

MATLAB プログラミング I

目次

1	プログラム（関数）あるいはスクリプトの実行	4
2	変数	7
3	基本関数	13
4	分岐（if 文, 演算子）	16
5	繰り返し計算（for 文）	20
6	繰り返し計算（while 文）	24
7	場合分け（switch 文）	26

8	関数	28
9	ベクトルの基本的な演算	33
10	行列の基本的な演算	41
11	入出力 (ワークスペース変数)	53
12	入出力 (テキストファイル)	55
13	2次元グラフィック	61
14	3次元グラフィック	65

1 プログラム（関数）あるいはスクリプトの実行

MATLAB を起動する。コマンドウィンドウで、対話的にコマンドを入力して結果が得られる。コマンドは、MATLAB で標準的に定義されたもの、オプションの Toolbox で定義されたもの、自分で定義したものが同じように使用できる。

■カレントディレクトリの設定 カレントディレクトリ・ウィンドウで表示されるディレクトリは、MATLAB を起動するショートカットのプロパティを開き、作業フォルダのところで指定できる。

■コマンドウィンドウ設定 ファイル → 設定を選択し、コマンドウィンドウ設定のタブを表示させる。テキスト表示の数値表示を compact に設定すると、コマンドウィンドウの `ans =` 等の結果を空行なしに表示できる。

■M-ファイル プログラムあるいはスクリプトのソースコードは、M-ファイルに記述する。ファイル名の拡張子は、`m` である。ファイル → 新規作成 → M-ファイルを選択すると、エディタが起動する。作成した M-ファイルをコマンドウィンドウで入力すれば、コマンドとして実行できる（コンパイル、リンクは不要）。

■一つの実行文

- 改行： 変数を入力する.
- 「 ; 」： 変数を入力しない.

■複数行からなる一つの実行文 行末に「 ... 」をつけると、実行文が次の行に継続していることを示す.

■コメント 「 % 」に続く行は、コメント文となる（行単位で使用）.

■disp 値や文字の標準出力. 文字の出力は、文字を ' と ' で囲む.

■スクリプト（M1_ex01.m） 文字を入力するだけの簡単な例

```
disp('hello, world'); % 画面に出力  
disp(' こんにちは'); % 日本語も出力できる
```

■出力結果

hello, world

こんにちは

2 変数

■変数の名前 最初の 31 文字を認識。アルファベット、数字、下線を使用できる。大文字と小文字は区別される。

■変数の型 型宣言はない。右辺の数値によって決まる。精度は倍精度 (double)。ただし、出力形式は、`format` の設定による。

■変数の初期化 `clear all`

通常、変数を最初に初期化することが多い。

■局所的な変数と大域的な変数 通常、`function` 中で定義した変数は、局所的な変数である。一方、`global` というコマンドを使用すれば、`function` にまたがるような大域的な変数を定義できる。慣習として、大域的な変数は大文字がよく使用される。

■基本的な数量 (スカラ) 変数 `pi` は π 、変数 `i` および `j` は虚数単位として予め定義されている。

■複素数 虚数単位 i および j は数値の後につける。 i および j に別の値を代入することもできるので、注意が必要。また、`sqrt(-1)` 等により自分で虚数単位を定義することもできる。複素数の変数を z とすると、

- 実部： `real(z)`
- 虚部： `imag(z)`
- 振幅： `abs(z)`
- 位相： `angle(z)` [rad]
- 共役複素数： `conj(z)`, z'

■演算について

- 整数, 実数, 複素数のスカラー演算, ベクトルおよび行列演算ができる。
- $+$, $-$, $*$, $/$, (四則演算), $^$ (べき乗) 等を用いてスカラー演算できるが, 行列演算と区別するため, スカラー演算では前に「 $.$ 」をつけた `.*`, `./`, `.^` も使用できる。
- $+$, $-$, $*$, $/$, $^$ (べき乗) 等を用いて, ベクトルおよび行列演算ができる (詳細は後述)。

■ `printf` 値や文字の出力（詳細は後述）。文字の出力は、文字を' と' で囲む。

■ 変数の情報 `whos`

■ スクリプト (M1_ex02.m) スカラ演算の説明

```
clear all
disp(' プログラム名:  M1_ex02.m ');

format long g
pi
fprintf(1,' pi= %f\n',pi); % pi

% 実数
format short g
a = 2 % ; がないと, 変数の値が出力される
fprintf(1,' 整数: a= %d\n',a); % 変換文字 d は 10 進数整数表示
b = 123.45
fprintf(1,' 実数 (変換文字 f): b= %f\n',b); % 変換文字 f は固定小数点表示
c = 6.0e12
fprintf(1,' 実数 (変換文字 e): c= %e\n',c); % 変換文字 e,E は指数表示
fprintf(1,' 実数 (変換文字 g): c= %g\n',c);
% 変換文字 g は固定小数点表示あるいは指数表示のコンパクト表示 (ゼロは表示しない)
```

```
b1 = a^10; b2 = a*b; % 一行に複数の文を書くことができる
fprintf(1,'b1= %5d, b2= %6.2f\n',b1,b2); % 表示桁数の指定

% 複素数
z1 = 1.0-1.0i;
fprintf(1,' 複素数: z1= %f + i %f\n',real(z1),imag(z1));
fprintf(1,' 振幅: %f, 位相: %f [rad] %f [deg]\n', ...
        abs(z1),angle(z1),angle(z1)*180.0/pi);
z2 = conj(z1); z3 = z1';
fprintf(1,' 共役複素数: z2= %f + i %f\n',real(z2),imag(z2));
fprintf(1,' 共役複素数: z3= %f + i %f\n',real(z3),imag(z3));
ii = sqrt(-1) % 虚数単位の定義

% ワークスペース内の変数情報の出力
whos
```

■ 出力結果

```
プログラム名:  m1_ex02.m
ans =
          3.14159265358979
pi= 3.141593
a =
     2
```

整数：a= 2

b =

123.45

実数 (変換文字 f) : b= 123.450000

c =

6e+012

実数 (変換文字 e) : c= 6.000000e+012

実数 (変換文字 g) : c= 6e+012

b1= 1024, b2= 246.90

複素数 : z1= 1.000000 + i -1.000000

振幅 : 1.414214, 位相 : -0.785398 [rad] -45.000000 [deg]

