

第5章 MATLAB の基本操作と行列演算


5.1 実験目的

Matlab とは、MATrix LABoratory(行列実験室)の略語で、その名の通り行列操作を得意とするプログラミング言語である。MATLAB は電子・通信・情報などの様々な理工学分野に利用されているだけでなく、産業分野にも幅広く利用されている。本章では、MATLAB によるシミュレーション実験の方法、特に MATLAB 対話形式の環境における MATLAB の操作やファイルの管理、コマンド・関数の利用、データ行列の演算などの基本機能を習得することを目的としている。

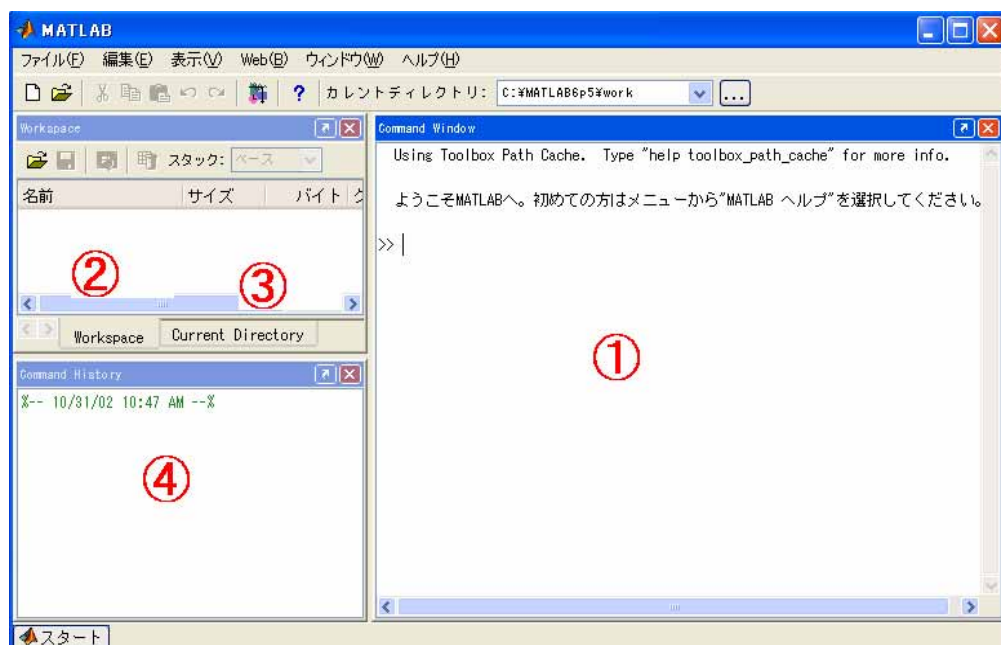
5.2 実験内容

5.2.1 MATLAB の基本操作

1. MATLAB の起動と終了

MATLAB を起動するには、デスクトップのアイコン  をダブルクリックする。MATLAB

が起動されると、次の初期ウィンドウが開かれる。



初期ウィンドウは異なる機能を持つ複数個のサブウィンドウで構成されている。各サブウィンドウの機能は下記の通りである。

- ① Command Window : プロンプト `>>` で始まり、各種コマンド・関数・プログラムを実行するためのサブウィンドウ。
- ② Workspace : 定義された変数の一覧を表示するためのサブウィンドウ。
- ③ Current Directory : フォルダやファイルを閲覧するためのサブウィンドウ。
- ④ Command History : ①で実行したステートメントの履歴を表示するためのサブウィンドウ。
- ⑤ Launch Pad : MATLAB に提供されている各種ツールを表示するためのサブウィンドウであるが、初期状態では表示されていない。表示させる場合には、メニューにある `表示(v)` から選択して他のサブウィンドウと切替えて表示させることができる。

初期ウィンドウの左下にある `スタート` というメニューボタンから各種ツール、デモ、ヘルプを起動させることができる。

Matlabを終了するには、Command Window にコマンド `>> quit` とタイプインする。但し、Matlabを終了させる前に、必要な変数を必ずファイルに保存して下さい。

2. 作業環境とパスの設定

Matlab の作業環境とパスの設定は、原則として参照すべきプログラムをカレントディレクトリに置くことである。この原則のもとで、環境とパスの設定を自由に変更することができる。本実験では次の手順で作業環境とパスを作成する。

(1) 作業用のフォルダの作成

最初の実験日に限り、ハードディスク D: にこれから作業するためのフォルダ `MYWORK` を新規に作成する。フォルダが既にあった場合にはそのまま利用してもよい。但し、フォルダ `MYWORK` のなかに他人のファイルがあれば、速やかに削除すること。

