MATLABの使い方

東京大学　橋梁研究室

MATLABとは

技術計算のための高性能言語

特徴

配列が基本的データ要素
変数宣言不要.
対話的システム.
豊富な関数ライブラリ，グラフィックスツール.

使用される分野

数値計算,アルゴリズムの開発,モデル化，シミュレーション,データ解析, GUIアプリケーションの開発,グラフィックス， etc.

MATLABの動かし方 1

コマンドウィンドウにプログラムを打ち込み，リターン

Ex. c1=2

割り当て

c2=3

c3=c1+c2 演算，割り当て

or

M-fileにプログラムを記述して保存，実行

(CやFortranと同様．コンパイル，リンクなど必要ない)

ファイル→ 新規作成 →M-file, プログラムを記述，
例えば test.mと保存，コマンドウィンドウで‘test’，リターン

MATLABの動かし方 2

<u>カレントディレクトリの変更，パスの設定.</u>

カレントディレクトリの変更

カレントディレクトリブラウザ ↓ 　保存したM-fileの存在するディレクトリを指定

サーチパスの追加

‘パスの設定’ dialog box. ↓ 保存したM-fileの存在するディレクトリを指定

MATLABの動かし方 3

M-fileにコマンドを記述

↓ カレントディレクトリ，パスの設定

コマンドウィンドウでファイル名を打ち込み，リターン
(.mは必要ない)

Basic Rule 1
記述

(1)コメント

%　コメントを表す．論理上の，行の終わり.
%以降の記述は無視される.

(2) 結果の非表示

行の最後に「; 」をつける→　結果を非表示

Ex. c1=2;c2=3;

(3) 大文字と小文字

MATLABでは，大文字と小文字の区別する.
ただし，記述するときにはどちらかに統一した方がよい.

Basic Rules 2
行列とベクトルの表現

数学上の表現

$$\{a\} = \{1 \quad 2 \quad 3\}$$

$$\{b\} = \begin{Bmatrix} 1 \\ 2 \\ 3 \end{Bmatrix}$$

$$[A] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

MATLAB上の表現

a=[1□2□3];

b=[1;2;3];

A=[1□2;3□4];

or

A=[1□2;
3□4];

列の区切り :space
行の区切り:semi-colon

Basic Rules 3
四則演算

$$[A] = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad [B] = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$[C] = [A] + [B]$$

$$[C] = [A] \cdot [B]$$

$$[C] = [A]^T$$

$$C_{11} = A_{11} B_{11}$$

$$C_{23} = A_{23} B_{23} \quad \text{etc}$$

A=[1□2;
3□4];

B=[5□6;
7□8];

C=A+B; → 6 8
10 12

C=A*B; → 19 22
43 50

C=A' → 1 3
2 4

C=A.*B; → 5 12
21 32

Basic Rules 4
行列の要素

a=[1□2□3;
4□5□6];

b=[1□2 □ 3□4];

行列aのm行n列成分 : a(m,n)　　Ex. c=a(2,3)→c=6

行列aのm行 : a(m,:)　　Ex. c=a(2,:)→c=[4 5 6]

行列aのn列 : a(:,n)　　Ex. c=a(:,3)→c=[3
6]

ベクトルbの第m成分 : b(m)　　Ex. c=b(3)

Q1: 行列の掛け算

(1) $[A]=\begin{bmatrix} 1 & -2 \\ -2 & 1 \end{bmatrix}$ $[B]=\begin{bmatrix} 5 & 3 \\ 3 & 8 \end{bmatrix}$

のとき，[A]･[B]を計算せよ.

(2) $\{b\}=\{1 \quad 2 \quad 3\}$

のとき，内積 $\{b\}\cdot\{b\}^T$ を計算せよ.

(3) $[D]=\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$

の第2行目ならびに第3列目を抜き出して表示せよ.

