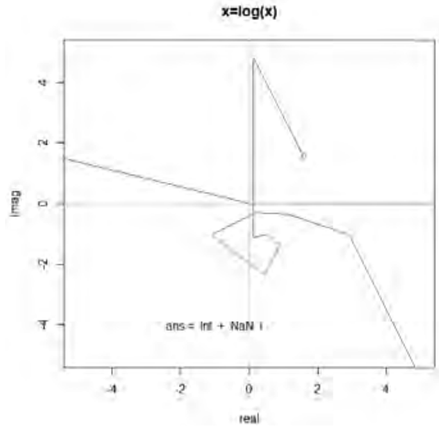

図 9　z[k ＋ 1] ＝ exp(z[k])

同様の方法で次の方程式が解けた。

① z = log(z − 2)
　z[k + 1] = log(z[k] − 2))　　z = 0.861 ± 2.073i
　z[k + 1] = exp(z[k]) + 2　　発散
② $z^2 = 2^z$
　z[k + 1] = sqrt(2^z[k])　　z = 2
　z[k + 1] = − sqrt(2^z[k])　　z = − 0.7067
　z[k + 1] = 2*log(z[k])/log(2)　z = 4
③ z^2 = log(z)
　z[k + 1] = sqrt(log(z[k]))　　z = 0.614 ± 0.681i
　z[k + 1] = − sqrt(log(z[k]))　z = −1.213−1.008i
　z[k + 1] = exp(z[k]^2)　　発散
④ zlog(z) = 1
　z[k + 1] = 1/log((z[k]))　　z = 0.170 ± 0.4545i
　z[k + 1] = exp(1/z[k])　　z = 1.763

② $z^2 = 2^z$ を解いてみよう。このグラフは図 10 のようになり実は 3 個の実数解が存在する。

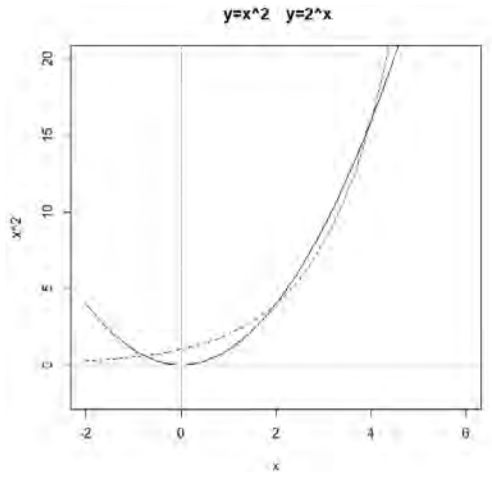

図 10　$y = x^2$　$y = 2^x$

与方程式を z[k + 1] = sqrt(2^z[k])
という漸化式に変形し前述のを実行すると z = 2 を得る。（図 11a）

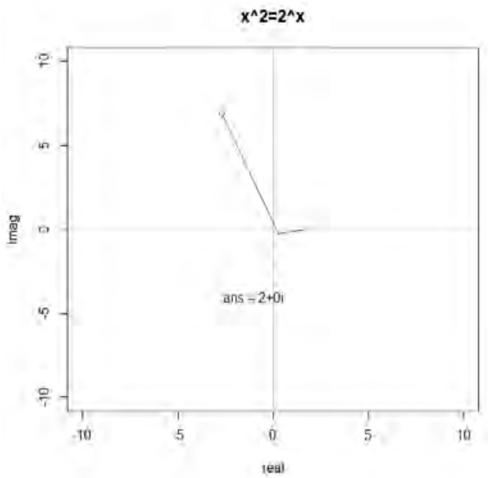

図 11a　z[k ＋ 1] ＝ sqrt(2^z[k])

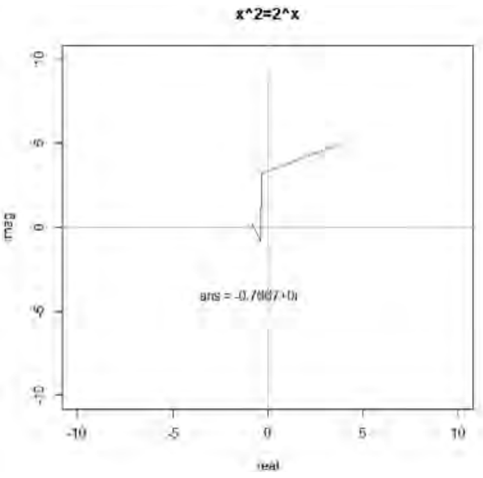

図 11b　z[k ＋ 1] ＝ sqrt(2^z[k])