(2) パスの設定

Command Window の上のカレントディレクトリの横にあるパス参照ボタン `<…>` をクリックし、ハードディスク D:にあるフォルダ `MYWORK` とリンクさせる。Matlab が再度立ち上がる際に、カレントディレクトリの中から設定したパスを選択する。

(3) ファイルの保存と削除

本実験終了前に、フォルダ `MYWORK` にあるすべてのファイルを各自が用意したフロッピーディスクか CD に保存し、ファイルの保存を確認した上で、フォルダ `MYWORK` にあるすべてのファイルを削除する。

5.2.2 行列の定義と演算

1. 線形代数における行列の定義

n 行 m 列の行列 A を次式のように定義する。

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = (a_{ij}) \quad (5-1)$$

ここで、 a_{ij} ($i=1, \dots, n; j=1, \dots, m$) は行列 A の要素という。式(4-1)において、 $m=n$ のとき、行列 A を正方行列と言う。

行列 A の要素が 1 行しかない場合には、 A を行ベクトルと言う。行ベクトルは通常次式のようなボールド体(太字)の小文字で記述する。

$$\mathbf{a}_i = [a_{i1} \ a_{i2} \ \cdots \ a_{in}] \quad (5-2)$$

同様に、行列 A の要素が 1 列しかない場合には、 A を列ベクトルと言う。

$$\mathbf{a}_j = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{bmatrix} = [a_{1j} \ a_{2j} \ \cdots \ a_{mj}]^T \quad (5-3)$$

上式にある記号^Tは転置を意味する。式(4-1)において、 $m=1$ 且つ $n=1$ の特別な場合、行列 A の要素は一つだけなので、 A をスカラと言う。

2. MATLAB による行列の定義

MATLAB では、前述のスカラ値、ベクトルと行列データはすべて配列として定義される。即ち、スカラとベクトルはそれぞれ 1 行 1 列の行列、1 行 n 列 (或いは n 行 1 列) の行列として取り扱われる。このような数値行列を定義する際、MATLAB は自動的にデータにメモリを割り当てるので、C 言語のように配列の宣言を行う必要はない。

例題 5_1： 次の数値をキーボードから入力し、実行してみよう。なお、例題に使われている二つのセミコロン<;>の役割に注意。

2 行 3 列の行列

```
>> A = [ 1 2 3 ; 4 5 6 ]  
A =  
  
     1     2     3  
     4     5     6
```

行ベクトル

```
>> b = [ 1 2 3 ];
```

例題 4_1 においては、セミコロン<;>が左右それぞれ 1 箇所に使われていたが、左の例題の<;>は 1 行目の要素と 2 行目の要素を区切りするために使われている。また、右の例題の<;>は Command Window に実行結果を表示しないために使われている。これらを確認するために、例題にあるセミコロン<;>を外して、もう一度実行して、前の結果と比べてみて下さい。

例題 5_2： 転置行列と文字行列を作成しよう。但し、アポストロフィ (apostrophe)<'>の使い方に注意。

行列の転置

```
>> C = [ 1 2 3 ; 4 5 6 ]'  
C =  
  
     1     4  
     2     5  
     3     6
```

文字列の定義

```
>> d = ' I like to use Matlab!'  
d =  
I like to use Matlab!
```

左のアポストロフィ<'>は行列を転置するために使われている。転置の操作 `>>C=A'` で実行しても同様な結果が得られる。それと対照的に右の二つアポストロフィ<' ... '>は文字要素全体を囲んで文字列を定義するために使われている。

3. 行列の保存と読み込み

例題 5_1 と例題 5_2 を実行すると Workspace ウィンドウに変数 A、b、C と d が表示される。これとは別に、Workspace にある変数と属性を確認するためには、コマンド whos で実行してもよい。

```
>> whos
```

Name	Size	Bytes	Class
A	2x3	48	double array
C	3x2	48	double array
b	1x3	24	double array
d	1x21	42	char array

上述の例題から MATLAB では変数名の大文字、小文字を区別されることが判る。

カレントのワークスペース内のすべての変数、或いは一部の変数を保存したい場合には、次のコマンドを実行する。

すべての変数の保存

```
>> save FILENAME
```

変数 A, C のみの保存

```
>> save FILENAME1 A C
```

例えば、上記の右を実行すると、変数 A, C が FILENAME1.mat というバイナリ形式の mat-ファイルに保存される。保存されたファイルは、Command Window から `>> ls` とタイプインして確認することができる。

ワークスペース内の変数の削除、再び変数を読み込みたい場合には、次の操作を行う。

変数 A, C のみの削除

```
>> clear A C
```

すべての変数の削除

```
>> clear
```

mat-ファイルの読み込み

```
>> load FILENAME1
```

4. 行列要素の配列操作

MATLAB では、定義されている行列に対して、要素の抽出、置き換え、結合などの操作が容易にできる。これらの操作の注意すべき共通点としては、行列の要素、ベクトル、子行列の行と列の番号 (i, j) を正確に指定することである。なお、コロン記号<:>を用いて、**行列の初期要素 : 行列の最終要素** のようにすれば、ある特定部分の行列要素を取り出すことが簡潔に行える。