共役複素数 : z2= 1.000000 + i 1.000000

共役複素数 : z3= 1.000000 + i 1.000000

ii =

Name	0 +	1i	Bytes	Class	Attributes
a	1x1		8	double	
ans	1x1		8	double	
b	1x1		8	double	
b1	1x1		8	double	
b2	1x1		8	double	
c	1x1		8	double	
ii	1x1		16	double	complex
z1	1x1		16	double	complex
z2	1x1		16	double	complex

z3

1x1

16 double complex

3 基本関数

■引数が1つの基本関数の例

- 平方根： `sqrt (引数)`
- 絶対値： `abs (引数)`, 以下引数省略
- 三角関数： `sin, cos, tan, cot, sec, csc , sinh, cosh, tanh, sech, csch`
- 三角関数の逆関数： `asin, acos, atan, acot, asec , asinh, acosh, atanh`
- 常用対数： `log10`
- 自然対数： `log`
- 指数関数： `exp`

■引数が2つの組み込み関数の例

- `atan2 (引数, 引数)`
- 除算の剰余： `rem (引数, 引数)`

■スクリプト (M1_ex03.m) 基本関数の使用例

```
clear all

% 基本数学関数
a1 = sqrt(2.0);
fprintf(1,'平方根:a1= %f\n',a1);
a2 = exp(1.0);
fprintf(1,'指数:a2= %f\n',a2);
a3 = log10(0.5);
fprintf(1,'常用対数:a3= %f\n',a3);
a4 = log(a2);
fprintf(1,'自然対数:a3= %f\n',a4);

% 三角関数
a5 = sin(pi/6.0); a6 = cos(0.0); a7 = tan(pi/4.0);
fprintf(1,'sin:a5= %f cos:a6= %f tan:a7= %f\n',a5,a6,a7);
a8 = asin(a5);
fprintf(1,'arcsin: %f %f\n',a8,pi/6.0);
a9 = acos(a6);
fprintf(1,'arccos: %f %f\n',a9,0.0);
a10 = atan(a7); a11 = atan2(-sin(pi/4.0),cos(pi/4.0));
fprintf(1,'arctan: %f %f %f\n',a10,pi/4.0,a11);

% その他
b1 = abs(-2.0);
```

```
fprintf(1, ' 绝对值 : b1= %f\n', b1);  
i1 = rem(53, 5);  
fprintf(1, ' 剩余 : i1= %f\n', i1);
```

■ 出力結果

```
平方根 : a1= 1.414214  
指数 : a2= 2.718282  
常用对数 : a3= -0.301030  
自然对数 : a3= 1.000000  
sin : a5= 0.500000  cos : a6= 1.000000  tan : a7= 1.000000  
arcsin : 0.523599  0.523599  
arccos : 0.000000  0.000000  
arctan : 0.785398  0.785398  -0.785398  
绝对值 : b1= 2.000000  
剩余 : i1= 3.000000
```

4 分岐 (if 文, 演算子)

■一つの判定条件

```
if expression (判定条件)
    statements (単一あるいは複数の実行文)
end
```

■2つに分岐

```
if expression (判定条件)
    statements1 (単一あるいは複数の実行文 1)
else
    statements2 (単一あるいは複数の実行文 2)
end
```

■一般的な形式

```
if expression1 (判定条件 1)
    statements1 (単一あるいは複数の実行文 1)
elseif expression2 (判定条件 2)
```


statements2 (単一あるいは複数の実行文 2)

else

statements3 (単一あるいは複数の実行文 3)

end

■ 比較演算子

- > 大きい (greater than, >)
- >= 大きいか等しい (greater than or equal to, \geq)
- ~= 等しくない (not equal to, \neq)
- == 等しい (equal to, =)
- < 小さいか等しい (less than or equal to, \leq)
- <= 小さい (less than, <)

■ 論理演算子

- ~ 否定
- & 論理積

- | 論理和

■NaN ゼロ割り等で計算できない不定値は, NaN (Not-a-Number) となる.

■スクリプト (M1_ex04.m) if 文の簡単な使用例

```
clear all

x = 0
y = sin(x)/x
if x==0
    y = 1.0;
else
    y = sin(x)/x;
end
fprintf(1,'x= %f, y= %f\n',x,y);
```

■出力結果

```
x =
    0
y =
NaN
```

x= 0.000000, y= 1.000000

5 繰り返し計算 (for 文)

- ループカウンタ変数 (*variable*) が 1 ずつ増加

```
for variable = initval (初期値) : endval (終値)
    statements (単一あるいは複数の実行文)
end
```

- 一般的な形式

```
for variable = initval (初期値) : stepval (増分) : endval (終値)
    statements (単一あるいは複数の実行文)
end
```

- スクリプト (M1_ex05.m) for 文のいくつかの使用例

```
clear all

n1 = 1; % 初期値
n2 = 10; % 終値
```

```

for I = n1:n2
    fprintf(1,' %d',I);
end

fprintf(1,'\n\n');
J = 0;
for I = 3:-1:1
    J = J+1;
    fprintf(1,' %d %d\n',I,J);
end

disp(' ');
for I = 1:3
    fprintf(1,' %d %f\n',I,cos(I*pi));
end

x0 = 1.0e3; dx = 3.0e3; nx = 3;
fprintf(1,'\n x0= %f, dx= %f, nx= %d\n',x0,dx,nx);
for I = 1:nx
    x = x0+dx*(I-1);
    fprintf(1,' %d %f\n',I,x);
end

fprintf(1,'\n %2s %2s %8s %8s %8s %8s\n',...
        'I','J','real(z1)','imag(z1)','real(z2)','imag(z2)');
for I = 1:2

```

```
for J = 1:3
    z1 = j*I*J;
    z2 = exp(z1*pi);
    fprintf(1,' %2d %2d %8.3f %8.3f %8.3f %8.3f\n',...
           I,J,real(z1),imag(z1),real(z2),imag(z2));
end
end
```

■ 出力結果

```
1 2 3 4 5 6 7 8 9 10
```

```
3 1
```

```
2 2
```

```
1 3
```

```
1 -1.000000
```

```
2 1.000000
```

```
3 -1.000000
```

```
x0= 1000.000000, dx= 3000.000000, nx= 3
```

```
1 1000.000000
```

```
2 4000.000000
```

```
3 7000.000000
```

I	J	real(z1)	imag(z1)	real(z2)	imag(z2)
1	1	0.000	1.000	-1.000	0.000
1	2	0.000	2.000	1.000	-0.000
1	3	0.000	3.000	-1.000	0.000
2	1	0.000	2.000	1.000	-0.000
2	2	0.000	4.000	1.000	-0.000
2	3	0.000	6.000	1.000	-0.000

6 繰り返し計算 (while 文)

- 一般的な形式 条件が真である間は実行文を繰り返す。

```
while expression (判定条件)
    statements (単一あるいは複数の実行文)
end
```

- スクリプト (M1_ex06.m) while 文の使用例

```
clear all

s1 = 0; s2 = 0; I = 0;
while I<100
    I = I+1;
    s1 = s1+I;
    s2 = s2+I.^2;
end
I, s1, s2
```

