

『数値計算アプリケーション MATLAB の基礎』

1. 目的

制御や画像処理などのデータ処理に幅広く利用されている、技術系向け数値計算ソフトウェアである MATLAB の基礎を理解する。

2. MATLAB の特徴

技術系向け数値計算言語

C言語などの一般的なプログラミング言語は、その使用法に熟達すればほぼ全ての目的（仕様）を満たすソフトウェアを自作できる。しかしながら、実験データなどの処理といった通常繰り返し行われる一般的（汎用的）作業や、計算対象のモデルを立てその計算を行うといった技術計算には、プログラムの作成（コーディング）に時間・労力をかける余裕がない場合があり、道具としての計算機の利用（コンピュータリテラシー）が必要になる。このような目的を満たすためのソフトウェアのひとつに、MATLABがある。

インタプリタ言語

プログラムの実行時に、逐次命令を解釈しながら実行させる言語。CPUが理解できるアセンブリコードを出力するためのコンパイル処理は不要。変数の値の問い合わせなどが容易で、修正が容易。

行列計算が得意

数値計算を目的に開発された FORTRAN 言語で利用されていた行列計算用の専用ライブラリを利用することができる。このため、高速な行列計算が可能。

可視化が用意

MATLABにもあらかじめグラフ化するツールが用意されており、計算結果を可視化することが容易である。

利用者が多く多くの関連ソフトが豊富

世界中に多くの利用者がいるため、書籍や関連ウェブサイトなどから多くの情報を得ることが可能。また SIMULINK と呼ばれる GUI(グラフィカルユーザインタフェース)環境下で利用可能なソフトウェアと関連付けながら実行することが可能である。

市販ソフトと互換性の高いフリーソフトが存在

MATLABは有償のソフトウェアであるが、基本的機能をほぼ実行可能なフリーソフトウェアである OCTAVE があり、基本技術の習得が無償環境下で行うことができる。

