

1. はじめに

MATLAB は、マトリクス演算を強力的に扱えるようにしたコンピュータ言語です。MATLAB はインタープリタ型のコンピュータ言語であり、コマンドラインから 1 行ずつ入力してその都度実行することが可能です。これより、一つの演算ごとのデータの変化を確認できます。マトリクスや複素数を扱う数値計算は大得意で、普通の足し算や掛け算の記号を使ってマトリクス演算をすることが可能です。また、コンピュータ言語としても使用可能であり、ループ・判断・サブルーチンを含む一連のプログラムによって複雑な処理もこなすことができます。しかも、インタープリタ型なので、デバッグは非常に簡単です。実際に使用することを念頭に開発されたという意味では、MATLAB は C 言語に似ているとも言えます。もともとの用途が、C 言語は「システム記述言語」、MATLAB は「数値計算をするためのツール」という点こそ違いますが、ユーザの使いごころ第一に機能を充実してきたため、ちょっと慣れれば、非常に使いやすく、極めて優れた道具になります。本稿では、その基本的操作方法を紹介します。

2. 数値計算の基本

2.1. 基本操作

2.1.1. コマンドラインからの実行

図 1 は、起動直後の MATLAB のウィンドウです。(以降では、Windows 版 MATLAB を用いて説明します。)先にも述べたように、MATLAB はマトリクス演算を得意としています。例えば、2 行 2 列の行列の足し算を実行させてみます。行列の定義は、コマンドラインに図 2 (1 ~ 3 行目) のように入力します。このように、列方向は、要素をスペースで区切って並べます。行方向は、各行ベクトルの間を ';' (セミコロン) で区切って並べます。また、行方向は改行によっても表すことができます。命令の最後にもセミコロンがついていますが、これは画面に命令を実行した結果を表示しないことを表すものです。定義した行列の要素を参照するには、図 2 のコマンド (4 行目) のように、“(” と “) ” で参照したい行列の要素番号を指定します。この 2 つの行列 A, B の和を計算し行列 C に代入するには、図 2 (10 行目) のようにコマンドラインに入力します。このように MATLAB では、行列を用いた線形代数の計算が、数式通りに書くだけで簡単に行えます。ここでは、2 行 2 列の行列について説明しましたが、ベクトルやスカラーは行・列の数のどちらかが、または、両方が 1 である特殊な場合と考えられるので、同様にして扱うことができます。

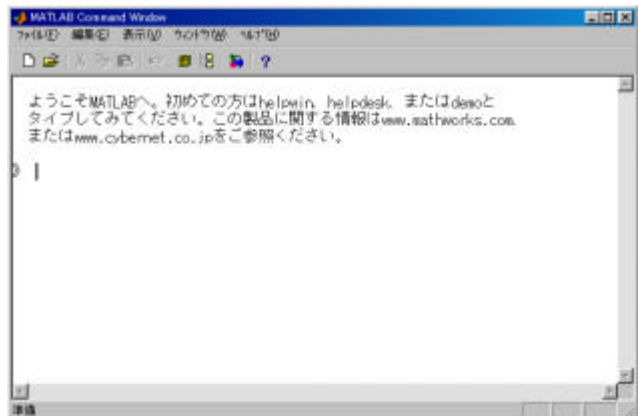


図 1 . 起動直後のウィンドウ

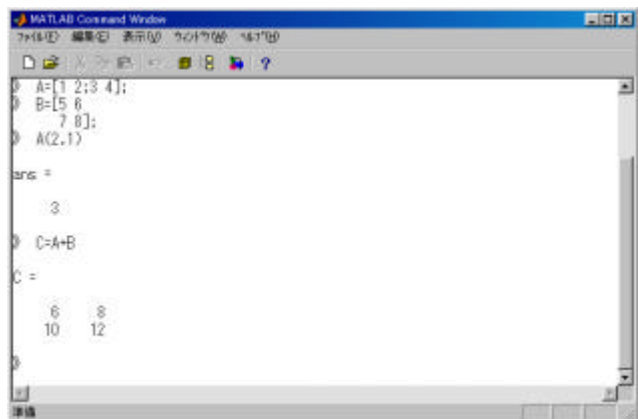


図 2 . 行列の定義

2.1.2. ファイルからの実行

2.1節でコマンドラインから入力したような一連の処理をテキストファイルにして拡張子に `.m` を付けておくと、コマンドラインにファイル名を入力することで、一連の処理を自動的に実行することができます。ちなみに、拡張子が `.m` のファイルを `m-ファイル` と呼びます。