ここで、はじめに定義した 2 行 3 列の整数行列 A を再び例にして、次の演習を行う。

```
>> A
A =
     1     2     3
     4     5     6
```

例題 5_3： 配列 A の要素を次のように抽出してみよう。

2 行 1 列目の要素の抽出

```
>> a21 = A(2, 1)
a21 =
     4
```

1 行目のベクトルの要素の抽出

```
>> a1 = A(1, 1:3)
a1 =
     1     2     3
```

子行列の要素の抽出

```
>> sA = A(1:2, 2:3)
sA =
     2     3
     5     6
```

例題 5_4： 配列 A の要素を次のように置き換えてみよう。

単一の要素の置き換え

```
>> A(2, 2) = 9
A =
     1     2     3
     4     9     6
```

行ベクトルの置き換え

```
>> A(2, :) = 7
A =
     1     2     3
     7     7     7
```

子行列の置き換え

```
>> A(1:2, 2:3)=[8 0;0 8]
A =
     1     8     0
     7     0     8
```

例題 5_5： 鍵括弧 [] を使って配列要素を結合してみよう。

スカラーの結合

```
>> a2 = [a21 a21 a21]
a2 =
     4     4     4
```

ベクトルの結合

```
>> A12= [a1 ; a2]
A12 =
     1     2     3
     4     4     4
```

子行列の結合

```
>> AA=[sA sA']
AA =
     2     3     2     5
     5     6     3     6
```

5. 関数による行列の生成

単位行列のような基本行列の生成は、MATLAB に内蔵されている関数を利用することで行うことができる。このような行列の生成方法は、サイズが大きな行列に対して特に効率的である。下記の表に主な基本行列の関数名とその説明、実行例を簡単にまとめる。

例題 5_6： 行列生成の関数や演算子を利用した例題を実行してみよう。

関数名	関数の説明	実行例
<code>zeros(m, n)</code>	m 行 n 列の零行列の生成	<pre>>> A=zeros(2, 3) A = 0 0 0 0 0 0</pre>
<code>ones(m, n)</code>	m 行 n 列の要素がすべて 1 である行列の生成	<pre>>> A=ones(2, 3) A = 1 1 1 1 1 1</pre>
<code>eye(N)</code>	N 行 N 列の単位行列の生成	<pre>>> A=eye(2) A = 1 0 0 1</pre>
<code>magic(N)</code>	魔法陣の生成：行方向、列方向、対角方向の和が等しくなる行列。但し、N = 1, 3, 4, 5, …	<pre>>> A=magic(3) A = 8 1 6 3 5 7</pre>

		4 9 2
rand(m, n)	一様分布の乱数行列の生成	<pre>>> A=rand(2, 3) A = 0.9501 0.6068 0.8913 0.2311 0.4860 0.7621</pre>
randn(m, n)	正規分布の乱数行列の生成	<pre>>> A=randn(2, 2) A = -1.1465 1.1892 1.1909 -0.0376</pre>
diag	対角行列の要素抽出と対角行列を生成するための演算子	<pre>>> D=diag(randn(2, 2)) D = -1.1465 -0.0376 >> A=diag(D) A = -1.1465 0 0 -0.0376</pre>
: colon	等間隔ベクトルの生成、例えば、 変数名=初期値: 増分: 最終値	<pre>>> t=1:0.5:2 t = 1.0000 1.5000 2.0000</pre>
linspace(x, y, n)	線形等間隔ベクトルの生成 例えば、x から y までの n 点の 等間隔ベクトルを作成する	<pre>>> linspace(0, 100, 3) ans = 0 50 100</pre>
logspace(x, y, n)	対数等間隔ベクトルの生成	<pre>>> logspace(0, pi, 3) ans = 1.0000 1.7725 3.1416</pre>

(上記の関数に関する詳しい説明は `>> help 関数名` で参照すること。)

6. 行列と配列の演算

行列および配列という二つの用語は、一般にしばしば混同して使われる。MATLAB では、行列で定義される数学演算は線形代数で取り扱われる内容で、配列で定義される代数演算は要素単位に行われることである。このことは、加算と減算においては、行列演算も配列演算も常に同じ結果が得られるが、乗算と除算を行う場合には、行列演算と配列演算の結果が異なることもあるので、十分な注意が必要である。

5.3 実験課題

1. 行列 $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ と $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix}$ は正則な非零行列である。行列の各要素 a と b