3. 基本的な MATLAB の使用方法

3.1 MATLAB の起動

Windows 起動後、画面上に MATLAB が起動すると図1のような画面が表示される。

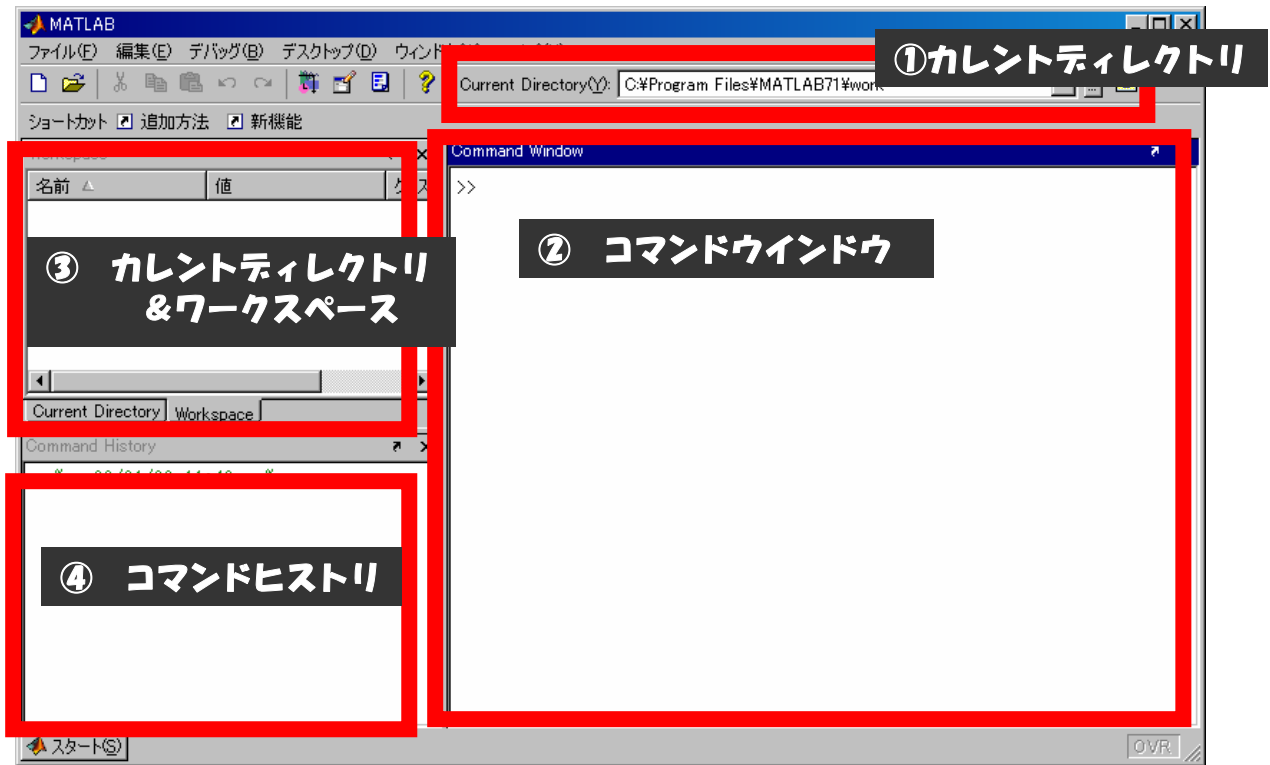


図1 MATLAB画面の例

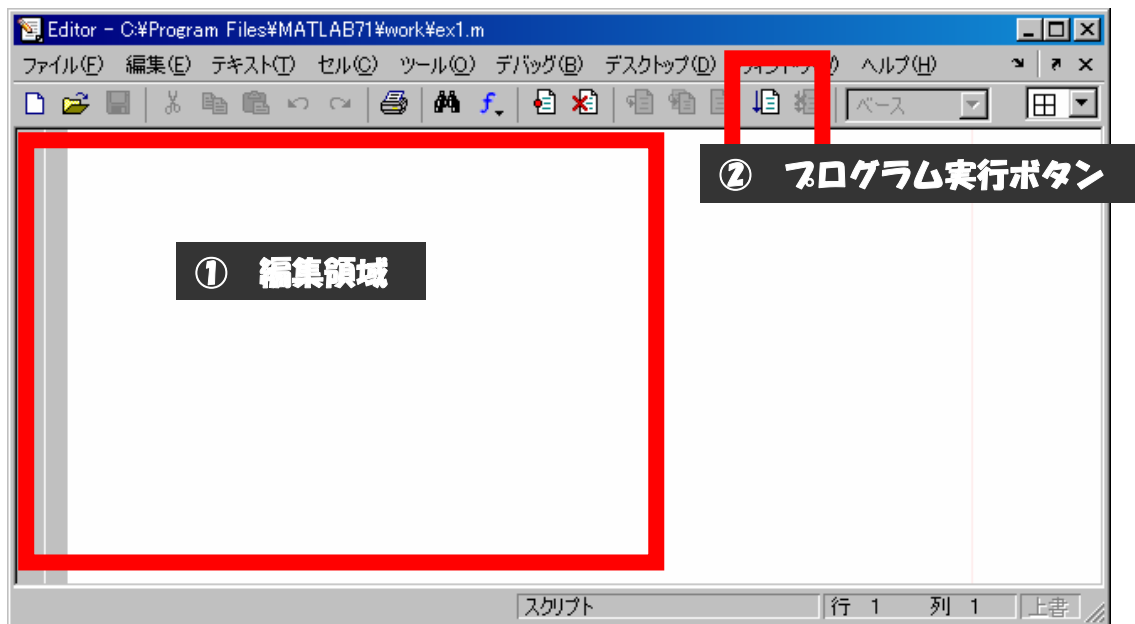


図2 MATLAB エディタの画面の例

3. 2 数値計算の実行（四則演算、複素数の計算）

MATLAB のコマンドウィンドウでは、入力を促す (prompt) 表示に `>>` がある。ここに

変数名 = 値 あるいは 計算式

の形式で入力すると、変数に値が代入される。C言語のように予めデータの型を宣言しなくても使える。虚数は、記号 i を用いれば利用することができる。

(例 1) 実行結果を画面に表示しない。 命令+セミコロン；

```
>> x=2
x =
    2
>> y=3+i*6
y =
    3.0000 + 6.0000i

>> z=3;
```