組み込み関数1
よく利用する関数のリスト

初等関数 : sin, cos, tan, log, log10, log2, exp, a^b

行列演算 : inv, size, length, ones, zeros, eye

2次元グラフィックス : plot, semilogx, semilogy, loglog, fill

3次元グラフィックス : fill3, surf, mesh

グラフィックスオプション : xlabel, ylabel, title, subplot, figure, axis

ファイル入出力 : fopen, fscanf, fprintf, fclose, save, load

その他 : fft, eig, sort, sum, fix, round, floor

定数: pi, i, j

詳細はヘルプ

Command windowで　　help□コマンド名　　Ex. help□inv

または MATLAB ヘルプ

組み込み関数2
初等関数

```
a=1          b=[1;2]          c=[1 2;
                                 3 4];
```

プログラム		意味	結果
d=sin(a)	→	d=sin(1)	0.8415
d=sin(b)	→	d=[sin(1);sin(2)]	0.8415 0.9093
c=sin(c)	→	d=[sin(1);sin(2); sin(3);sin(4)];	0.8415 0.9093 0.1411 -0.7568

組み込み関数3
行列演算

a=1	b=[1;2;3]	c=[1 2; 3 4];

d=length(b):ベクトルbの長さを計算　　d=3

d=inv(c):行列cの逆行列を計算　　d=[-2 1
1.5 -0.5

d=ones(m,n):m行n列で要素が全部1の行列

d=zeros(m,n):m行n列で要素が全部0の行列

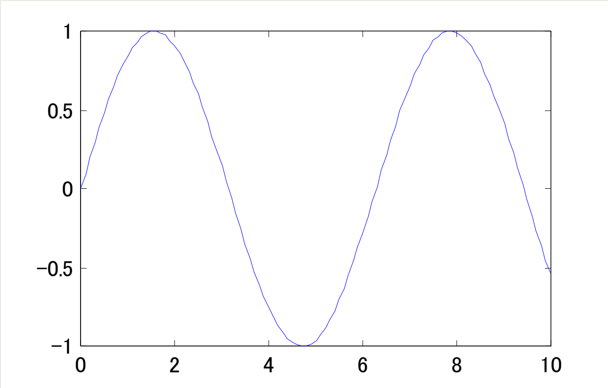
d=eye(m,n):m行n列で対角要素が1，その他が0の行列

組み込み関数4
2次元グラフィックス

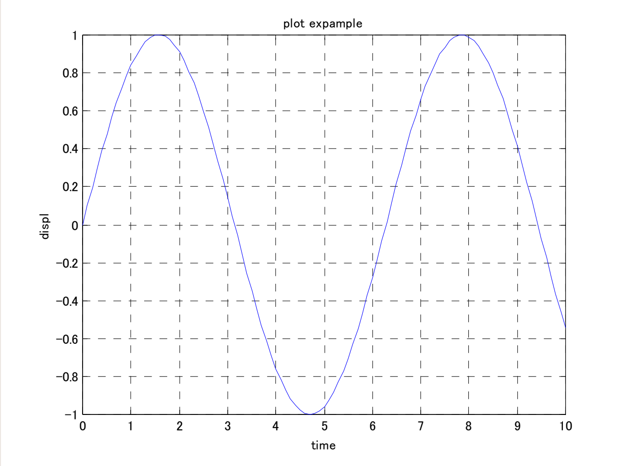
t=t0:dt:t1; → t0～t1までdt刻みのベクトル.

Ex. t=0:1:10 → t=[0 1 2 3 4 5 6 7 8 9 10]

```
x=sin(t)
plot(t,x)
```




```
x=sin(t)
plot(t,x)
xlabel('time');ylabel('displ')
title('plot example')
grid
```



組み込み関数5
固有値問題

[V,D]=eig(A) : 正方マトリクスAの固有値と固有ベクトルを求める.

V: 各列がAの固有ベクトル
行列の大きさはAに等しい.