2.2. 行列と配列

MATLAB には行列(matrix)と配列(array)の二つの変数のタイプがあります。

2.2.1. 行列の演算

行列の演算は、テキスト等に記載されている方法と同じように命令を記述することができます。図3に示すように、和・差・積は数式通りに記述します。商については、右からの商と左からの商の演算子が異なります。図4に示すように、右からの商の場合は `/`、左からの商の場合 `\` (バックスラッシュ) を用います。また、行列のべき乗は図4の3つ目の演算のように `^` を用いることで計算できます。

2.2.2. 配列の演算

行演算は行列のものと同じですが、配列の演算は要素毎に演算を行うので、図5に示すように演算子が若干異なります。和と差については同じですが、積は `.*`、右からの商は `./`、左からの商は `.\`、べきの計算は `.^` となります。

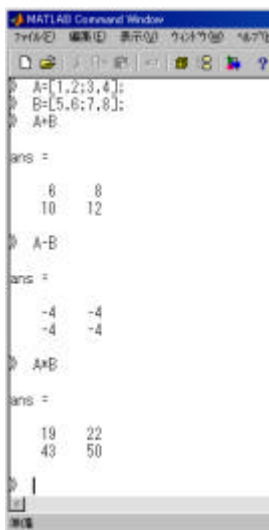


図3 . 行列の和・差・積の計算

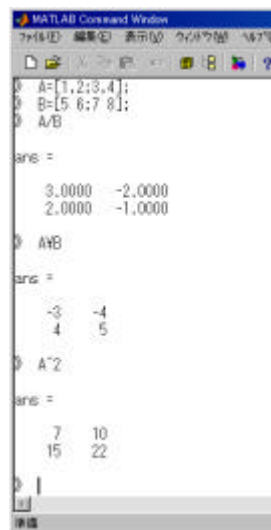


図4 . 行列の商の計算

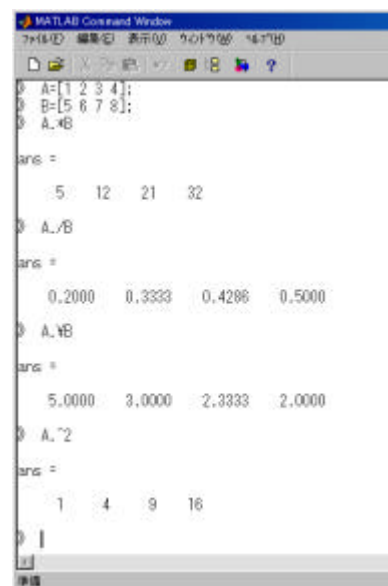


図5 . 配列の演算

2.3. 比較演算と論理演算

表 1 に比較演算子を示します。比較演算子は、同じサイズの行列に対して用い、行列の要素毎に比較した結果を 2 つの行列と同じサイズの行列に出力します。出力は、比較結果が真ならば 1、偽ならば 0 となります。(図 6 , 7 参照)

表 1 . 比較演算子

$A < B$	A は B より小さい	$A \leq B$	A は B より小さいか等しい
$A > B$	A は B より大きい	$A \geq B$	A は B より大きいか等しい
$A == B$	A と B は等しい	$A \sim B$	A と B はひとしくない

表 2 に論理演算子を示します。論理演算子は、同じサイズの行列、または、一方がスカラーに対して用います。“&”演算子は、比較する要素の両方が 0 でない場合に 1 を、そうでない場合は 0 を与えます。“|”演算子は、比較する要素のどちらかが 0 でない場合に 1 を、そうでない場合に 0 を与えます。“~”演算子は、要素が 0 でない場合に 1 を、そうでない場合に 0 を与えます。

表 2 . 論理演算子

$A \& B$	A かつ B (and)	$A B$	A または B (or)
$\sim A$	否定 (not)		

```

>> A=[1 2; 3 4];
>> B=[5 2; 1 4];
>> C=A<B
C =
     1     0
     0     0
>> C=A<=B
C =
     1     1
     0     1
>> C=A>B
C =
     0     0
     1     0

```

図 6 . 比較演算子 (1)

```

>> A=[0 1; 1 0];
>> B=[1 1; 0 0];
>> A&B
ans =
     0     1
     0     0
>> A|B
ans =
     1     1
     1     0
>> ~A
ans =
     1     0
     0     1

```

図 7 . 比較演算子 (2)

```

>> A=[1 2; 3 4];
>> B=[5 2; 1 4];
>> C=A==B
C =
     0     1
     1     1
>> C=A~B
C =
     0     1
     0     1

```

図 8 . 論理演算子

2.4. プログラミング

MATLAB は、C 言語と同様に制御構文をもっています。これにより、MATLAB をプログラミング言語として扱うことができます。

2.4.1. For 文

指定した回数だけ命令文を繰り返すための命令です。for ループの構文を以下に示します。

```
for ループ
for i=1:10
    x(i)=i;
end
```

