函数論講義メモ No.7; 計算機演習

金子　晃

【準備】 計算機に自分のアカウントとパスワードでログインする．パスワードを失った人は早めに申し出ること．間に合わなかった人は友達のを見せてもらう．どうしても自分でやりたい場合は小生のアカウントを臨時に使う．
本日，小生のアカウントで入った人は，【準備】の以下の部分をやってはいけない．代わりに，ターミナルエミュレータのウインドウを一つ開いて **tcsh** を立ち上げ，`cd Kansuron` を実行して直ちに課題１以下に進むこと．ここに書かれたその他の処理は，後で自分のアカウントを取得し，それでログインしたときに一人で実行する．あるいは，情報の３年生は５階の機械で `~kanenko/Func/EnshuY19` から必要なものをコピーして同じことをすればよい．

以下を順次実行する．

1) 質問をしたい人は，ド近眼の小生のために，まず画面のデザインアイコンをクリックして，画面の設定ツールを開き，漢字フォントを最大の 7 に変更しておく．

2) ターミナルエミュレータのウインドウを開く．以下の指令はこのウインドウで打ち込む．

3) `cp ~kanenko/Func/.tcshrc .` を実行した後，tcsh を立ち上げる．

4) `mkdir Kansuron` を実行して，`cd Kansuron` する．

5) `cp ~kanenko/Func/EnshuY19/* .` を実行する．

6) `make` を実行する．

これで本日のデモプログラムがコンパイルされます．

【課題１ :複素線積分】 integral を実行する．
これは $\oint_{|z|=1} z^{-m}dz$ の数値計算をするプログラムです．講義で学んだ手計算の結果と比較してみよう．なお，ソースは Fortran で書かれており，被積分函数と積分路の方程式を取り替えれば，別の問題に使える．(後述の Mathematica のように数式処理ソフトではないので，m の値を何か指定しなければ計算してはくれない.) ソースを自分で直した後で，それをコンパイルする仕方がわからない人は，make を再実行すればよい．(どういうオプションでコンパイルしているかを知りたければ Makefile をエディターで読めば良い.)

【課題２ :等角写像】 confmap を実行する．
これは平面の写像による直交格子と円周の像のゆがみ具合を表示するものである．ソースは C で書かれており，写像の例は始めの二つが一般の写像，後の三つが等角写像です．特に，最後のものは一次変換で円円対応が成立しています．挙動の違いを味わって下さい．

ソースで写像の定義式の選択肢を追加すれば，別の写像の様子を調べることができます．また，パラメータ `XMIN`，`XMAX`，`YMIN`，`YMAX` の値を書き直せば原領域の正方形を別の領域に取り替えることもできます．

【課題３ :Mandelbrot 集合】 mandelbrot を実行する．
複素力学系 $z \mapsto z^2 + c$ を原点を初期値として反復適用したとき何回で無限遠に飛び去るかを $c \in \boldsymbol{C}$ の値で分類し色分けしたときに得られる図形です．ソースは Fortran で書かれており，グラフィックの部分は C で別ライブラリとなっています．

【課題４ :Julia 集合】 julia を実行する．
複素力学系 $z \mapsto z^2 + c$ を初期値 z_0 に対して反復適用したとき何回で無限遠に飛び去るかを $z_0 \in \boldsymbol{C}$ の値で分類し色分けしたときに得られる図形です．Mandelbrot と違い，c は固定されていますが，これを

取り替えることでいろんな Julia 集合が得られます. ソースは Fortran.

【課題5 :Newton 法の収束先】 cubic を実行する.
これは代数方程式 $z^3-1=0$ に対する Newton 法の反復公式が, 初期値が z_0 のとき, どの根に収束するかを, 三つの根に従って平面を三色で色分けしたものです. 黒いところは本当は無限遠に飛ぶところであるべきですが, 実際には $\{|\mathrm{Re}\, z| \ge 10000\} \cup \{|\mathrm{Im}\, z| \ge 10000\}$ の逆像です. ソースは Fortran.

【課題6 :Mathematica の使用法】 math を実行する.
Mathematica が立ち上がるので, 以下の計算を順にやってみよ.