D: 対角成分がAの固有値，その他の成分は0
行列の大きさはAに等しい.

$$[A]\{v\}=d\,\{v\}$$

$$[A]\cdot[V]=[V]\cdot[D]$$

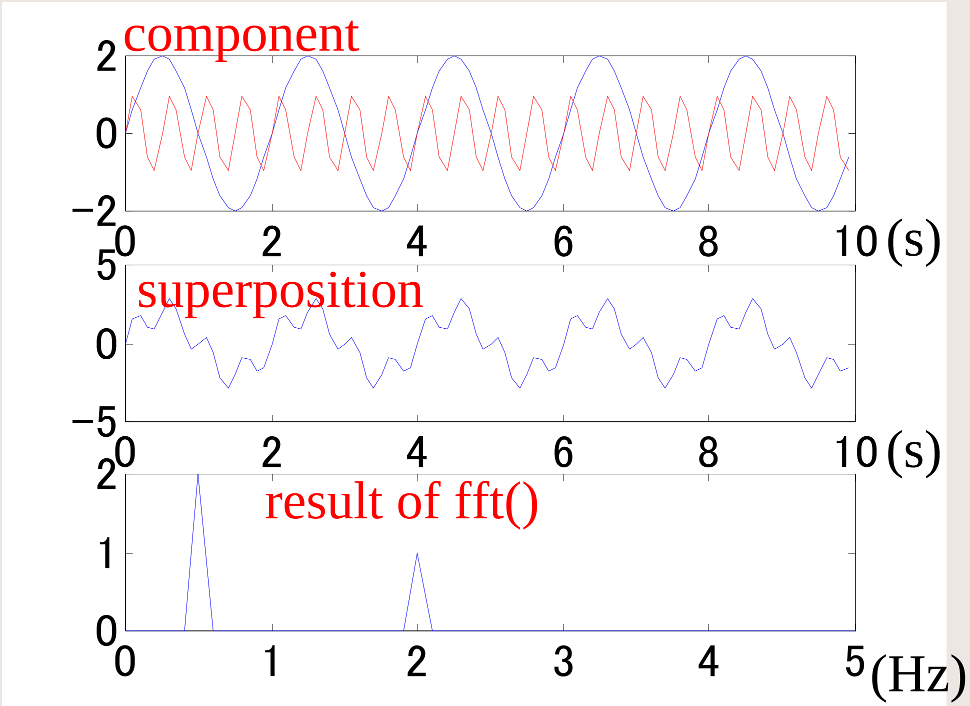
組み込み関数6
フーリエ変換1

c=fft(X): ‘X’の離散フーリエ変換を出力する

周波数領域での解析に有効

Ex.

```
dt=0.1; t=0:dt:10-dt;
f1=0.5; f2=2;
y1=2*sin(2*pi*f1*t);
y2=sin(2*pi*f2*t);
y3=y1+y2;
Y1=fft(y3)/n*2;
```