- 出力結果


```
I =  
    100  
s1 =  
    5050  
s2 =  
    338350
```

7 場合分け (switch 文)

■一般的な形式

```
switch switch_expr
    case case_expr (switch_expr が case_expr と一致すれば実行)
        statement, ... , statement
    case case_expr1, case_expr2, case_expr3, ...
        statement, ... , statement
    otherwise
        statement, ... , statement
end
```

■スクリプト (M1_ex07.m) switch 文の簡単な使用例

```
ip = 1;
fprintf(1,'ip = %d\n',ip);
switch ip
    case 1
        disp('hello, world');
    case 2
```

```
        disp('こんにちは');  
otherwise  
        disp('otherwise');  
end
```

■出力結果

```
ip = 1  
hello, world
```

8 関数

■M-ファイル関数, サブ関数の一般的な形式

```
function [out1, out2, ...] = funname(in1, in2, ...) (関数の定義行)
    ...
    ...
end
```

ただし,

- `function`: 関数 (`function`) の始まりを示す.
- `funname`: 関数名. MATLAB で予め定義されている関数は小文字であるので, これらと区別するために, 1文字目を大文字にすることが多い.
- `in1`, `in2`, ...: 入力変数名
- `out1`, `out2`, ...: 出力変数名
- `end`: `function` の終わりを示す (省略可能).

M-ファイルには, 一つの M-ファイル関数 (スクリプトの場合は不要), 複数のサブ関数を定義できる.

■ その他の関数定義行の例

```
function funname1 % 関数名のみで, 入出力変数なし
function funname2(in1, in2, ...) % 出力変数なし
function out = funname3(in1, in2, ...) % 出力変数が一つだけ
```

■ 関数の呼び出しの例

```
funname1 % 関数名のみで, 入出力変数なし
funname2(in01, in02, ...) % 出力変数なし
out00 = funname3(in01, in02, ...) % 出力変数が一つだけ
[out01, out02, ...] = funname(in01, in02, ...) % 一般的な例
```

■ コメント行

```
function [out1, out2, ...] = funname(in1, in2, ...)
% 関数の機能 (概略)
% ヘルプテキスト行 (入出力変数の型, 説明等)
```

⌘ ヘルプテキスト行

... (以下, 関数の本体)

...

end

■ nargin 引数の数を与える組込関数

■ プログラム (M1_ex08.f) 関数の例

```
function M1_ex08
clear all
x0 = -8.0; dx =2.0; nx = 9;
fprintf(1,'%10s %10s %10s %10s\n','x','sin(x)/x',' [dB]',' [deg]');
for I=1:nx
    x = x0+dx*(I-1);
    z = Func1(x);
    [db,ph] = Func2(z);
    fprintf(1,'%10.3f %10.3f %10.3f %10.3f\n',x,z,db,ph);
end
end

function y = Func1(x)
% sinc 関数
```

```

if x==0
    y = 1.0;
else
    y = sin(x)/x;
end
end

function [db, ph_deg] = Func2(z)
% デシベル, 位相
% z: real or complex
% dB: [dB], ph_deg: phase [deg]
db = 20.0*log10(abs(z));
ph = angle(z);
ph_deg = ph*180.0/pi;
end

```

■出力結果

x	sin(x)/x	[dB]	[deg]
-8.000	0.124	-18.155	0.000
-6.000	-0.047	-26.638	180.000
-4.000	-0.189	-14.462	180.000
-2.000	0.455	-6.846	0.000
0.000	1.000	0.000	0.000

2.000	0.455	-6.846	0.000
4.000	-0.189	-14.462	180.000
6.000	-0.047	-26.638	180.000
8.000	0.124	-18.155	0.000

9 ベクトルの基本的な演算

■行ベクトルの定義 n 個の要素からなる行ベクトル $[A]$, $[B]$

$$[A] = (a_1 \ a_2 \ \dots \ a_n), \quad [B] = (b_1 \ b_2 \ \dots \ b_n) \quad (1)$$

は次のように定義される.

$$A = [a_1, a_2, \dots, a_n]$$

$$B = [b_1 \ b_2 \ \dots \ b_n] \quad (「,」は省略可能)$$

■行ベクトルの要素 行ベクトル $[A]$ の I 列目の要素は, $A(I)$ で指定する.

■スカラ変数 c と行ベクトル A との演算

- $c+A=A+c$:

$$(c + a_1 \ c + a_2 \ \dots \ c + a_n) \quad (2)$$

- $c-A=-A+c$:

$$(c - a_1 \ c - a_2 \ \dots \ c - a_n) \quad (3)$$

- $c \cdot *A = A \cdot *c$:

$$(ca_1 \quad ca_2 \quad \dots \quad ca_n) \quad (4)$$

- $c \cdot /A$:

$$(c/a_1 \quad c/a_2 \quad \dots \quad c/a_n) \quad (5)$$

- $A \cdot /c$:

$$(a_1/c \quad a_2/c \quad \dots \quad a_n/c) \quad (6)$$

- $A \cdot \hat{c}$:

$$(a_1^c \quad a_2^c \quad \dots \quad a_n^c) \quad (7)$$

- $c \cdot \hat{A}$:

$$(c^{a_1} \quad c^{a_2} \quad \dots \quad c^{a_n}) \quad (8)$$

■ 行ベクトルの要素に関するスカラー演算

- $A+B=B+A$:

$$(a_1 + b_1 \quad a_2 + b_2 \quad \dots \quad a_n + b_n) \quad (9)$$

- $A-B=-B+A$:

$$(a_1 - b_1 \quad a_2 - b_2 \quad \dots \quad a_n - b_n) \quad (10)$$

- $A \cdot B=B \cdot A$:

$$(a_1 b_1 \quad a_2 b_2 \quad \dots \quad a_n b_n) \quad (11)$$

- $A \cdot /B$:

$$(a_1/b_1 \quad a_2/b_2 \quad \dots \quad a_n/b_n) \quad (12)$$

- $A \cdot ^B$:

$$(a_1^{b_1} \quad a_2^{b_2} \quad \dots \quad a_n^{b_n}) \quad (13)$$

■列ベクトルの定義 n 個の要素からなる列ベクトル $[D]$

$$[D] = \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} \quad (14)$$

は次のように定義される.

$$D = [d_1; d_2; \dots ; d_n] \quad (n \text{ 行 } 1 \text{ 列の行列})$$

スカラ変数と列ベクトルとの演算は、行ベクトルの場合と同様である.