に対して具体的な数値を各自で定義し、MATLAB を利用して次の演算を求めよ。

数値演算子	演算子の説明	数式演算例
$+$ plus	加算	$A + B = \begin{bmatrix} a_{11} + b_{11} & a_{12} + b_{12} \\ a_{21} + b_{21} & a_{22} + b_{22} \end{bmatrix}$
$-$ minus	減算	$A - B = \begin{bmatrix} a_{11} - b_{11} & a_{12} - b_{12} \\ a_{21} - b_{21} & a_{22} - b_{22} \end{bmatrix}$
$*$ mtimes	行列の乗算	$A * B = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$
$.*$ times	配列(要素毎)の乗算	$A .* B = \begin{bmatrix} a_{11}b_{11} & a_{12}b_{12} \\ a_{21}b_{21} & a_{22}b_{22} \end{bmatrix}$
\backslash mldivide	行列の左除算	$A \backslash B = \text{inv}(A) * B$, 但し、 $\text{inv}(\cdot)$ は逆行列演算子
$.\backslash$ ldivide	要素毎の左除算	$A . \backslash B = \begin{bmatrix} b_{11} \div a_{11} & b_{12} \div a_{12} \\ b_{21} \div a_{21} & b_{22} \div a_{22} \end{bmatrix}$
$/$ mrdivide	行列の右除算	$A / B = A * \text{inv}(B)$, 但し、 $\text{inv}(\cdot)$ は逆行列演算子
$./$ rdivide	要素毎の右除算	$A ./ B = \begin{bmatrix} a_{11} \div b_{11} & a_{12} \div b_{12} \\ a_{21} \div b_{21} & a_{22} \div b_{22} \end{bmatrix}$
$^$ mpower	行列のべき乗	$A^2 = A * A = \begin{bmatrix} a_{11}^2 + a_{12}a_{21} & a_{11}a_{12} + a_{12}a_{22} \\ a_{11}a_{21} + a_{21}a_{22} & a_{12}a_{21} + a_{22}^2 \end{bmatrix}$

power	要素毎のべき乗	$B.^3 = \begin{bmatrix} b_{11}^3 & b_{12}^3 \\ b_{21}^3 & b_{22}^3 \end{bmatrix}$
-------	---------	---

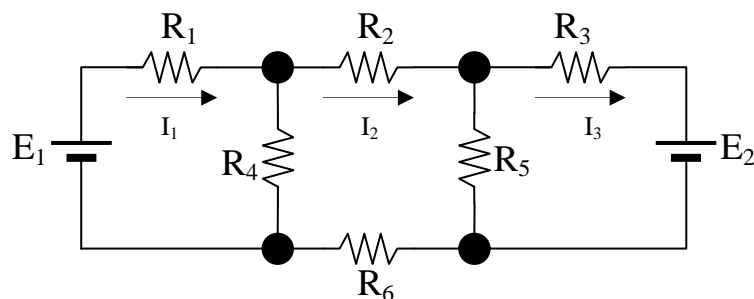
2. 三つの正方行列 A, B, C を次のように定義する。

$$A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} -2 & 1 \\ 4 & -1 \end{bmatrix}, \quad B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} = \begin{bmatrix} -3 & 0 \\ -1 & 2 \end{bmatrix}, \quad C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} 1 & -2 \\ 3 & -4 \end{bmatrix}$$

行列 A, B, C の各要素を Matlab の Command Window から入力し、次の課題を求めよ。

- 1) $(A+B)C = AC+BC$ と $C(A+B) = CA+CB$ をそれぞれ手計算し、MATLAB でその計算結果が一致するかを確認せよ。
- 2) AB と BA が等しくないことを MATLAB の数値計算で示せ。また、行列の交換性 $AB=BA$ が満たされる一例を挙げよ。
- 3) $(AB)^T = B^T A^T$, $(A^T)^T = A$, $(A^T A)^T = A^T A$ (T は転置) が成り立つことを数式で証明し、定義した数値を代入して演算せよ。
- 4) MATLAB の行列演算において、① $A.*B \neq A*B$, ② $A \setminus B \neq A/B$ (行列の左除算と右除算) であることを数値例で示せ。

3. Matlabの行列演算の操作方法を用いて次の回路網に流れている電流 I_1, I_2, I_3 を求めよ。但し、回路のパラメータは、 $E_1 = 10$ [V], $E_2 = 5$ [V], $R_1 = 3$ [Ω], $R_2 = 5$ [Ω], $R_3 = 2$ [Ω], $R_4 = 1$ [Ω], $R_5 = 4$ [Ω], $R_6 = 7$ [Ω] とする。



[手順 : ①回路方程式を立てる。②連立方程式 \Rightarrow 行列方程式に置換える。③行列演算]