上の例では、変数 x に値 2 を代入すると、結果が画面に表示されています。>>がない表示は、MATLAB の出力結果。二つ目の命令は、 $y = 3 + i \times 6$ と入力しているので、変数 y に複素数 $3 + i6$ が代入されるが、画面には表示されない。(注：ここで掛け算の記号は、 \times を用いているが、キーボードからはアスタリスク $*$ 記号を用いる。ちなみに MATLAB では複素数は、 i でも j のどちらでも良い。但し、大文字の I は変数とみなされ、虚数とならない。)

(例 2) 変数の値の問い合わせ。 変数名 を入力

```
>> z
z =
    3
```

(例 3) 四則演算、複素共役、絶対値の計算

```
>> x+y
ans =
    5.0000 + 6.0000i
>> x*y
ans =
    6.0000 +12.0000i
>> y*y
ans =
   -27.0000 +36.0000i
>> y'
ans =
    3.0000 - 6.0000i
>> y*y'
ans =
    45
>> abs(y)
ans =
    6.7082
>> abs(y)^2
ans =
    45.0000
```

実数および複素数の四則演算や、予め用意されている関数が利用可能である。複素数の虚数部分の符号を反転させた値を求める命令は、

共役複素数 複素数あるいは変数 + シングル クォーテーション'

この命令を実行することで、上の例では y' を実行すると $3 - i6$ が計算される。複素数と共役な複素数の積を求めると絶対値の二乗が求まる。絶対値を求める関数は次の関数で求められる。

絶対値 `abs()`

3. 3 行列計算の実行（掛け算、逆行列、連立方程式の計算）

（例 4）行列の計算

```
>> A=[1 2 3; 4 5 6; 7 8 9]
A =
     1     2     3
     4     5     6
     7     8     9
>> 2*A
ans =
     2     4     6
     8    10    12
    14    16    18
>> B=[1 0 1 1 1 1 1 1]
B =
     1     0     1     1     1     1     1     1
>> C=[2 3 0 2 5 1 0 1]'
C =
     2
     3
     0
     2
     5
     1
     0
     1
>> B*C
ans =
    11
```

行列の値の代入は、カギ括弧 [] に値を記入する。セミコロン ; を入れると、次の行の値を入力することになる。上の例では行列 A は、3 行 3 列の正方行列になっている。行列 A に値（スカラー）2 をかけると、各要素がそれぞれ 2 倍となり、結果はやはり 3 行 3 列の行列である。行列 B は、1 行 8 列の行列であるが、このような行方向にのみ成分があるものを一般に行ベクトルと呼ぶ。行列 C は、行列の行と列を入れ替える転置をシングルクォーテーション'で行う。転置を行うと 8 行 1 列の行列となり、これは行方向にのみ値があるので一般に行ベクトルと呼ぶ。行列 B と行列 C を掛け算することはできる。その結果は 1 行 1 列となり値のみのスカラーとなる。

（例 5）逆行列の計算 `inv()`

```
>> D=[1 1 1; 1 2 3; 1 3 6]
D =
     1     1     1
     1     2     3
     1     3     6
>> det(D)
ans =
     1
>> E=inv(D)
E =
     3    -3     1
    -3     5    -2
     1    -2     1
```

```
>> D*E
ans =
    1    0    0
    0    1    0
    0    0    1
```

上の例では、行列 D の逆行列を関数 `inv()` で求めている。求められた逆行列を行列 E に代入し、それを元の行列 D と掛け合わせると、対角成分のみが 1 でその他の値はすべてゼロである単位行列が求まる。このことから、正しく逆行列を計算していることが分かる。逆行列がいつも求められるとは限らず、行列式の値がゼロの場合には、逆行列が求まらない。上の例では、行列式の値は 1 であることが計算 `det(F)` の値からわかる。

(例 6) 数学的に許されない計算の例 1。 階 rank()

```
>> F=[1 1 0; 0 2 2; 1 3 2]
F =
    1    1    0
    0    2    2
    1    3    2
>> det(F)
ans =
    0
>> rank(F)
ans =
    2
>> inv(F)
警告: 行列が特異なため正確に処理できません
ans =
    Inf    Inf    Inf
    Inf    Inf    Inf
    Inf    Inf    Inf
```