■転置「 $'$ 」

- $D.'$: 行ベクトル (1 行 n 列の行列) $[d_1, d_2, \dots , d_n]$ に変換される.

$$(d_1 \quad d_2 \quad \dots \quad d_n) \quad (15)$$

- $A \cdot v$: 列ベクトル (n 行 1 列の行列) $[a_1; a_2; \dots ; a_n]$ に変換される.

$$\begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}$$

(16)

■全要素がゼロのベクトル `zeros` や 1 のベクトル `ones`

- n 行ベクトル (1 行 n 列の行列) : `zeros(1,n)`, `ones(1,n)`
- m 列ベクトル (m 行 1 列の行列) : `zeros(m,1)`, `ones(m,1)`

■等間隔な成分をもつベクトル (等差数列)

- $A = \text{init}A$ (初期値) : $\text{step}A$ (増分) : $\text{end}A$ (終値)
`for` のループ変数と同様に定義できる.
- $A = \text{linspace}(\text{init}A, \text{end}A, n)$
組み込み関数より定義できる. ただし, n は成分の数を示し, 省略すると $n = 100$ になる.

■定義済のベクトル A の成分の数 `length(A)`

■ベクトルの内積 `dot(A, B)`

■ プログラム (M1_ex09.m) ベクトルの簡単な計算例

```
function M1_ex09
format short g

va = [3.0, 4.0, 0.0]
length(va)

% ベクトルのスカラー積 (内積) の計算
va2 = va.*va;
sp = sum(va2);
fprintf('sp = %f\n', sp);

% ベクトルの振幅および単位ベクトルの計算
amp = sqrt(sp);
vb = va./amp;
fprintf('amp = %f, va= %f %f %f\n', amp, vb(1:3));

fprintf('zeros = %f %f %f\n', zeros(1,3));
fprintf('ones = %f %f %f\n', ones(3,1)');

% 等間隔な成分をもつベクトル
vc = 1:0.25:2
vd = linspace(1,2,5)
end
```

■ 出力結果

```
va =  
    3    4    0  
ans =  
    3  
sp = 25.000000  
amp = 5.000000, va= 0.600000 0.800000 0.000000  
zeros = 0.000000 0.000000 0.000000  
ones  = 1.000000 1.000000 1.000000  
vc =  
      1      1.25      1.5      1.75      2  
vd =  
      1      1.25      1.5      1.75      2
```


10 行列の基本的な演算

■行列の定義 m 行 n 列の行列 $[E]$, $[F]$

$$[E] = \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & & & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{pmatrix}, [F] = \begin{pmatrix} f_{11} & f_{12} & \dots & f_{1n} \\ f_{21} & f_{22} & \dots & f_{2n} \\ \vdots & & & \vdots \\ f_{m1} & f_{m2} & \dots & f_{mn} \end{pmatrix} \quad (17)$$

は次のように定義される.

$E = [e_{11}, e_{12}, \dots, e_{1n}; e_{21}, e_{22}, \dots, e_{2n}; \dots \dots \dots e_{mn}]$

$F = [f_{11} \ f_{12} \ \dots \ f_{1n}; f_{21} \ f_{22} \ \dots \ f_{2n}; \dots \dots \dots f_{mn}]$

(行の区切りは「;」)

■行列の要素 行列 $[E]$ の I 行 J 列目の要素は, $E(I, J)$ で指定する.

■定義済のベクトルによる行列の定義

$$[E_1] = \begin{pmatrix} e_{11} \\ e_{21} \\ \vdots \\ e_{m1} \end{pmatrix}, \quad [E_2] = \begin{pmatrix} e_{12} \\ e_{22} \\ \vdots \\ e_{m2} \end{pmatrix}, \quad \dots \quad [E_n] = \begin{pmatrix} e_{1n} \\ e_{2n} \\ \vdots \\ e_{mn} \end{pmatrix} \quad (18)$$

で定義された E_1, E_2, \dots, E_n を用いると,

$$E = [E_1, E_2, \dots, E_n]$$

$$[F_1] = (f_{11} \quad f_{12} \quad \dots \quad f_{1n}), \quad [F_2] = (f_{21} \quad f_{22} \quad \dots \quad f_{2n}), \quad \dots$$

$$[F_m] = (f_{m1} \quad f_{m2} \quad \dots \quad f_{mn}) \quad (19)$$

で定義された F_1, F_2, \dots, F_m を用いると,

$$F = [F_1; F_2; \dots; F_m]$$

■スカラ変数と行列との演算 ベクトルのときと同様.

■2つの行列 E, F (行列の大きさは同じ) の成分ごとの演算

• $E+F$:

$$\begin{pmatrix} e_{11} + f_{11} & e_{12} + f_{12} & \dots & e_{1n} + f_{1n} \\ e_{21} + f_{21} & e_{22} + f_{12} & \dots & e_{2n} + f_{1n} \\ \vdots & & & \vdots \\ e_{m1} + f_{m1} & e_{m2} + f_{m2} & \dots & e_{mn} + f_{mn} \end{pmatrix} \quad (20)$$

• $E-F$:

$$\begin{pmatrix} e_{11} - f_{11} & e_{12} - f_{12} & \dots & e_{1n} - f_{1n} \\ e_{21} - f_{21} & e_{22} - f_{12} & \dots & e_{2n} - f_{1n} \\ \vdots & & & \vdots \\ e_{m1} - f_{m1} & e_{m2} - f_{m2} & \dots & e_{mn} - f_{mn} \end{pmatrix} \quad (21)$$

• $E \cdot *F$:

$$\begin{pmatrix} e_{11}f_{11} & e_{12}f_{12} & \dots & e_{1n}f_{1n} \\ e_{21}f_{21} & e_{22}f_{12} & \dots & e_{2n}f_{1n} \\ \vdots & & & \vdots \\ e_{m1}f_{m1} & e_{m2}f_{m2} & \dots & e_{mn}f_{mn} \end{pmatrix} \quad (22)$$

• $E \cdot / F$:

$$\begin{pmatrix} e_{11}/f_{11} & e_{12}/f_{12} & \dots & e_{1n}/f_{1n} \\ e_{21}/f_{21} & e_{22}/f_{12} & \dots & e_{2n}/f_{1n} \\ \vdots & \vdots & \vdots & \vdots \\ e_{m1}/f_{m1} & e_{m2}/f_{m2} & \dots & e_{mn}/f_{mn} \end{pmatrix} \quad (23)$$

■ 2つの行列 E, F の乗算 $E * F$ は, 通常 of $[E][F]$.