For 文の構造で C 言語と異なる点は、ループの終端を “end” で表すことぐらいです。上記例では、1 ずつ加算されますが、任意の値ずつ加算する場合は、「for i=1:0.5:10」のように記述します。この場合、1 から 0.5 刻みで 10 まで i の値を加算することになります。

2.4.2. While 文

ある条件を満たしている間、命令文を繰り返すための命令です。while ループの構文を以下に示します。

```
While ループ
n=1;
while prod(1:n)<100
    n=n+1;
end
```

prod(1:n)は、n ! を計算する関数です。for 文と同様、ループの終端を “end” で表します。

2.4.3. If 文

条件によってプログラムの流れを分岐させるための命令です。if 文の構造を以下に示します。

```
if 文
if x<-1
    y=x.^2-1;
elseif (x>=-1)&(x<=1)
    y=1-x.^2;
else
    y= x.^2-1;
end
```

If 文の構造で C 言語と異なる部分は、“elseif” の様に “else” と “if” の間にスペースを入れないことです。

2.5. 関数

MATLAB では、あらかじめ非常に便利な関数が多数用意されています。ここでは、その一部を紹介します。詳しい説明は、help 機能を使えば見ることができます。

2.5.1. 数学で使われる関数

数学で使われる基本的な関数です。ただし、この関数は配列に対して用います。以下に sin 関数を用いた例を示します。

```
sin 関数の例  
  
x=0:pi/10:2*pi;  
y=sin(x);
```

1 行目の命令は、ベクトル x を生成する命令です。ベクトル x の要素は、0 から 2 まで /10 刻みで格納されます。2 行目で、各要素に対して sin 関数により計算をおこない、ベクトル y を生成します。以下は、三角関数に関する関数です。

cos	asin	sinh	atanh
sin	atan	tanh	
tan	atan2	acosh	
acos	cosh	asinh	

そのほかの基本的な関数を以下に示します。

abs	conj	sign	log
angle	round	rem	log10
sqrt	fix	gcd	
real	floor	lcm	
imag	ceil	exp	

2.5.2. いろいろな特殊行列を作る関数

線形代数などで使われる特殊な行列を作る関数です。

inv	逆行列	magic	魔方陣
eig	固有値	invhilb	逆ヒルベルト行列
compam	コンパニオン行列	pascal	パスカルの三角
hadamard	アダマール	hankel	ハンケル行列
daig	対角行列	toeplitz	トープリッツ
kron	クロネッカーのテンソル積	hilb	ヒルベルト行列
gallery	テスト行列	vander	ヴァンデルモンド

応用上便利な行列を、以下の関数により作成することができます。

zeros	零行列	rand	一様分布乱数成分行列
eye	単位行列	logspace	対数スケールで等間隔ベクトル
ones	定数行列	rnadn	正規分布乱数成分行列
linspace	等間隔ベクトル	meshgrid	2 変数関数で使うメッシュグリッド

2.5.3. データ解析に使う関数

データ解析に用いる基本的な関数を以下に示します。

max	最大値	prod	積
min	最小値	cumsum	類和
mean	平均値	cumprod	累積
median	メジアン	diff	差
std	標準偏差	hist	ヒストグラム
sort	ソート	corrcoef	相関係数
sum	総和	cov	分散行列

2.5.4. 多項式に関する関数

MATLABにおける多項式とは、多項式の係数を降べきの順に並べた行ベクトルのことです。以下に、多項式に関する関数を示します。

poly	特性多項式	deconv	デコンボリューションによる 多項式の商
roots	多項式の零点	residue	部分分数分解
conv	合成積による多項式の積	polyval	与えられた点での多項式の値
polyfit	多項式近似		
polyder	多項式の導関数		

2.5.5. 信号処理に関する関数

信号処理に用いる基本的な関数を以下に示します。

abs	複素振幅	deconv	デコンボリューション
angle	位相角	fft	高速フーリエ変換
cov	共分散	ifft	逆高速フーリエ変換
conv	コンボリューション（合成積）		

3. おわりに

本稿では、MATLABを使った基本的な数値計算について説明しました。ここで説明したものは、本当に基本だけですので、興味を持たれた方は参考文献等を読んでください。今回は、数値計算について説明しましたが、MATLABはグラフィックス機能も持っています。次回は、これについて説明したと思います。最後に、先にも述べた様に、MATLABは、「プログラミング+数値計算ツール」ですので、少し慣れれば非常に強力な味方になります。これを機会に、MATLABを使って頂ければ幸いです。

【参考文献】

1. 芦野隆一, Remi Vaillancourt: はやわかりMATLAB, 共立出版株式会社, (1997)
2. 小国力: MATLABの利用と実際, サイエンス社, (1995)