

C言語の学習 数学関数

山本昌志*

2007年6月27日

概要

数学関数の取り扱い方法を学ぶ。はじめは、実数型の関数の取り扱いについて、説明する。それに慣れたならば、複素数型の取り扱い方法を学習する。

1 本日の学習内容

本日の内容は、教科書 [1] の 17 章の実数型の数学関数 (p.351-) と付録にある複素数型 (p.471) の取り扱いについて学習する。ただし、教科書には、複素数型の関数についての説明がないので、このプリントで補わなくてはならない。技術者にとって、複素数の扱いは、非常に重要である。

本日の学習のゴールは、以下の通りである。

- 実数型の数学関数の使い方が分かる。ヘッダーファイルの書き方と関数の使い方、マクロ定数、コンパイル方法を理解する必要がある。
- 複素数型の数学関数の使い方が分かる。ヘッダーファイルの書き方と複素数の変数宣言、複素数の表示方、複素関数の使い方、コンパイル方法を理解する必要がある。

2 実数の関数

2.1 数学関数の例

C言語では、ヘッダーファイル `math.h` をインクルードすることにより、おなじみの数学の初等関数を使うことができる。具体的には、リスト 1 のようにする。

リスト 1: 実数型の関数の使用例

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main(void){
5     double x, s, c, t, l, e;
6
7     x=M_PI;
8 }
```

*独立行政法人 秋田工業高等専門学校 電気工学科

```

9   s=sin(x);
10  c=cos(x);
11  e=exp(x);
12  l=log(x);
13
14  printf("sin(pi)=%f\n",s);
15  printf("cos(pi)=%f\n",c);
16  printf("tan(pi)=%f\n",t);
17  printf("exp(pi)=%f\n",e);
18  printf("log(pi)=%f\n",l);
19
20  return 0;
21 }

```

実行結果

```

sin(pi)=0.000000
cos(pi)=-1.000000
tan(pi)=0.000000
exp(pi)=23.140693
log(pi)=1.144730

```

このプログラムの各行の内容は、次の通りである。

- 2行目 `#include <math.h>`
 数学関数を使うために、ヘッダーファイル `math.h` をインクルードしている。数学関数を使う場合、必ず必要である。
- 7行目 `x=M_PI;`
`M_PI` は円周率を表すマクロである。`math.h` をインクルードすると、`M_PI` はコンパイル時に、`3.1415926...` に置き換わる。
- 9-12行目 `s=sin(x);` など

おなじみの数学関数が並んでいる。右辺の関数の戻り値を左辺の変数に代入している。

2.2 数学関数の使用方法

2.2.1 記述方法

数学関数を使うためには、`math.h` をインクルードすることを忘れてはならない。まずは、これを書く。C 言語の数学関数は、数学で使う初等関数とほとんど同じ記述のため簡単である。必要な関数を数学で学習したように記述すればよい。`math.h` に用意されている関数は、表 1 のとおである。引数も戻り値も倍精度実数型である。

数学の計算でしばしば使われる定数は、`math.h` でマクロとして定義されている。定義されているマクロを表 2 に示す。いろいろ定義されているが、円周率を表す `M_PI` を覚えておけば、ほとんどの場合事足りる。次に重要なのは、ネピア数を表す `M_E` くらいである。

2.2.2 コンパイル方法

数学関数を含んだソースファイルをコンパイルする場合には、`libm` というライブラリーをリンクする必要がある。このライブラリーが数学関数の実体である。数学関数が使われているときには、

