



MATLABの使い方

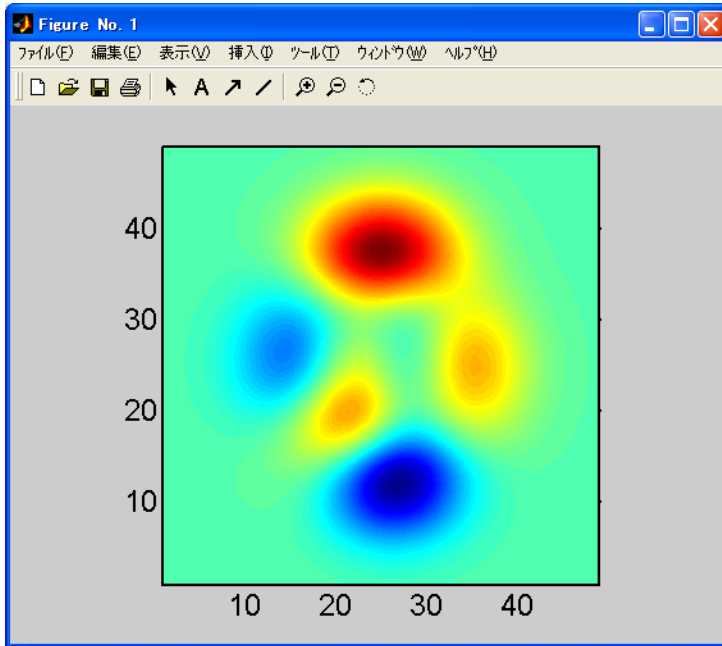
第7回：関数



まとめページ:

<http://www.nuee.nagoya-u.ac.jp/labs/plaene/koukai/purakaku85/tsukaikata/>

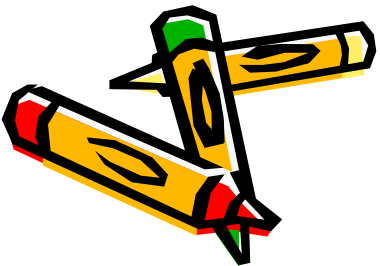
第6回の復習



・グラフの作成、設定、
種類、保存

今回はMATLABに用意済みの関数と、
自分で新しく関数を定義する方法の説明

☆3の内容はひとまず今回まで！
次回以降は少し込み入った内容。



関数の種類

□ 内部MATLAB関数(MATLAB付属の関数)

- ・ 組み込み関数(編集不可)
- ・ M-ファイル関数(toolboxディレクトリ内)

□ ユーザ定義の関数(新しく作る関数、後述)

exist plot

```
>> exist plot
ans =
     5
```

exist surf

```
>> exist surf
ans =
     2
```

EXIST 変数や関数が定義されているかどうかのチェック

EXIST('A') は、つぎの数値を出力します。

- 0 : Aは、存在しません。
- 1 : Aは、ワークスペース内の変数です。
- 2 : Aは、MATLABのサーチパス上のM-ファイルです。また、Aがファイルへのフルパス名、またはAがMATLABのサーチパス上の通常のファイル名の場合にも2を出力します。
- 3 : Aは、MATLABサーチパス上のMEX-ファイルです。
- 4 : Aは、MATLABサーチパス上のMDL-ファイルです。
- 5 : Aは、組み込みのMATLAB関数です。
- 6 : Aは、MATLABサーチパス上のP-ファイルです。
- 7 : Aは、ディレクトリです。
- 8 : Aは、Java クラスです。

組み込み関数の実行は高速

内部MATLAB関数



- 環境変数
- 数学
- プログラミングとデータタイプ

○ ファイルI/O

MATLABの使い方 第5回: データの入出力

<http://hdp.nifs.ac.jp/soken/tanaka/tsukaikata/matlab5.pdf>

○ グラフィックス

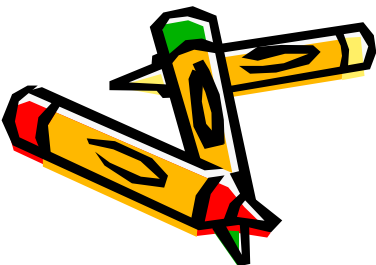
○ 3-D可視化

MATLABの使い方 第6回: グラフィックス機能

<http://hdp.nifs.ac.jp/soken/tanaka/tsukaikata/matlab6.pdf>

○ グラフィカルインターフェース (GUI)