また z[k + 1] = − sqrt(2^z[k])
という漸化式に変形し前述の方法を実行すると
z = 0.7067 を得る。図 11b
さらに z[k + 1] = 2*log(z[k])/log(2) という漸化式に変形し前述の方法を実行すると z = 4 を得る。（図 11c）

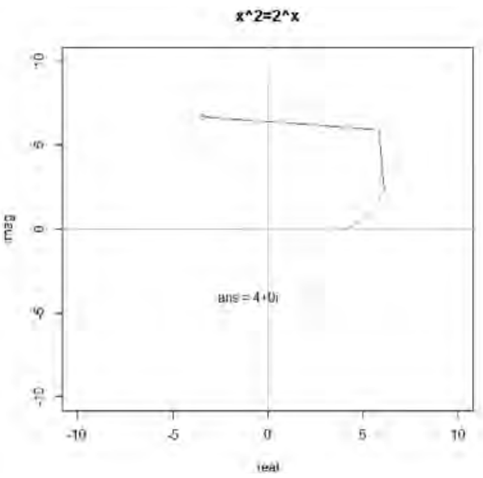

図 11c　z[k ＋ 1] ＝ 2log(z[k])/log2

なお③は最初の漸化式と 3 番目の漸化式による解は原式を満たさない。

このように変形する漸化式によって異なる値に収束したり発散したりすることもありる。これらの漸化式で得られた解がユニークな解であるともいえず，さらに考察を進めていく必要がある。

本研究は「国際化を考慮した社会的ネットワーキング指向の次世代 e ラーニング基盤の開発（日本学術振興会研究費番号 16H03087，研究者代表・岡本敏雄氏）」のサポートの下で行われている。

◆著者紹介

作花　一志　Kazushi Sakka

京都情報大学院大学教授
京都大学理学研究科修了　理学博士
元京都コンピュータ学院鴨川校校長
元京都大学理学部・総合人間学部講師
元国際日本文化研究センター研究員
元日本天文教育普及研究会編集委員長

	初期値　実部＝ 1.0　虚部＝ 2.0	
	z1	**z2 = log(z1)**
1	1 + 2i	0.80471895621705 + 1.10714871779409i
2	0.80471895621705 + 1.10714871779409i	0.313864371353433 + 0.942283727201657i
3	0.313864371353433 + 0.942283727201657i	-0.0068418652546109 + 1.24926566277629i
4	-0.0068418652546109 + 1.24926566277629i	0.222570905856397 + 1.57627298164417i
5	0.222570905856397 + 1.57627298164417i	0.464933936134835 + 1.43052292556221i
6	0.464933936134835 + 1.43052292556221i	0.408248326417302 + 1.25655558534005i
7	0.408248326417302 + 1.25655558534005i	0.278548777085706 + 1.25665961584999i
8	0.278548777085706 + 1.25665961584999i	0.252438828211082 + 1.35266502472157i
9	0.252438828211082 + 1.35266502472157i	0.319194479618421 + 1.38629542388733i
10	0.319194479618421 + 1.38629542388733i	0.352463788729392 + 1.34449054090608i
11	0.352463788729392 + 1.34449054090608i	0.329248244075426 + 1.31441156926825i
12	0.329248244075426 + 1.31441156926825i	0.303817025678083 + 1.32535561684757i
13	0.303817025678083 + 1.32535561684757i	0.307287941803104 + 1.34545524788161i
14	0.307287941803104 + 1.34545524788161i	0.322155879781029 + 1.34625799696145i
15	0.322155879781029 + 1.34625799696145i	0.325170725822681 + 1.33591589769881i
16	0.325170725822681 + 1.33591589769881i	0.318396150720572 + 1.3320327934074i
17	0.318396150720572 + 1.3320327934074i	0.314487621846114 + 1.33616846864771i
18	0.314487621846114 + 1.33616846864771i	0.316764571223859 + 1.33963826255667i
19	0.316764571223859 + 1.33963826255667i	0.319601627145393 + 1.33860567414545i
20	0.319601627145393 + 1.33860567414545i	0.319348229932126 + 1.33642687214758i
21	0.319348229932126 + 1.33642687214758i	0.3177643507187 + 1.33623800525547i
22	0.3177643507187 + 1.33623800525547i	0.317363270621038 + 1.33732763868911i
23	0.317363270621038 + 1.33732763868911i	0.318067375933872 + 1.33779493862944i
24	0.318067375933872 + 1.33779493862944i	0.318516445454639 + 1.33737519781771i
25	0.318516445454639 + 1.33737519781771i	0.318295066809781 + 1.33698678696656i