```
gcc -o fugafuga hoge hoge.c -lm
```

のようにする。`hoge hoge.c` がソースファイルで、`fugafuga` が実行ファイルである。オプション `-lm` をつけることにより、数学関数のライブラリー `libm` をリンクしている。

表 1: `math.h` で定義されている関数。関数の引数は倍精度実数である。戻り値も倍精度実数である。滅多に使わない関数 `fmod`, `ldexp`, `modf` は省略。

数学関数名	C 言語関数	引数 x	戻り値
三角関数	<code>sin(x)</code>	単位はラジアン	$\sin x$ の値
	<code>cos(x)</code>	単位はラジアン	$\cos x$ の値
	<code>tan(x)</code>	単位はラジアン	$\tan x$ の値
逆三角関数	<code>asin(x)</code>	範囲 $[-1, +1]$	範囲 $[-\pi/2, +\pi/2]$ ラジアン
	<code>acos(x)</code>	範囲 $[-1, +1]$	範囲 $[0, \pi]$ ラジアン
	<code>atan(x)</code>		範囲 $[-\pi/2, +\pi/2]$ ラジアン
	<code>atan2(x, y)</code>		$\arctan(x/y)$ の値で範囲 $[-\pi, \pi]$ ラジアン
双曲線関数	<code>sinh(x)</code>		$\sinh x$ の値
	<code>cosh(x)</code>		$\cosh x$ の値
	<code>tanh(x)</code>		$\tanh x$ の値
指数関数	<code>exp(x)</code>		e^x の値
対数	<code>log(x)</code>	$0 \leq x$	自然対数 $\log_e x$ の値
	<code>log10(x)</code>	$0 \leq x$	常用対数 $\log_{10} x$ の値
絶対値	<code>fabs(x)</code>		$ x $
平方根	<code>sqrt(x)</code>		\sqrt{x}
べき乗	<code>pow(x, y)</code>	x も y も実数可	x^y の値。複素数の場合エラー
整数部	<code>floor(x)</code>		x 以下の最大の整数値を <code>double</code> 型で返す
	<code>ceil(x)</code>		x 以上の最小の整数値を <code>double</code> 型で返す

表 2: math.h で定義されているマクロ定数 .

数学定数名	数学記号	C 言語マクロ	値
円周率	π	M_PI	3.14159265358979323846
	$\pi/2$	M_PI_2	1.57079632679489661923
	$\pi/4$	M_PI_4	0.78539816339744830962
	$1/\pi$	M_1_PI	0.31830988618379067154
	$2/\pi$	M_2_PI	0.63661977236758134308
	$2/\sqrt{\pi}$	M_2_SQRTPI	1.12837916709551257390
ネピア数	e	M_E	2.7182818284590452354
	$\log_2 e$	M_LOG2E	1.4426950408889634074
	$\log_{10} e$	M_LOG10E	0.43429448190325182765
対数	$\log_e 2$	M_LN2	0.69314718055994530942
	$\log_e 10$	M_LN10	2.30258509299404568402
平方根	$\sqrt{2}$	M_SQRT2	1.41421356237309504880
	$1/\sqrt{2}$	M_SQRT1_2	0.70710678118654752440

2.3 練習問題

[練習 1] キーボードより変数の値を読み込み，以下の関数の値を表示せよ．

- 三角関数 (sin, cos, tan)
- 指数関数
- 自然対数関数と常用対数関数
- 平方根と立方根

[練習 2] キーボードより変数の値を読み込んで，`sqrt(x)` と `pow(x, 0.5)` の関数の値を比較せよ．

[練習 3] `math.h` で定義されたマクロ定数を用いて，円周率 π とネピア数 e の値を小数点以下 20 桁表示せよ．

[練習 4] 自然対数の関数 `log(x)` の変数 x を負の値とした場合，どのようなになるか？ プログラムを作って調べよ．

3 複素数と複素関数

3.1 複素関数を使った例

以前の C 言語は複素数がサポートされていなかった。数値計算をする場合、複素数が使えないとかなり不便を強いられる。そのため、複素数ができる FORTRAN から抜け出せない人が多くいた。新しい C 言語では、複素数がサポートされている。これは非常にありがたい。

実際に複素数や複素関数を使った例をリスト 2 に示す。これは、オイラーが発見した式

$$e^{i\pi} = -1 \quad (1)$$

の計算結果である。この式はとても不思議で、25 年くらい前にはじめてみたときには大変驚いた記憶がある。それまではなんの関係もないと思っていた円周率 π とネピア数 e と虚数単位 i が、こんなにも簡単な式で結ばれるのである。

リスト 2: 複素数型の関数の使用例

```
1 #include <stdio.h>
2 #include <complex.h>
3 #include <math.h>
4
5 int main(void){
6     double _Complex z, x;
7
8     x=I*M_PI;
9     z=cexp(x);
10
11     printf(" real=%f\timag=%f\n", creal(z),cimag(z));
12
13     return 0;
14 }
```

実行結果