○ 外部インターフェース ← 今回は省略



環境変数

help: 関数の説明

lookfor: キーワード検索

who, whos: カレントの変数の表示

which: 関数の位置の出力

exist: 変数や関数の定義確認

cd: カレントディレクトリの移動

dir, ls: カレントディレクトリのリスト

pwd: カレントディレクトリの表示

edit: エディタの起動

exit, quit: MATLABの終了

clc: コマンドウィンドウのクリア

home: コマンドウィンドウ左上隅に
カーソルを移動

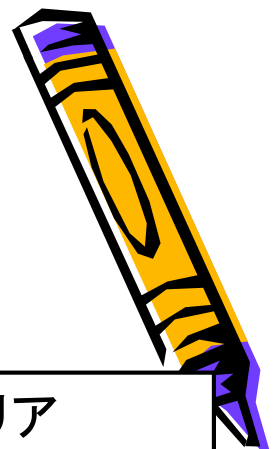
pack: ワークスペースメモリの集約

clear: 変数の消去

path: サーチパスの表示と設定

addpath: サーチパスの追加

genpath: パス文字列を作成



数学

sum: 和

prod: 積

diff: 差分

mean: 平均

std: 標準偏差

var: 分散

abs: 絶対値

sqrt: 平方根

norm: ノルム

mod, rem: 剰余

floor, ceil, round:
小数の丸め

max, min: 最大、最
小値

pi: 円周率

i, j: 虚数単位

NaN: 不定値

Inf: 無限大

cart2sph: 直交座標→球座標

cart2pol: 直交座標→極座標

pol2cart: 極座標→直交座標

sph2cart: 球座標→直交座標

real, imag, conj: 複素数

exp: 指数関数

log, log2, log10: 対数関数

sin, cos, tan: 三角関数

poly, polydar, polyfit, polyval: 多
項式フィッティング

pchip, spline, interp1, interp2,
interp3: 補間

zeros: ゼロの配列

ones: 1の配列

eye: 単位行列

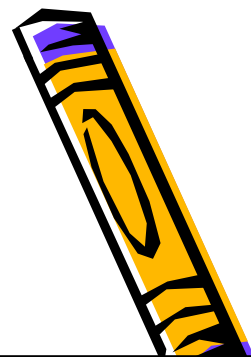
rand: 一様分布乱数

randn: 正規分布乱数

linspace: 線形的に等間隔ベクトル

logspace: 対数的に等間隔ベクトル

プログラミング



length: 長さ

ndims: 次元

size: サイズ

fliplr, flipud, flipdim: 配列の反転

sort: 昇順にソート

repmat: 配列の複製

find: 非ゼロ要素の検出

varargin, varargout: 可変入出力引数

nargin, nargout: 入出力引数の数

if, elseif, else: if文

for: for文

while: while文

break, continue: ループの中止とスキップ

switch, case: switch文

try, catch: try文

==: イコール

<, <=: 超、以上

>, >=: 未満、以下

~=: ノットイコール

error: エラー

warning: 警告

return: 戻る

input: ユーザ入力

disp: 配列の表示

pause: 一時停止

tic, toc: ストップウォッチ

waitbar: ウェイトバー

eval: 文字列の実行

feval: 指定関数の実行

drawnow: 図の更新

&: 論理積

|: 論理和

~: 否定

xor: 排他的論理和

all: 全て1

any: 1が1つ以上

union: 和集合

intersect: 共通部分

setdiff: 差集合



データタイプ



[]: 配列の作成、連結

{ }: セル配列の作成と解除

num2str: 数字→文字列

str2num: 文字列→数字

char: 文字列に変換

double: 倍精度に変換

uint8: 符号無し8ビット整数に変換

upper, lower: 大文字・小文字に変換

strcat, strvcat: 文字列の結合

strcmp: 文字列の比較

findstr: 文字列の検出