■ 列ベクトル D と行列 E の演算

- 乗算 $G=E \cdot D$ (G は列ベクトル) :

$$[G] = [E][D]$$
$$\begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix} = \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & & & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{pmatrix} \begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} \quad (24)$$

- $D=E^{-1} \cdot G$:

$$[D] = [E]^{-1}[G]$$
$$\begin{pmatrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{pmatrix} = \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & & & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{pmatrix}^{-1} \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_n \end{pmatrix} \quad (25)$$

■ 行ベクトル A と行列 E の演算

- 乗算 $B=A * E$ (B は行ベクトル) :

$$[B] = [A][E]$$

$$(b_1 \quad b_2 \quad \dots \quad b_n) = (a_1 \quad a_2 \quad \dots \quad a_n) \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & & & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{pmatrix} \quad (26)$$

- $A=B/E$:

$$[A] = [B][E]^{-1}$$

$$(a_1 \quad a_2 \quad \dots \quad a_n) = (b_1 \quad b_2 \quad \dots \quad b_n) \begin{pmatrix} e_{11} & e_{12} & \dots & e_{1n} \\ e_{21} & e_{22} & \dots & e_{2n} \\ \vdots & & & \vdots \\ e_{m1} & e_{m2} & \dots & e_{mn} \end{pmatrix}^{-1} \quad (27)$$

■ 転置

• $E' :$

$$\begin{pmatrix} e_{11} & e_{21} & \cdots & e_{m1} \\ e_{12} & e_{22} & \cdots & e_{m2} \\ \vdots & & & \vdots \\ e_{1n} & e_{n2} & \cdots & e_{mn} \end{pmatrix} \quad (28)$$

• $E' :$

$$\begin{pmatrix} e_{11}^* & e_{21}^* & \cdots & e_{m1}^* \\ e_{12}^* & e_{22}^* & \cdots & e_{m2}^* \\ \vdots & & & \vdots \\ e_{1n}^* & e_{n2}^* & \cdots & e_{mn}^* \end{pmatrix} \quad (29)$$

ただし, $e_{kl}^* (k = 1, 2, \dots, m, l = 1, 2, \dots, n)$ は e_{kl} の複素共役を示す.

■全要素がゼロの行列 zeros や 1 の行列 ones

- m 行 n 列の行列: zeros (m, n), ones (m, n)
- m 行 m 列の正方行列: zeros (m), ones (m)

■単位行列 eye (m)

- 対角行列 $\text{diag}(A)$ ただし, A はベクトルを示す.
- 定義済の行列 E の行数 m , 列数 n $[m, n] = \text{size}(E)$
- 逆行列 行列 $[E]$ の逆行列 $[F] = [E]^{-1}$ は, $F = \text{inv}(E)$
- 行列式 行列 $[E]$ の行列式は, $d = \text{det}(E)$

■プログラム (M1_ex10.m) 行列の簡単な計算例

```
function M1_ex10
format short g

E = [11.0, 12.0; 21.0, 22.0] % 2行2列の行列の設定
F = ones(2) % 要素が全て1の2行2列の行列
EpF = E+F % 行列の和
EmF = E-F % 行列の差
E.' % 行列の転置

Um = eye(2) % 2行2列の単位行列
Eu = E*Um % 行列の積

D = [1.0; 2.0] % 2列の列ベクトルの設定
G = E*D % 行列と列ベクトルの積
D2 = E\G % 逆の計算
Ei = inv(E) % 逆行列の計算
D3 = Ei*G % 逆行列による計算
DG = [D, G] % 2つの列ベクトルから行列を生成

A = [1.0, 2.0] % 2行の行ベクトルの設定
B = A*E % 行ベクトルと行列の積
A2 = B/E % 逆の計算
AB = [A; B] % 2つの行ベクトルから行列を生成
```

```
end
```

■ 出力結果

```
E =  
    11    12  
    21    22  
F =  
     1     1  
     1     1  
EpF =  
    12    13  
    22    23  
EmF =  
    10    11  
    20    21  
ans =  
    11    21  
    12    22  
Um =  
     1     0  
     0     1  
Eu =  
    11    12
```

$$D = \begin{matrix} 21 & 22 \\ 1 & \\ 2 & \end{matrix}$$

$$G = \begin{matrix} 35 \\ 65 \end{matrix}$$

$$D2 = \begin{matrix} 1 \\ 2 \end{matrix}$$

$$Ei = \begin{matrix} -2.2 & 1.2 \\ 2.1 & -1.1 \end{matrix}$$

$$D3 = \begin{matrix} 1 \\ 2 \end{matrix}$$

$$DG = \begin{matrix} 1 & 35 \\ 2 & 65 \end{matrix}$$

$$A = \begin{matrix} 1 & 2 \end{matrix}$$

$$B = \begin{matrix} 53 & 56 \end{matrix}$$

$$A2 = \begin{matrix} 1 & 2 \end{matrix}$$

$$AB =$$

1	2
53	56

11 入出力（ワークスペース変数）

■ **save** ワークスペース変数をファイル（拡張子 `mat`）として保存する（指定しなければ、バイナリファイル）。

- `save`（作業ディレクトリにファイル名 `matlab.mat` で保存）
- `save filename`（ファイル名）
- `save filename content`（セーブする変数名）
- `save filename -ascii -double`（テキストファイル、倍精度で保存）

■ **load** ファイル（拡張子 `mat`）からワークスペース変数に読み込む。

- `load`（作業ディレクトリの `matlab.mat` を読み込む）
- `load filename`（ファイル名）
- `load filename content`（ロードする変数名）
- `load -ascii filename`（テキストファイルを読み込む）

■ **スクリプト**（`M1_ex11.m`） ワークスペース変数の入出力の例

```
clear all
x = eye(3);
y = ones(3);
z = x + y.*i;
save MAT1_ex11
```

```
clear all
load MAT1_ex11
```

```
save M1_ex11_y.txt y -ascii
load -ascii M1_ex11_y.txt
```

12 入出力 (テキストファイル)

■ `fopen` ファイルのオープン.

```
fid = fopen('filename', permission_code)
```

- `fid` (ファイル識別子): ファイルがオープンできれば, 2 より大きい整数が返される.
- `filename`: ファイル名
- `permission_code`: 'rt' (読み込み), 'wt' (新規書き込み), 'at' (追加書き込み)

■ `fprintf` 書式付きデータのファイルへの書き込み.

```
fprintf(fid, format, A, ...)
```

- `fid`: オープン文で決められた値, あるいは, 1 にすると標準出力, 2 にすると標準エラー出力となる.

■ **fscanf** 書式付きデータのファイルからの読み込み. 一つの列の最後は, *format* において `\n` で指定する.

- $A = \text{fscanf}(fid, format)$: ファイルの内容をすべて読み込む.
- $A = \text{fscanf}(fid, format, size)$: *size* で列数 n , 行数 m を指定する場合は, $[n, m]$ とする. また, $[n, \text{inf}]$ とすると, ファイルの終わりまでデータを読み込む.