上の例では、行列式の値を求めるとゼロであることがわかる。したがってこの行列の逆行列は求められない。

(補足) 階数 rank

上の例の行列 F は見かけ上、三つの異なる行ベクトルあるいは列ベクトルから構成されているように見える。しかし、その基底ベクトルの個数 (階数) を求める命令 `rank()` を実行すると、二階の行列であることが分かる。(例えば、一行目と二行目を足し合わせると、この場合は三行目の値になる。ここでは、意図的にそのような値を選んだ。したがって、見かけ上は三つの列ベクトルからのこの行列は構成されているように見えるが、実際には二つの列ベクトルで他の一つの列ベクトルが従属的に決定付けられる。) このような行列で逆行列を求めることはできないので、`inv()` 関数を実行すると警告メッセージが表示され、見かけ上、無限(infinity)と値が表示される。これは、次に示す割り算でも同様である。

(例 7) 数学的に許されない計算の例 2。 1 ÷ 0

```
>> 1/0
警告: ゼロ割です
ans =
    Inf
```

3. 4 データ列の作成

(例 8) 等間隔目盛り、三角関数

```
>> x=[0:pi/10:pi]
x =
Columns 1 through 8
    0    0.3142    0.6283    0.9425    1.2566    1.5708    1.8850    2.1991
Columns 9 through 11
    2.5133    2.8274    3.1416
>> y=sin(x)
y =
Columns 1 through 8
    0    0.3090    0.5878    0.8090    0.9511    1.0000    0.9511    0.8090
Columns 9 through 11
    0.5878    0.3090    0.0000
```

一定間隔で値が増加する行列を作成するための簡単な表現形式が用意されている。

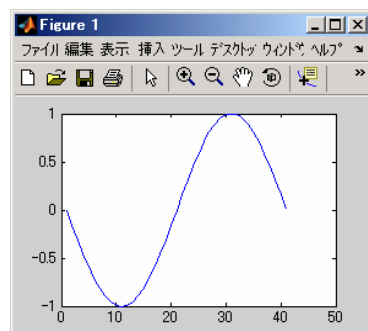
[最小値 : 値の増加幅 : 最大値]

カギ括弧 [] は行列をあらわし、関数への値の引渡し (引数) はマル括弧 () を用いて区別している。技術系の数値計算でしばしば用いられる円周率 π はあらかじめ用意されており、pi で表せば内部では π で計算が行われる。また、計算結果の x は、リストとして扱われる。次に、y に x の Sin を求めた結果が代入される。y も行列になる。出力の行列が画面上の一行で表されない場合には、列 (column) の何番目から何番目を取りあえずこの行に表示するかを示し、続きのデータを次の行以降に表示する。

3. 5 グラフの作成

(例 9) 二次元グラフ

```
>> x=[-pi:pi/20:pi];
>> y=sin(x);
>> plot(x,y)
```



行列 x に代入する右辺のカッコ内は、コロン : で区切られており、

「始まり値」 : 「刻み幅の値」 : 「終わりの値」

データ列を容易に作成することができる。例 8 と同様に、データ列の三角関数を計算するとその結果もデータ列となる。plot() はグラフをプロットする命令。

3. 6 mファイルの作成

MATLABの複数の命令を実行する際、命令文を一行ずつ入力するのは大変である。そこで、Mファイルと呼ばれるファイルにあらかじめプログラムを一括して記述しバッチファイルを作成すると、MATLABのコマンドウィンドウでそのMファイルの名前を入力すると、記述されてあった命令（スクリプト）が実行される。mファイルは、拡張子がmに決められている。p 2の図1. MATLAB画面の例と図2. エディタ画面の例を参照にしながら、以下の手順で行えばmファイルを作成・保存ができる。

mファイル作成の手順

- (1) 作成するファイルを、MyDocumentフォルダに変更する。
- (2) MATLABのコマンドウィンドウから、（ファイル名.m）をMATLABエディタで扱うように命令する。
- (3) mファイルに逐次(順番に)実行する命令を入力し保存する。保存後、エディタの実行ボタンを押すと、MATLAB上でmファイルが実行される。ファンクションキーの（F5）ボタンがショートカット(近道)として用意されているので、これを利用してもよい。