```
real=-1.000000  imag=0.000000
```

このプログラムの各行の内容は、次の通りである。

- 2 行目 `#include <complex.h>`
複素数および複素関数を使うために、ヘッダーファイル `complex.h` をインクルードしている。複素数を使う場合、必ず必要である。
- 6 行目 `double _Complex z, x;`
倍精度複素数型の変数の宣言である。複素数型の変数 `z` と `x` が使えるようになる。
- 8 行目 `x=I*M_PI;`
先に述べたように、`M_PI` は円周率を表すマクロである。`I` は虚数単位である。したがって、C 言語の `I*M_PI` は、数学の $i\pi$ を表す。
- 9 行目 `z=cexp(x);`

`cexp(x)` は、数学の e^x を表す。ただし、変数も関数の値も複素数となる。

- 11 行目 `creal(z), cimag(z)`

`creal(z)` で複素数 z の実数部を、`cimag(z)` で虚数部を取り出している。

3.2 複素関数の使い方

3.2.1 記述方法

ヘッダーファイル ヘッダーファイル `complex.h` をインクルードする必要がある。プログラムの前の方に、

```
#include <complex.h>
```

と書く。

変数宣言 複素数の計算では、複素数型の変数宣言が必要である。変数宣言の例を、以下に示す。

```
float _Complex a, b, hoge;
double _Complex c, d, fuga;
long double _Complex e, f, foo;
```

のようにする。通常は、`double _Complex` を使うこと。C 言語で実数を扱う場合は `double`、複素数を扱う場合は `double _Complex` とするのが無難である。

複素数 虚数単位は I である。数学は小文字を使うが、C 言語では大文字である。複素数型の変数に値を代入するためには、次のようにする。

```
z=x+I*y;
w=3.1415+I*2.718281828;
```

四則演算 四則演算は特に気にすることもなく、普通に演算子 (+, -, *, /) が使える。

複素関数 表 3 のような関数が用意されている。よほどのことがない限り、倍精度型を使うこと。

表 3: `complex.h` で定義されている関数。関数の引数と戻り値は同じ型である。引数が倍精度複素数であれば、戻り値は倍精度複素数または倍精度実数である。

数学関数名	倍精度	単精度	拡張倍精度	戻り値
三角関数	<code>csin(x)</code>	<code>csinf(x)</code>	<code>csinl(x)</code>	複素数
	<code>ccos(x)</code>	<code>ccosf(x)</code>	<code>ccosl(x)</code>	複素数
	<code>ctan(x)</code>	<code>ctanf(x)</code>	<code>ctanl(x)</code>	複素数
逆三角関数	<code>casin(x)</code>	<code>casinf(x)</code>	<code>casinl(x)</code>	複素数
	<code>cacos(x)</code>	<code>cacosf(x)</code>	<code>cacosl(x)</code>	複素数
	<code>catan(x)</code>	<code>catanf(x)</code>	<code>catanl(x)</code>	複素数
双曲線関数	<code>csinh(x)</code>	<code>csinhf(x)</code>	<code>csinhl(x)</code>	複素数
	<code>ccosh(x)</code>	<code>ccoshf(x)</code>	<code>ccoshl(x)</code>	複素数
	<code>ctanh(x)</code>	<code>ctanhf(x)</code>	<code>ctanhl(x)</code>	複素数
逆双曲線関数	<code>casinh(x)</code>	<code>casinhf(x)</code>	<code>casinhl(x)</code>	複素数
	<code>cacosh(x)</code>	<code>cacoshf(x)</code>	<code>cacoshl(x)</code>	複素数
	<code>catanh(x)</code>	<code>catanhf(x)</code>	<code>catanhl(x)</code>	複素数
指数関数	<code>cexp(x)</code>	<code>cexpf(x)</code>	<code>cexpl(x)</code>	複素数
自然対数	<code>clog(x)</code>	<code>clogf(x)</code>	<code>clogl(x)</code>	複素数
絶対値	<code>cabs(x)</code>	<code>cabsf(x)</code>	<code>cabsl(x)</code>	実数
平方根	<code>csqrt(x)</code>	<code>csqrtf(x)</code>	<code>csqrtl(x)</code>	複素数
べき乗	<code>cpow(x,y)</code>	<code>cpowf(x,y)</code>	<code>cpowl(x,y)</code>	複素数 (x^y)
実部	<code>creal(x)</code>	<code>crealf(x)</code>	<code>creall(x)</code>	実数
虚部	<code>cimag(x)</code>	<code>cimagf(x)</code>	<code>cimagl(x)</code>	実数
偏角	<code>carg(x)</code>	<code>cargf(x)</code>	<code>cargl(x)</code>	実数
複素共役	<code>conj(x)</code>	<code>conjf(x)</code>	<code>conjl(x)</code>	複素数
リーマン球の射影	<code>cproj(x)</code>	<code>cprojf(x)</code>	<code>cprojl(x)</code>	複素数

3.2.2 コンパイル方法

実数型と同じように、オプション `-lm` をつける。