■ **fclose** ファイルのクローズ.

- `fclose(fid)`
- `fclose('all')` : 標準の `fid=0, 1, 2` 以外のすべてファイルをクローズする.

■ 変換キャラクタ

- `s` : 文字列表示 (`%s`, `%ns`)
- `d` : 10進整数表示 (`%d`, `%nd`)
- `f` : 固定整数点表示 (`%f`, `%nf`, `%n.mf`)

- e : 浮動整数点表示 (%e, %ne, %n.me)
- E : 浮動整数点表示 (%E, %nE, %n.mE)
- g, G : 固定整数点表示と浮動整数点表示の簡潔な方

ただし, n は表示桁数, m は少数以下桁数を示す. 複素数については, 実部, 虚部を上
の書式で各々指定すればよい.

■制御文字 (エスケープ文字)

- 改行 : \n
- タブ : \t

■スクリプト (M1_ex12.m) データをファイルから読み込む例

```
clear all
IR = 0;

% ファイルのオープン
fid = fopen('input_M1_ex12.d','rt');

% ファイルの読み込み
while IR==0
```

```
IS= fscanf(fid,'%4d\n',1);
if IS==0
    disp('stop')
    break
else
    fprintf(1,'\nSection%4d\n',IS)
end
switch IS
    case 10
        ip1=fscanf(fid,'%4d',1);
        a1=fscanf(fid,'%8f\n',1);
        b1=fscanf(fid,'%12e\n',1);
        fprintf(1,' ip1=%4d, a1=%8f, b1=%12e\n',ip1,a1,b1);
    case 20
        a2=fscanf(fid,'%8f%8f%8f\n',3);
        b2=fscanf(fid,'%8f%8f\n',2);
        fprintf(1,' a2=%8f %8f %8f\n',a2);
        fprintf(1,' b2=%8f %8f\n',b2);
    case 30
        a3=fscanf(fid,'%8f\n',1);
        fprintf(1,' a3=%8f\n',a3);
    case 0
        IR = 1;
    otherwise
        IR = -1;
end
```

```
end

% ファイルのクローズ
fclose(fid);
```

■入力データ (input_M1_ex12.d) の例

```
10
  1    10.5
      0.5e6
20
  0.1    0.2    0.7
 20.0    3.3
10
  9     9.5
      0.2e8
30
  5.5
0
```

■出力結果

Section 10

```
ip1= 1, a1=10.500000, b1=5.000000e+005
```

```
Section 20
```

```
a2=0.100000 0.200000 0.700000
```

```
b2=20.000000 3.300000
```

```
Section 10
```

```
ip1= 9, a1=9.500000, b1=2.000000e+007
```

```
Section 30
```

```
a3=5.500000
```

```
stop
```

13 2次元グラフィック

■ **plot** 数値データ $Y = f(X)$ あるいは関数のプロット

```
plot(X, Y, ...)
```

■ **スクリプト (M1_ex13.m)** 数値データの plot による作図の例

```
clear all

% ファイルのオープン
fid = fopen('f_s.d','rt');

% ファイルの読み込み
tmp = fgetl(fid); % 1行 (コメント行) 読み込み
tt = fscanf(fid,'%f',[6,inf]); % データ全て読み込み
[m,n]=size(tt);
Freq = tt(1, 1:n); a11 = tt(3, 1:n); a21 = tt(5, 1:n);

% ファイルのクローズ
fclose(fid);

% データのプロット
figure('Name','帯域通過フィルタの伝送特性','NumberTitle','on')
plot(Freq,a11,'r',Freq,a21,'b'); grid on;
```

```

title('Band-pass filter response');
ymin = -60.0; ymax = 0.0;
axis([-inf inf ymin ymax]);
xlabel('f [GHz]'); ylabel('S_{11}, S_{21} [dB]');
legend('S_{11}', 'S_{21}');
f0 = 10.0; fw = 0.5;
text(Freq(1)*1.03, ymin*0.90, ['Center frequency = ', num2str(f0), ...
    ' GHz, bandwidth = ', num2str(fw), ' GHz']);
print -depsc -tiff fig_f_s % eps ファイルの出力

```

■ 数値データファイル (f_s.d) の例

f [GHz]	L [dB]	S11 [dB]	S11 [deg]	S21 [dB]	S21 [deg]
9.00000	-69.68650	-0.00000	155.80584	-69.68695	65.80584
9.00500	-69.44933	-0.00000	155.67124	-69.44978	65.67124
9.01000	-69.21097	-0.00000	155.53518	-69.21142	65.53518
9.01500	-68.97139	-0.00000	155.39764	-68.97184	65.39764
.....					
(省略)					
.....					
10.99000	-64.74271	-0.00000	-152.83364	-64.74316	-62.83364
10.99500	-64.95978	-0.00000	-152.97178	-64.96023	-62.97178
11.00000	-65.17564	-0.00000	-153.10843	-65.17609	-63.10843