1) 正則函数 $\log z = \log\sqrt{x^2+y^2} + i\,\mathrm{Arctan}\,\dfrac{y}{x}$ に対する Cauchy-Riemann の関係式の確認

```
F=Log[Sqrt[x^2+y^2]]+I* ArcTan[y/x]
G=D[F,x]+I*D[F,y]
Simplify[G]
```

2) 複素線積分 $\displaystyle\oint_{|z|=1}\frac{1}{z^m}dz$ の計算

```
z=Exp[I*t];
Integrate[D[z,t]/z^m,{t,0,2*Pi}]
Integrate[D[z,t]/z,{t,0,2*Pi}]
```

同じく, 複素線積分 $\displaystyle\oint_{\max\{|x|,|y|\}=1}\frac{1}{\bar{z}^m}dz$ の計算

```
z=1+I*y; bz=1-I*y; I1=Integrate[D[z,y]/bz^m,{y,-1,1}]
z=x+I; bz=x-I; I2=Integrate[D[z,x]/bz^m,{x,1,-1}]
z=-1+I*y; bz=-1-I*y; I3=Integrate[D[z,y]/bz^m,{y,1,-1}]
z=x-I; bz=x+I; I4=Integrate[D[z,x]/bz^m,{x,-1,1}]
f[m_]:=I1+I2+I3+I4
```

最後の答を数学の普通の答に変換するのは結構大変である. `f[2]`, `f[3]`, `f[4]`, `f[5]`, `f[6]`, `f[7]` を順に実行してみると結果の予想がつくだろう. $m=1$ の場合は別にやってみないと信用できないことは円周のときと同様である. `bz` は $\bar{z}$ のつもり. 複素共役を表す `Conjugate[z]` という函数は存在するが, 数値のときは正しい答を返すのに, 実数宣言された変数も変換してくれない:

```
y/:Head[y]=Real
Conjugate[y]
```

同様に m を整数宣言しても, `Conjugate[(-1)^m]` は動かない. だからといって, `Integer` が複素整数をも考慮に入れていると云う訳でもない:

```
m/:IntegerQ[m]=True
Conjugate[m]
m=1+I
IntegerQ[m]
Conjugate[m]
```

3) 等角写像 $w=z^2-1$ による正方形 $\max\{|x|,|y|\}<1$ の像の表示

```
z=x+I*y;
points=Table[N[z^2-1],{x,-1,1,0.1},{y,-1,1,0.1}];
coords=Map[{Re[#],Im[#]}&,points,{2}];
lines=Map[Line,Join[coords,Transpose[coords]]];
Show[Graphics[lines],AspectRatio->1,Axes->Automatic]
```

同じく, 等角写像 $w=(z-1)/(z+2)$ による単位円 $|z|<1$ の像の表示

```
f[z_]:=(z-1)/(z+2);
points=Table[N[f[r*Exp[I*t]]],{r,0,1,0.1},{t,0,2*Pi,Pi/16}];
coords=Map[{Re[#],Im[#]}&,points,{2}];
lines=Map[Line,Join[coords,Transpose[coords]]];
Show[Graphics[lines],AspectRatio->1,Axes->Automatic]
```

補足 Mathematica の主な特徴 (一部は数式処理に共通) :

1) 函数の引数は ( ) でなく [ ] で括る.

2) 函数名は頭文字が大文字になっている.

3) 一指令ごとに結果が出力されるが, 指令の末尾に ; を付けることでそれが抑制される. 最初はやたらに ; を使わず, 途中結果を見ながら進む方が安全である.

4) 必ずしも自動的には一番簡単な形にまで持って行ってくれず, Simplify 指令が必要となることがある.

5) 計算結果は必ずしも人間がやったときの一番自然な表現とは一致しない. うまく利用するには頭も使う必要がある.

6) 数値は有理数でどこまでも正確に計算しようとするので, 近似数値計算をするには `N` 指令を付ける必要がある.

【課題７ :TeX の使用法】 とりあえずは, 函数論の講義メモが読めるかどうか実験してみてください.

```
ghostview ~kanenko/Func/TeX/func1.ps
```

ソースの書き方に興味のある人は, 同じドキュメントのソース`~kanenko/Func/TeX/func1.tex` を自分のところにコピーして来て

```
platex func1
```

を実行すると, `func1.dvi` が出来,

```
xdvi func1
```

によりイメージが確認できます.

```
dvips <func1.dvi >func1.ps
```

により自分で PS ファイルを作ることも出来ます. PS ファイルは lpr 指令で印刷できます.

メモによっては画像ファイルを取り込むものがあります. エラーメッセージで止まったら, ctrl+d で取りあえず修了させてから`fcfig*.eps` の形のファイルを `~kanenko/Func` からコピーしてやり直してください.

より詳しい TeX の用法は, 函数論のホームページの report.html を参照のこと.