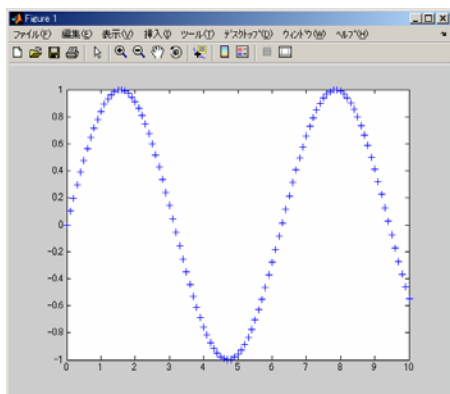
注意：mファイルを初期設定（default）のまま保存すると、パソコンのハードディスクの共有領域のワークスペースへ保存してしまうので、必ず変更すること。

4. MATLABの利用例

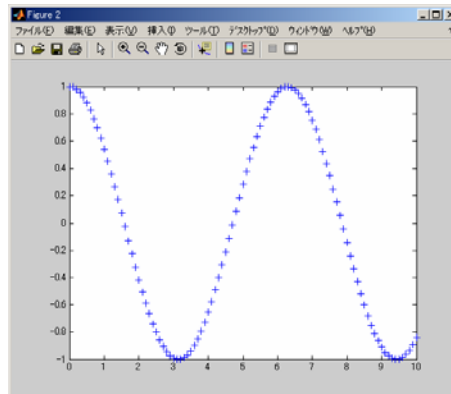
基本

（例10）リサージュ図形のプロット

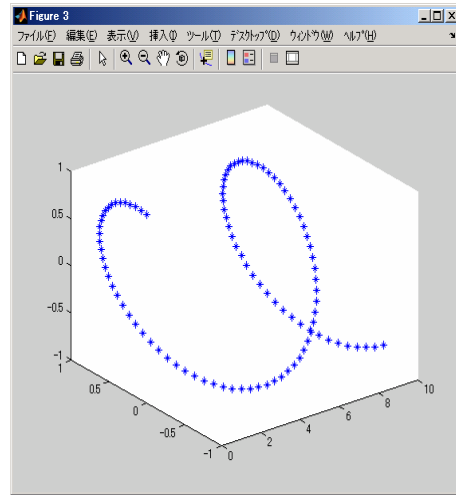
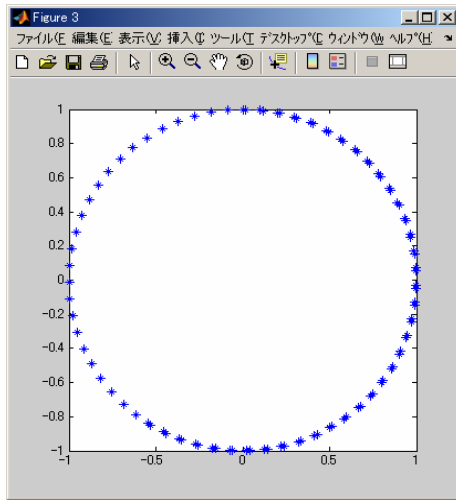
```
t=[0:1/10:10];  
x=sin(t);  
y=sin(t+pi/2);  
figure(1);  
plot(t,x,'+')  
figure(2);  
plot(t,y,'+')  
figure(3);  
plot3(t,x,y,'*')
```