組み込み関数6
フーリエ変換2

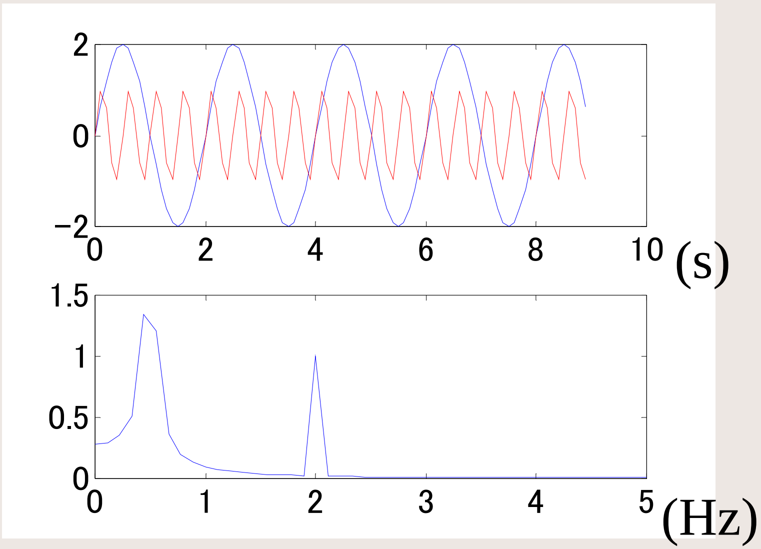
注意1: 見かけ上の性質

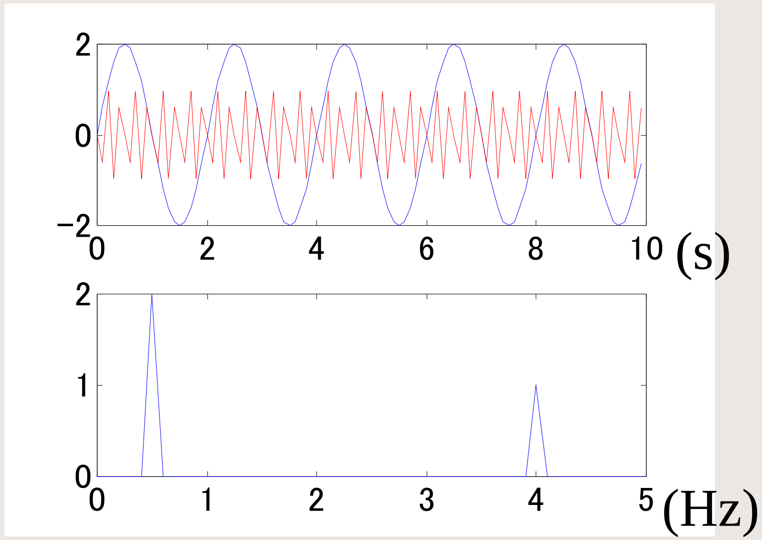
周期性に関する仮定
有限データ長

不十分なサンプリング
周波数　f<1/2/dt

dt=0.1; t=0:dt:9-dt;
f1=0.5; f2=2;

dt=0.1; t=0:dt:10-dt;
f1=0.5; f2=6;





組み込み関数6
フーリエ変換3

注意2: 計算時間

fft(X,n) はn点離散フーリエ変換を計算する.
'n'が2の累乗の場合fft は高速フーリエ変換を行う.

Ex. fft(1:16384) と fft(1:16382)の比較

16384= =2^14

長さ 2^mのデータを使うことが多い.

k=length(X)	X=1,2,…1000
n=2^nextpow2(k)	n=1024

Q2 : sort, sum

(1) helpを利用してsortの意味を理解し,
a={-1 5 3 8}のときsort(a)を計算してベクトルaの要素が並び替えられていることを確認せよ.

(2) helpを利用してsumの意味を理解し,
b={1 2 3 4 5 6 7 8 9 10}のときsum(b)を計算し,
結果が要素の和55になっていることを確認せよ.

Q3：逆行列

逆行列コマンドinvを用いて連立1次方程式を解け.

$$x_1 - 2x_2 + 3x_3 = 6$$
$$2x_1 - 5x_2 + 8x_3 = 16$$
$$x_1 + 2x_2 + 5x_3 = 20$$

答え： $\{x_1, x_2, x_3\} = \{1,2,3\}$

Q4:固有値問題

以下の行列Aの固有値と固有ベクトルを求めよ.

$$[A] = \begin{bmatrix} 1 & 3 & 0 \\ 3 & 1 & 4 \\ 0 & 4 & 1 \end{bmatrix}$$

固有値:1, 6, -4

固有ベクトル:

$$\begin{bmatrix} -0.8 \\ 0 \\ 0.6 \end{bmatrix} \quad \begin{bmatrix} 0.4243 \\ 0.7071 \\ 0.5657 \end{bmatrix} \quad \begin{bmatrix} 0.4243 \\ -0.7071 \\ 0.5657 \end{bmatrix}$$

Q5:フーリエ変換, plot

減衰振動を周波数領域で表示せよ.