blanks: ブランク文字列

isnan: NaN検出

isinf: Inf検出

isempty: 空行列検出

isa: 指定クラスの検出

class: クラスの出力

permute, reshape, squeeze:
配列の次元の再配列、サイズ
変更、次元の削除

global: グローバル変数の宣言

persistent: 永続変数の宣言

mlock, munlock: 変数削除の
防止と解除



ファイルI/O

load, save: MATファイルI/O

dlmread, dlmwrite: ASCII区切り値ファイルI/O

csvread, csvwrite: カンマ区切り値ファイルI/O

xmlread, xmlwrite: XMLファイルI/O

textread: テキストファイル読み込み

fopen, fread, fwrite, fscanf, fprintf, fclose: 低水準I/O

imread, imwrite, imfinfo: 画像ファイルI/O

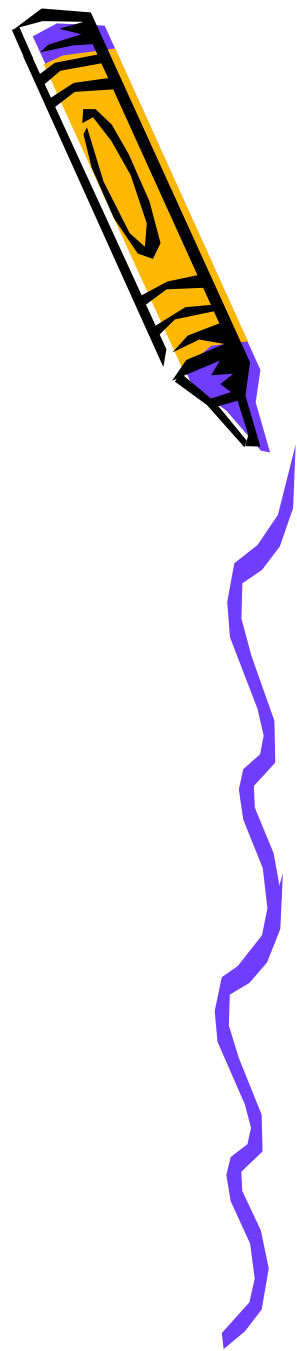
aviread, movie2avi, aviinfo: AVIファイルI/O

wavread, wavwrite, wavinfo: WAVEファイルI/O

wavrecord, wavplay: 録音と再生

VideoReader, VideoWriter: ビデオファイルI/O

sendmail: メールの送信



☆2

グラフィックス/3-D可視化



hold: グラフの保持

grid: グリッドライン

title: タイトル

legend: 凡例

text, gtext: テキストの配置

xlabel, ylabel, zlabel: 軸のラベル

xlim, ylim, zlim: 軸範囲の設定

axis: 軸の範囲と外観

daspect: 縦横比の設定

view: 視点の設定

box: Axesの境界

rotate: オブジェクトの回転

figure: Figureの作成

axes: 任意のAxesを作成

subplot: タイル状にAxesを作成

close: Figureの削除

clf, cla: Figure、Axesのクリア

saveas: Figureの保存

get: オブジェクトプロパティの取得

set: オブジェクトプロパティの設定

reset: オブジェクトプロパティのリセット

findobj, gcf, gca, gco: ハンドル番号の取得

ginput: マウスによるデータ入力

caxis: 色軸のスケールリング

clabel: 標高ラベル

colorbar: カラーバー

colormap: カラーマップの設定

shading: カラーシェーディング

lighting: ライティング

camlight: ライトの作成

alpha: 透明度の設定

plot, ploty, semilogx, semilogy, loglog, stairs, bar, hist, contour, contourf, patch, scatter, polar, area, pie, quiver, fill, image, movie: 2次元プロット

plot3, mesh, surf, surface, contour3, bar3, scatter3, pie3, quiver3, fill3: 3次元プロット

isosurface, isonormals, reducepatch, slice, coneplot, contourslice, streamline, streamparticles: ボリュームデータの可視化



☆2

ユーザ定義関数



○ 1行目に「function 関数名」を宣言することで定義

○ 拡張子はm

例: MATLABエディタで次のように記入して保存

出力

```
function pg=gauss(x,mu,sigma) 入力
pg=1/sqrt(2*pi)*sigma.^-1.*exp(-(x-mu).^2./(2*sigma.^2));
```