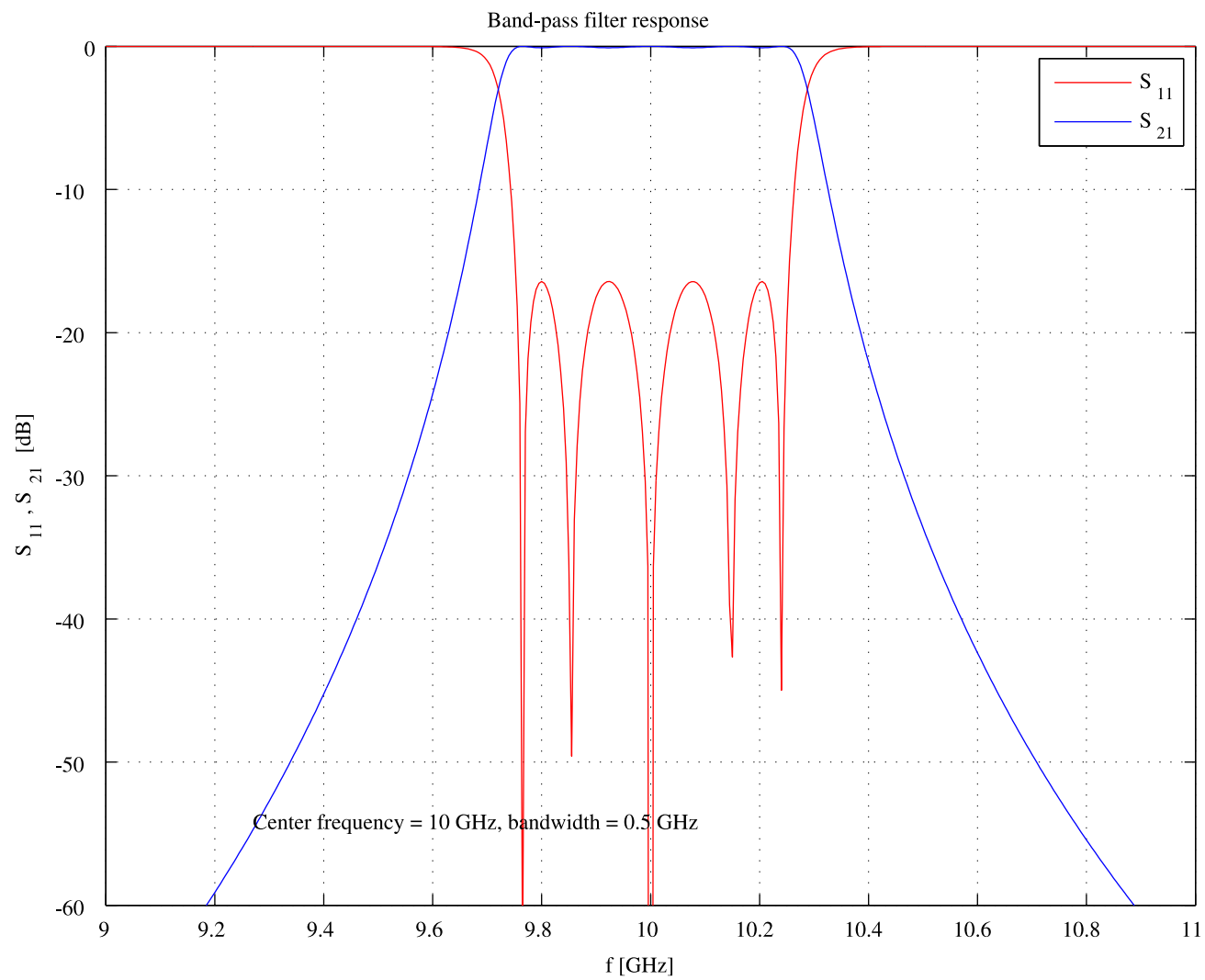


図 1 出力結果 (eps ファイル)

14 3次元グラフィック

■等高線図 数値データあるいは関数 $z = f(x, y)$ の等高線図のプロット

- `contour(X, Y, Z, ...)` : 等高線の表示
- `contourf(X, Y, Z, ...)` : 等高線の表示および等高線間の塗りつぶし

ただし, X, Y は変数 x, y の数値データ (行列), Z は関数 $z = f(x, y)$ の値 (行列) を示す.

■立体図 数値データ $Z = f(x, y)$ あるいは関数の立体図のプロット

- `mesh(X, Y, Z, ...)` : 離散値を結ぶ線のみを色づけする立体図 (ワイヤフレームサーフェス)
- `surf(X, Y, Z, ...)` : 立体表面の塗りつぶし
- `surfc(X, Y, Z, ...)` : 立体表面の塗りつぶし, およびその下に等高線図もプロット

■関連するデータ処理

- $[X, Y] = \text{meshgrid}(x, y)$: 作図のための等間隔 2 次元データ (行列) の作成

■ スクリプト (M1_ex14.m) 2次元数値データの作図の例

```
clear all
x = -6.0:0.1:6; y = x;
nx = length(x); ny = nx;
g = zeros(nx,ny);
for I=1:nx
    for J=1:ny
        r = sqrt(4.0*x(I)^2+y(J)^2);
        if r==0.0
            g(I,J) = 1.0;
        else
            g(I,J) = sin(r)./r;
        end
    end
end
end
[xx,yy] = meshgrid(x,y);

figure('Name','sinc function (contour map)','NumberTitle','on')
[cc,hh]=contourf(xx,yy,g);
title('sinc function (contour map)');
xlabel('x'); ylabel('y'); zlabel('g(x,y)');
axis square; colormap jet; colorbar; caxis([-0.4 1.0]); clabel(cc,hh);
```

```
print -depsc -tiff contour_sinc
figure('Name','sinc function (surface)','NumberTitle','on')
surf(xx,yy,g)
shading interp; axis([-inf inf -inf inf -0.4 1.0]);
title('sinc function (surface)');
xlabel('x'); ylabel('y'); zlabel('g(x,y)');
colormap jet;
print -depsc -tiff surface_sinc
```