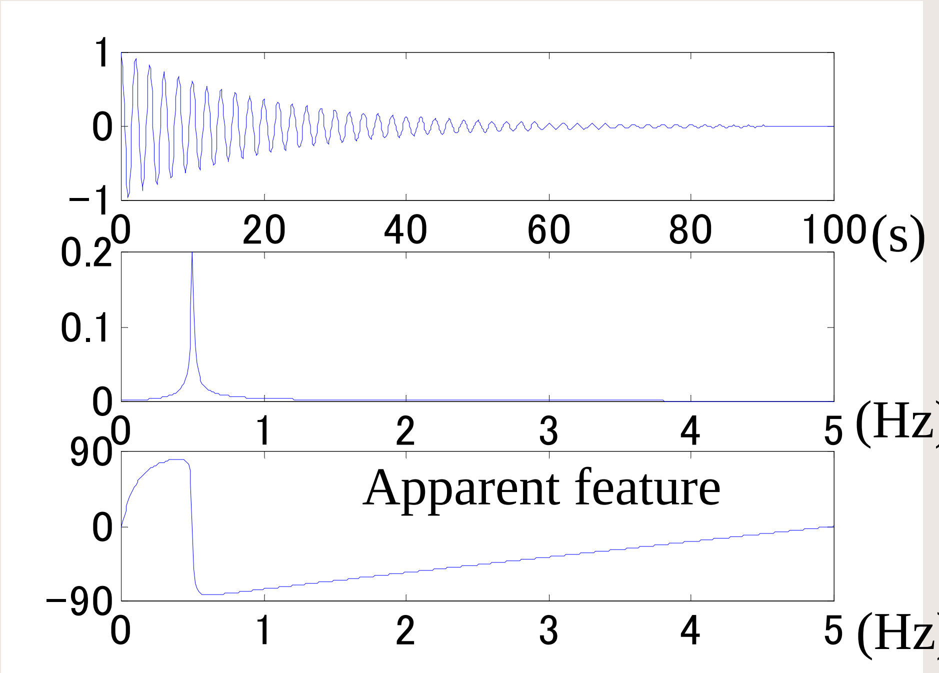
y1=real(exp((i*2*pi*f1-damp)*t));

Y1=fft(y1)/n*2;

時間領域

周波数領域
(振幅)

周波数領域
(位相)



繰り返し計算
for, while

```
for i1=nst:ned
    文
end
```

‘i1’ はnstからnedまで1ずつ増加し
計ned-nst+1回
‘文’
が繰り返される.

Ex.

```
s=0;
for i1=1:100
    s=s+i1;
end
```

1から100までの整数の和

条件文
if, else, else if

```
if  条件1
    文1
elseif 条件2
    文2
else
    文3
end
```

条件1が満たされれば
文1を実行.

条件1 が満たされず
条件2が満たされれば
文2を実行.

条件 1も 2も満たされなければ
文3を実行

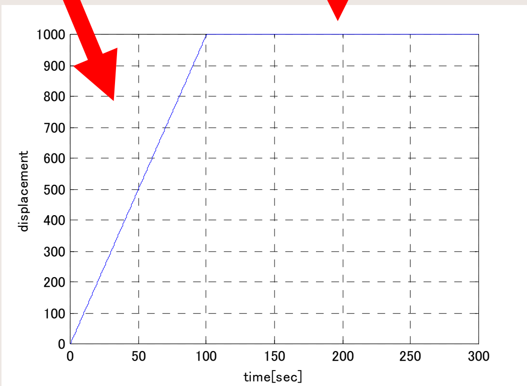
比較演算子
<, >, <=, >=, ==, ~=

論理演算子
&, | , ~

Q6：繰り返し計算，条件文

次のステップ波を得よ.