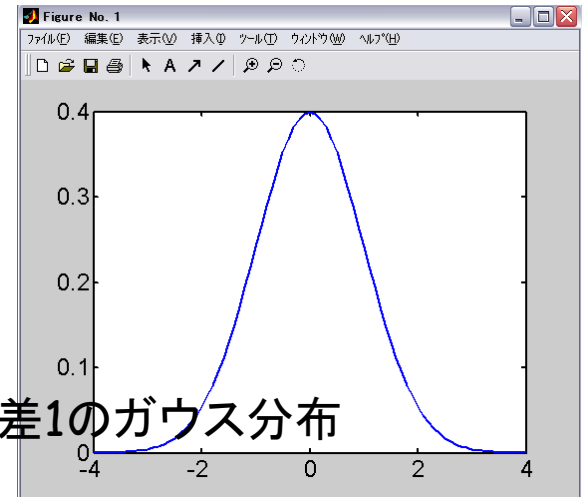
```
1 function pg=gauss(x,mu,sigma)
2
3 - pg=1/sqrt(2*pi)*sigma.^-1.*exp(-(x-mu).^2./(2*sigma.^2));
4
```

コマンドウィンドウで次のように入力すると...

```
>> x=(-4:0.1:4)';
>> plot(x,gauss(x,0,1));
```

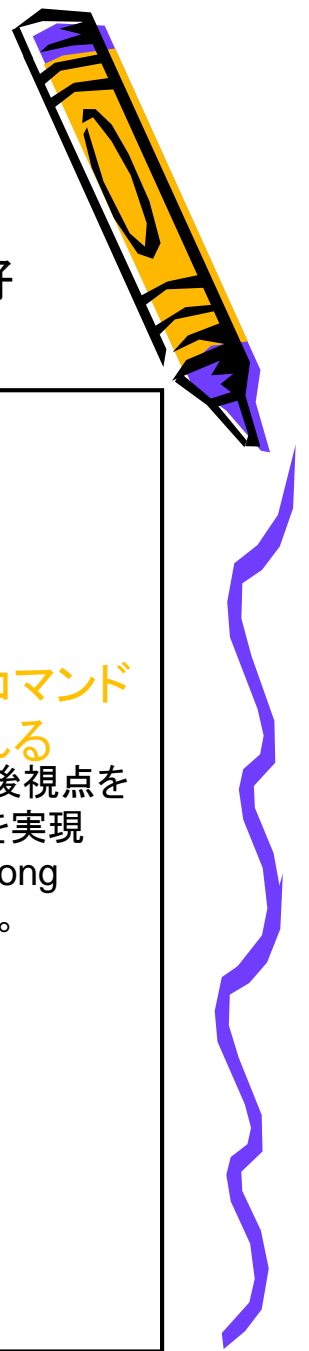


平均値0、標準偏差1のガウス分布



☆1

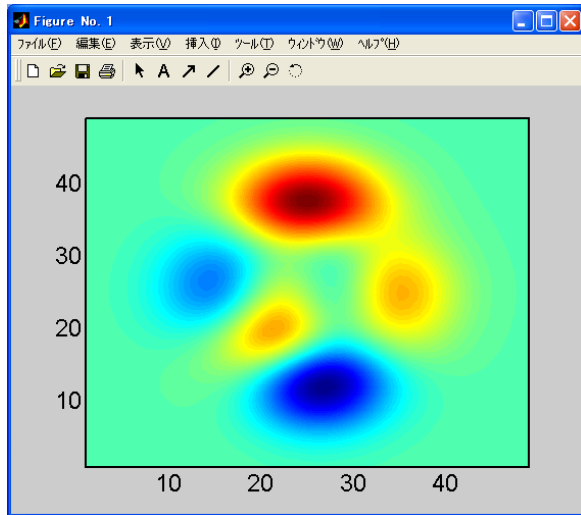
2次元プロット関数の作成



2次元塗りつぶしプロット関数として用意されているcontourfはいろいろと好ましくないことも多いのでsurfプロットを2次元で見る関数を作成してみる

右の枠内をコピーして保存

surf2(peaks)で ↓



```
function h=surf2(varargin)
%-----
% 2D surface plot
%   SURF関数を利用した2次元サーフェスプロット
%                               Data: 2009/12/14
%                               Modified: 2012/01/04
%-----
%                               この部分はHelpコマンド
%                               使用時に表示される
% SURF2(X,Y,Z,C)はSURF(X,Y,Z,C)を実行し、その後視点を
% 2次元の視点へ移すことで2次元サーフェスプロットを実現
% します。またシェーディングを補間、ライティングをphong
% に設定します。さらに軸のスケールリングを変更します。

% Author(s): H. Tanaka (Nagoya University)

hh=surf(varargin{:});
view(2); box on;
shading interp; lighting phong;
axis tight;

if nargout==1, h=hh; end
```