【付録 :mule の使用法 (最小限)】 ソースを直すときに使います. 例えば, `mule integral.f` でファイルを読み込み (指定ファイルが存在しなければ新規作成モードとなる), 普通の常識を働かせてキー操

作をして編集し終えたら, ctrl+x ctrl+s でセーヴし, ctrl+x ctrl+c で終了する. 普通の状態でカーソルの前を消去するのは del キーの方であり. bs キーはヘルプ画面が開いてしまうように設定されていることが多いので注意.

漢字の入力は canna のシステムなら ctrl+o で, Wnn なら ctrl+ で辞書が開閉する. 漢字変換中にカーソルの前を消去するのは ctrl+h が普通で, del や bs は利かない. 漢字変換中でないときに ctrl+h をやるとヘルプ画面が開いてしまう.

間違ったキー操作で困ったときは ctrl+g を何回か押してみる. どうしても直らなければ, 一度終了してやり直す (その際にセーヴするかどうか聞いて来るので, 作りかけのファイルについては y と答えるのを忘れないように).

【付録 :C と Fortran の違い】 C では函数の引数は原則として値渡しなのに対し, Fortran では原則としてアドレス渡しであるので, プログラムを移植するときは, `&` や `*` を使う必要が出てくる. また, 値渡しの引数はサブルーチンの中で値を変更してもメインに影響を与えないのに対し, アドレス渡しの引数の場合は, サブルーチン内で臨時に変えたつもりでもメインの対応変数の値が変わってしまう.

Fortran には大域変数の概念は無いので, ある変数を共通で使いたいときは common 宣言が必要.

Fortran には複素数型の変数のカテゴリーがあるが, 多くの処理系でまだ単精度しかサポートしていない. 上のプログラム見本では, integral.f が分かりやすさのため複素数型を使っているが, 誤差のオーダーを見るのにやや不満なので, intdbl.f では複素数型を使わずに実部 ·虚部をそれぞれ独立に倍精度実数として扱っている. この他, mandelbrot.f, julia.f は複素数型を, cubic.f は倍精度実数のペアを用いている.

演習問題 1(講義メモ No.5 より再録) 次の多項式は各象限内にそれぞれ何個の複素根を持つか ? 偏角原理を用いて適当な閉曲線 C に関する線積分を数値計算することにより明らかにせよ. (計算は `integral.f` を修正して用いるか, Mathematica を用いよ.)

1) $z^3 - z + 5$ 2) $z^4 + 2z^2 + 1$ 3) $z^5 - z + 2$ 4) $z^6 + z^5 - 1$

演習問題 2(講義メモ No.5 より再録) 上の問題で挙げた各多項式の各々について, プログラム例 `cubic.f` を別名でコピーしたものを適当に修正することにより, すべての根を Newton 法で近似計算せよ. また, Newton 法による反復がどの根に収束するかを初期値に対して色分けした図を描け.

演習問題 3(講義メモ No.5 より再録) $\sin z$ の Taylor 展開の部分和として得られる多項式

$$z - \frac{z^3}{3!} + \frac{z^5}{5!} - + \cdots + (-1)^n \frac{z^{2n+1}}{(2n+1)!}$$

の根は始めの $1/3$ 個くらいは $0, \pi, 2\pi, \ldots$ に近い実根となる. では残りの根はどこにあるか ? プログラム例 `cubic.f` を参考にして研究せよ. (打ち切る階数 n も入力できるようにするのが望ましい.)

演習問題 4(講義メモ No.6 より再録) 次の正則函数による, 指定された領域の像を描け. プログラム例 `confmap.c` を参考にするか, Mathematica を使え.

1) $w = z^2$, 円板 $|z-1| \le 1$ の像 2) $w = z^3$, 円板 $|z-1| \le 1$ の像
3) $w = e^z$, 長方形 $0 \le x \le 1$, $0 \le y \le 2\pi i$ の像 4) $w = z^3 - 3$, 長方形 $-2 \le x \le 2$, $0 \le y \le 1$ の像

演習問題 5(追加) 次の函数が正則かどうかを Cauchy-Riemann の方程式を使って Mathematica により判定せよ.

1) $x^2 - y^2 + 2xyi$ 2) $x^2 - y^2 - 2xyi$ 3) $e^{x^2-y^2}(\cos 2xy + i \sin 2xy)$
4) $\dfrac{x^2 - y^2}{(x^2+y^2)^2} - \dfrac{2xyi}{(x^2+y^2)^2}$

演習問題 6(追加) 正則函数の合成函数が正則函数となることを, Cauchy-Riemann の方程式を確かめることにより Mathematica で示せ.