結果1 sin(t)のグラフ



結果2 cos(t)のグラフ



結果3 $\sin(t)$ と $\cos(t)$ の描く軌跡（リサージュ図形）。
時間軸方向から見た2次元グラフ（左）と3次元グラフ（右）。

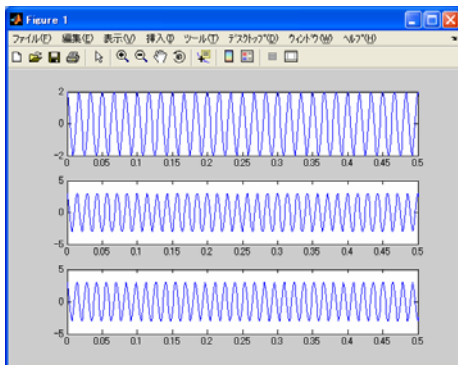
発展

（例 1 1）波の合成と周波数解析

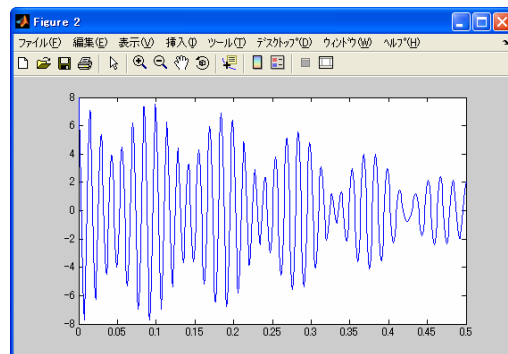
```
clear;

figure(1);
dt=0.001;t=(1:2048)*dt-dt; tend=0.5;
f1=60; f2=70; f3=71;
Amp1=2; Amp2=3; Amp3=3.5;
phi2=0.1; phi3=0.15;
y1=Amp1*cos(2*pi*f1*t);
y2=Amp2*cos(2*pi*f2*t+phi2);
y3=Amp2*cos(2*pi*f3*t+phi3);
y=y1+y2+y3;
subplot(3,1,1);plot(t,y1,'b');xlim([0 tend]);
subplot(3,1,2);plot(t,y2,'b');xlim([0 tend]);
subplot(3,1,3);plot(t,y3,'b');xlim([0 tend]);

figure(2);
plot(t,y,'b');xlim([0 tend]);
```



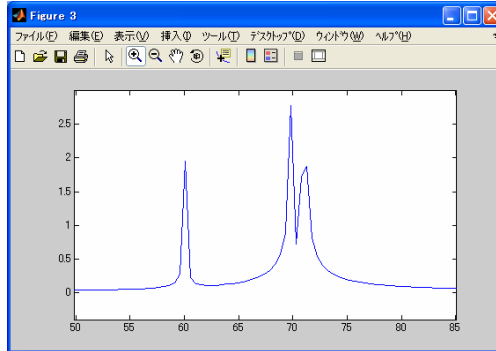
結果4 合成前の3つの波



結果5 足し合わせた合成波

(例 1 2) 高速フーリエ変換

```
figure(3);
ffty=abs(fft(y))/(length(y)/2);
maxfreq=1/2/dt; f=t/dt/dt/length(y);
plot(f, ffty, 'b');xlim([0 maxfreq]);
```



結果 6 合成波形をフーリエ変換した結果

三つの周波数(f)と振幅(amp)と位相(phi)の異なる Sin 波(y1,y2,y3)を足し合わせた合成波(y)のグラフと、関数 ffty()を用いた高速フーリエ変換の結果のグラフ。周波数 f1=60[Hz], f2=70[Hz], f3=71[Hz]が正しく分離されている。

5. 実験課題

次の課題を MATLAB を用いて数値計算を行い、レポートにその実行結果を提出しなさい。

5. 1 課題① 軌跡(リサージュ)

変数 t が変化するとき、次の表で示された関数で与えられる x と y 値を計算し、求められた x の値を水平方向に、y の値を垂直方向にした二次元グラフを描きなさい。

$$x = A \cdot \cos(\omega_A \cdot t + \phi_A)$$

$$y = B \cdot \sin(\omega_B \cdot t + \phi_B)$$

	振幅 A	振幅 B	角速度 omegaA	角速度 omegaB	位相 faiA	位相 faiB
場合 1	1	1	$2 \cdot \pi \cdot 10$	$2 \cdot \pi \cdot 2$	0	0
場合 2	1	2	$2 \cdot \pi \cdot 10$	$2 \cdot \pi \cdot 20$	0	$\pi / 2$
場合 3	2	1	$2 \cdot \pi \cdot 10$	$2 \cdot \pi \cdot 8$	0	$\pi / 4$