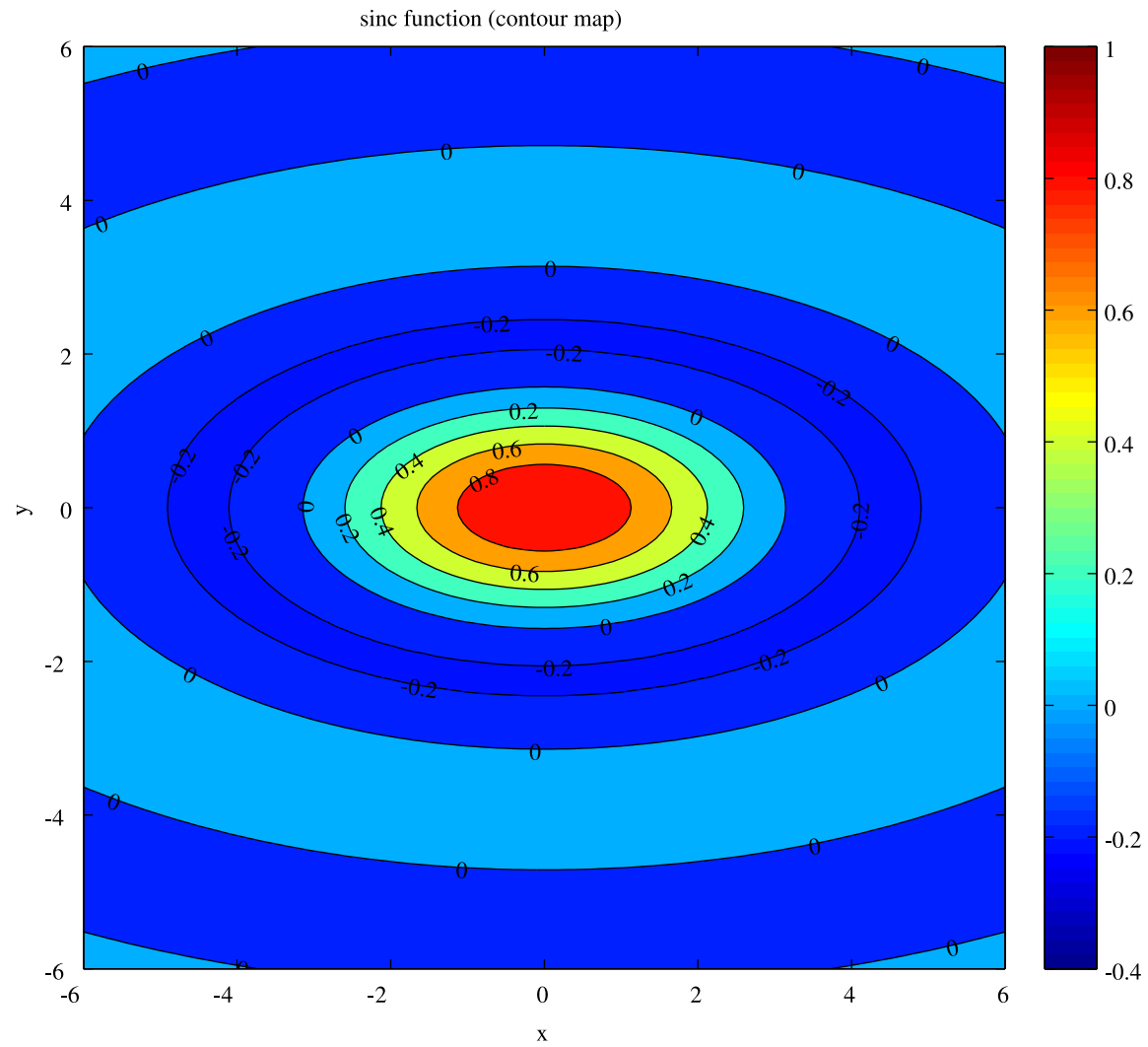


図2 等高線図の出力結果 (eps ファイル)

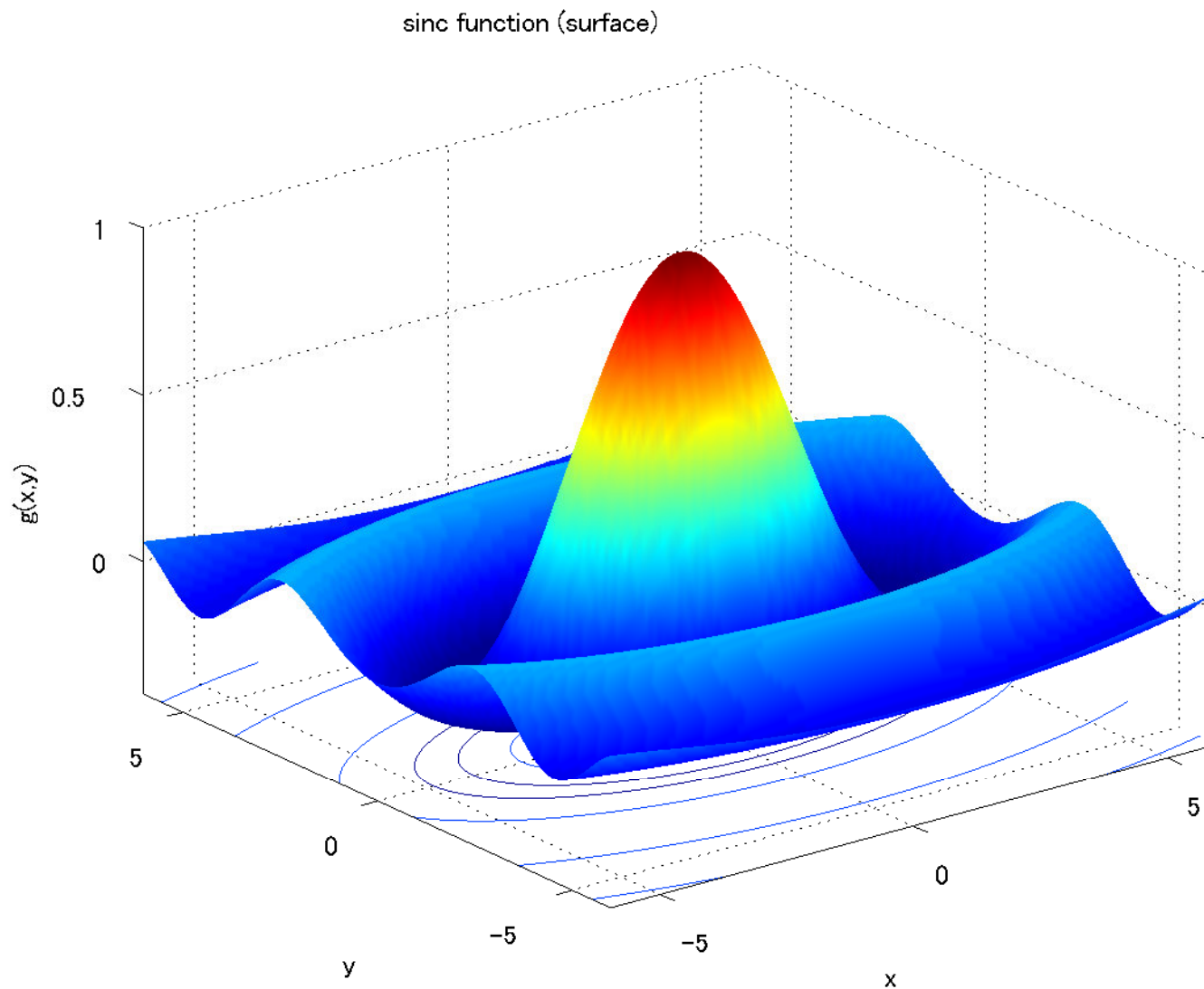


図3 立体図の出力結果 (eps ファイル)

■スクリプト (M1_ex14b.m) MATLAB の sinc のヘルプから一部変更したスクリプト

```
clear all
[X,Y] = meshgrid(-8:.5:8); % 作図のための格子点での座標データ
R = sqrt(X.^2 + Y.^2) + eps; % ゼロ割を避けるためのデータ処理
Z = sin(R) ./R;
mesh(X,Y,Z) % ワイヤフレームサーフェスの作図
```

■スクリプト (M1_ex14c.m) MATLAB の sinc のヘルプから一部変更したスクリプト (ゼロ割回避の別の方法)

```
clear all
[X,Y] = meshgrid(-8:0.5:8);
R = sqrt(X.^2 + Y.^2);
R = R + (R==0)*eps; % ゼロのときだけ eps を加える (if を使わない方法)
Z = sin(R) ./R;
surf(X,Y,Z,'FaceColor','red','EdgeColor','none') % 表面赤色, メッシュ除去
camlight left; % ライティングにおける光源の位置
lighting phong; % ライティングアルゴリズムの選択
```