```
gcc -o fugafuga hoge.hoge.c -lm
```

3.3 練習問題

[練習 5] 複素数 $z_1 = i$ と $z_2 = 1 + i$ について、以下の値を計算せよ。

- 2 乗と 3 乗
- 平方根と立方根

- 絶対値
- 複素共役

[練習 6] オイラーの関係式 $e^{i\theta} = \cos \theta + i \sin \theta$ が成り立つことを, $0 \leq \theta \leq 2\pi$ の範囲で調べよ. この間を 360 等分して, 両辺の値を出力する.

[練習 7] i^i の値を計算してみよ. そして, 解析解と比較せよ.

4 課題

4.1 内容

以下の課題を実施し, レポートとして提出すること.

[問 1] (復) 教科書 [1] の第 17 章 (pp.342-374) を 3 回読め. レポートには「3 回読んだ」と書け.

[問 2] (復) 本日配布したプリントを 2 回読め. レポートには「2 回読んだ」と書け. さらに, 誤字・脱字, 表現の悪いところ, 間違いを指摘せよ.

[問 3] (復) 三角関数 ($\sin x, \cos x, \tan x$) の値を $0 \sim 360[\text{deg}]$ まで, $1[\text{deg}]$ 間隔で計算して表示せよ.

[問 4] (復) オイラーの関係式 $e^{i\theta} = \cos \theta + i \sin \theta$ が成り立つことを, $0 \leq \theta \leq 2\pi$ の範囲で調べよ. この間を 360 等分して, 両辺の値を出力する.

[問 5] (復) 積分の定義—リーマン和—to に従い以下の積分を行え.

$$S = \oint_C \frac{1}{z} dz$$

ただし, 積分の経路 C は, 図 1 の通りとする. もちろん, $z = x + iy$ である.

[問 6] (予) 教科書 [1] の第 18 章を 2 回読め. レポートには「2 回読んだ」と書け.

[問 7] ここでの学習内容でわからないところがあれば, 具体的に記述せよ.

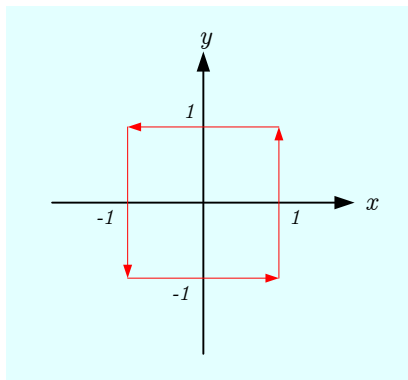


図 1: 積分の経路.

4.2 レポート提出要領

期限	7月3日(水) AM 8:45
用紙	A4のレポート用紙．左上をホッチキスで綴じて，提出のこと．
提出場所	山本研究室の入口のポスト
表紙	表紙を1枚つけて，以下の項目を分かりやすく記述すること． 授業科目名「計算機応用」 課題名「数学関数」 提出日 5E 学籍番号 氏名
内容	2ページ以降に問いに対する答えを分かりやすく記述すること．

参考文献

- [1] 林春比古. 新訂 C 言語入門 シニア編. ソフトバンク パブリッシング, 2004.