実験時間内に上の課題ができた場合、次の発展課題を解きなさい。

5. 2 発展課題① 波の合成

次の波を計算により求めなさい。時間 t の範囲はいろいろ変化させてみて、波形の概要がわかるように適切な代表的な値を選ぶこと。

$$\text{(場合 1)} \quad y = 10 \sin^2(2\pi \cdot 10 \cdot t)$$

$$\text{(場合 2)} \quad y = 10 \sin(2\pi \cdot 10 \cdot t) \cdot \cos(2\pi \cdot 10 \cdot t)$$

$$\text{(場合 3)} \quad y = (10 + 2 \cdot \cos(2\pi \cdot 10 \cdot t)) \cdot \cos(2\pi \cdot 200 \cdot t)$$

$$\text{(場合 4)} \quad y = \sin(2\pi \cdot 10 \cdot t + 6 \cdot \sin(2\pi \cdot t))$$

5. 3 発展課題② 軌跡(リサージュ)

t を 0 から 4 の範囲で変化させたときの様子を描きなさい。
ここで、`round()` は四捨五入された整数値を返す関数である。

$$x = t - \text{round}(t)$$

$$y = \sin(2\pi \cdot x / 0.5)$$

5. 4 発展課題③ 複素数の計算とオイラーの公式

x, y, z が次の式を満たすとする。

$$x = \pi / 30$$

$$y = \cos(x) + i \cdot \sin(x)$$

$$z = y^a$$

実数 a の値が 0 から 120 まで変化したときに、複素数 z の実数部分と虚数部分がどのように変化するかを計算により求め、グラフに描きなさい。

6. レポート

- (1) 実験課題のソースプログラムと実行結果をレポートとして提出すること。
- (2) MATLAB の命令の中で、今回利用しなかった命令を一つ取り上げそれを複数の条件の下で実行し、その命令の概要を説明しなさい。
- (3) MATLAB を用いた感想を書きなさい。
 - Excel と比較してどう思ったか?
 - C 言語と比較してどう思ったか?
 - 便利だと思ったか?
 - 説明は十分か?
 - よくわかったか?
 - 興味がもてたか?

A 1. 付録(課題のヒントと解説)

課題① オシロスコープで観測できる面白図形 “リサージュ波形”

電子回路などで二つの波形の周波数の比と位相差を求める際にしばしば利用されるリサージュ波形の計算。変数 x と y は変数 t の変化を介して変化しているこのような場合に、変数 t を媒介変数と呼ぶ。 t の変化に対して従属的に変化する x と y の値の変化の“アト”を軌跡と呼ぶ。

周波数 f の逆数が周期 T である。波形を計算するためには、一周を適当に分割した刻み幅で計算しないと、波形が表現できない。例えば、100Hz の波形を描くのであれば、一周は 10msec であるので、これを 100 分割すれば計算の刻み幅は 0.1msec となる。

発展課題① 電気回路、電子回路の中の波 「電力」と「変調」

場合 1 : 抵抗の電力。

場合 2 : コイルあるいはコンデンサの電力。

場合 3 : 振幅変調 (AM変調)。

場合 4 : 周波数変調 (FM変調)

発展課題② “反対周りの時計”

これは、半径 1 の単位円と呼ばれる図形が、複素平面で時計と反対方向に回転している。オイラーの公式を実際に数値を代入して計算していることと同じ。

複素数の絶対値と角を求める命令は `abs()` と `angle()` である。また、複素数の実数部分と虚数部分を求める命令は、`real()` と `imag()` である。

A 2. 付録(デモソフトの実行とヘルプファイルの参照)

コマンドプロンプトで `demo` と入力すると、用意されている多くのデモファイルが実行できる。同様にヘルプファイルを実行する場合は、`help` を入力すると実行できる。その他、ウェブサイト上に有用な情報が数多く公開されているので、必要に応じて参考にすることができる。

A 3. 付録(参考になるリンク)

MATLAB 関連ページのリンク集

- MATLAB を用いた教育に関するリンク
(<http://www.cybernet.co.jp/matlab/solution/education/link.shtml>)
- MATLAB ホームページ
(<http://www.cybernet.co.jp/matlab/>)
- MATLAB 日本語ドキュメント
(<http://www.cybernet.co.jp/matlab/support/manual/r14/toolbox/matlab/index.shtml>)
- 行列と線形代数
(http://www.cybernet.co.jp/matlab/support/manual/r14/toolbox/matlab/math/mat_lina.shtml)

電子回路における波に関する解説

- 『波と音のシンセサイザ』
(<http://www.f-kmr.com/synthesizer/no1.htm>)