(i1 <= n1)　　(i1 > n1) is not true




```
t1=100;
t2=200;
vel=1;
dt=0.1;
```

```
n1=fix(t1/dt);
n2=fix(t2/dt);
nn=n1+n2;
tt=0:dt:(n1+n2-1)*dt;
dd=zeros(1,nn);
for i1=1:nn
    if  i1  <= n1
        dd(i1)=vel*(i1-1);
    else
        dd(i1)=dd(n1);
    end
end
plot(t,dd)
xlabel('time[sec]')
ylabel('displacement')
grid
```

fix(): ゼロ方向への丸め

関数の定義，追加 1

自作関数：入力値‘in1’,’in2’,…を受け取り
ユーザが定義した何らかの演算を施し
出力‘out1’,’out2’,…を返す.

・新規M-fileに次の形式で関数を定義する

```
function [out1,out2, …]=func_name(in1,in2,….)

        (ユーザー定義の演算)
        …
        out1=…
        out2=…
        …
```

・関数名と同じ名前で保存(func_name.m)

・カレントディレクトリかサーチパスの下に保存する

関数の定義，追加 2

function out1=extract(inp1,n)
　ベクトルinp1を受け取り，n個刻みのデータout1を出力する

呼び出し側 M-file

```
extract(1:10,2)
```

関数　extract.m

```
function out1=extract(inp1,n)
temp1=1:n:length(inp1);
out1=inp1(temp1);
```

```
ans =

     1     3     5     7     9
```

数値シミュレーション

[y,x]=lsim(A,B,C,D,u,t) は線形システム

$$\begin{cases} \{\dot{x}\} = [A]\{x\} + [B]\{u\} & \text{状態方程式} \\ \{y\} = [C]\{x\} + [D]\{u\} & \text{出力方程式} \end{cases}$$

の応答を任意の入力$\{u\}$に対してシミュレーションする.

<u>Ex. 運動方程式</u>

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{u\}$$

$$\{x\} = \begin{bmatrix} I & [0] \\ [0] & [0] \end{bmatrix} \begin{Bmatrix} \{x\} \\ \{\dot{x}\} \end{Bmatrix}$$

$$\begin{bmatrix} [C] & [M] \\ [M] & [0] \end{bmatrix} \begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix} + \begin{bmatrix} [K] & [0] \\ [0] & [M] \end{bmatrix} \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix} = \begin{Bmatrix} \{u\} \\ \{0\} \end{Bmatrix}$$

出力方程式

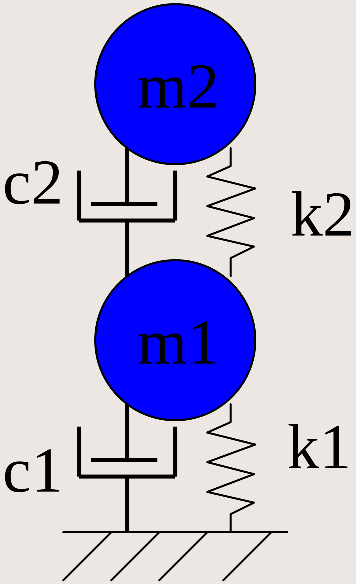
状態方程式

$$\begin{Bmatrix} \dot{x} \\ \ddot{x} \end{Bmatrix} + \begin{bmatrix} [0] & -I \\ [M]^{-1}[K] & [M]^{-1}[C] \end{bmatrix} \begin{Bmatrix} x \\ \dot{x} \end{Bmatrix} = \begin{Bmatrix} \{0\} \\ [M]^{-1}\{u\} \end{Bmatrix}$$

数値シミュレーション

Ex. simulation

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} c_1 + c_2 & -c_2 \\ -c_2 & c_2 \end{bmatrix} \begin{Bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_1 + k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \{0\}$$



dt=0.1;
m1=1;m2=2;
c1=0.1;c2=0.1;
k1=10;k2=15;