

MATLAB® 基本関数リファレンス

MATLAB 環境

<code>clc</code>	コマンドウィンドウをクリア
<code>help fun</code>	関数のインラインヘルプを表示
<code>doc fun</code>	関数のドキュメンテーションを開く
<code>load("filename","vars")</code>	<code>*.mat</code> ファイルから変数を読み込む
<code>uiimport("filename")</code>	対話形式のインポートツールを開く
<code>save("filename","vars")</code>	変数をファイルに保存
<code>clear item</code>	ワークスペースから項目を削除
<code>examplescript</code>	<code>examplescript</code> という名前のスクリプトファイルを実行する
<code>format style</code>	出力の表示形式を設定
<code>ver</code>	インストールされているツールボックスの一覧を取得
<code>tic, toc</code>	タイマーの開始および終了
<code>Ctrl+C</code>	現在の計算を中止

演算子および特殊文字

<code>+, -, *, /</code>	行列の算術演算
<code>.*, ./</code>	配列の乗算および除算 (要素単位の演算)
<code>^, .^</code>	行列および配列のべき乗
<code>\</code>	左除算または線形最適化
<code>.' ,'</code>	転置および複素共役転置
<code>==, ~=, <, >, <=, >=</code>	関係演算子
<code>&&, , ~, xor</code>	論理演算 (AND, NOT, OR, XOR)
<code>;</code>	出力の非表示
<code>...</code>	次の行に続ける (... を入力して改行)
<code>% Description</code>	コメント
<code>'Hello'</code>	文字ベクトルの定義
<code>"This is a string"</code>	string の定義
<code>str1 + str2</code>	string の連結

特殊な変数と定数

<code>ans</code>	直前の実行結果
<code>pi</code>	$\pi=3.141592654\dots$
<code>i, j, 1i, 1j</code>	虚数単位
<code>NaN, nan</code>	非数 (ゼロ除算)
<code>Inf, inf</code>	無限大
<code>eps</code>	浮動小数点相対精度

配列変数の定義と変更

<code>a = 5</code>	変数 <code>a</code> に値 5 を定義
<code>A = [1 2 3; 4 5 6]</code> <code>A = [1 2 3 4 5 6]</code>	A を 2 行 3 列の行列として定義 空白で列を区切る ";" または改行で行を区切る
<code>[A,B]</code>	配列の水平方向の連結
<code>[A;B]</code>	配列の垂直方向の連結
<code>x(4) = 7</code>	x の 4 番目の要素を 7 に変更
<code>A(1,3) = 5</code>	A(1,3) を 5 に変更
<code>x(5:10)</code>	x の 5 ~ 10 番目の要素を取得
<code>x(1:2:end)</code>	x の要素を 1 つおきにすべて取得 (1 番目から最後まで)
<code>x(x>6)</code>	6 より大きい要素を一覧表示
<code>x(x==10)=1</code>	条件を使用して要素を変更
<code>A(4,:)</code>	A の 4 行目を取得
<code>A(:,3)</code>	A の 3 列目を取得
<code>A(6, 2:5)</code>	A の 6 行目の 2 番目から 5 番目の要素を取得
<code>A(:,[1 7])=A(:,[7 1])</code>	1 列目と 7 列目を入れ替える
<code>a:b</code>	$a+n \leq b$ のとき $[a, a+1, a+2, \dots, a+n]$
<code>a:ds:b</code>	間隔 <code>ds</code> の等間隔ベクトルを作成
<code>linspace(a,b,n)</code>	等間隔な <code>n</code> 個の値を含むベクトルを作成
<code>logspace(a,b,n)</code>	対数的に等間隔な <code>n</code> 個の値を含むベクトルを作成
<code>zeros(m,n)</code>	要素がすべて 0 の <code>m</code> 行 <code>n</code> 列の行列を作成
<code>ones(m,n)</code>	要素がすべて 1 の <code>m</code> 行 <code>n</code> 列の行列を作成
<code>eye(n)</code>	<code>n</code> 行 <code>n</code> 列の単位行列を作成
<code>A=diag(x)</code>	ベクトルから対角行列を作成
<code>x=diag(A)</code>	行列の対角要素を取得
<code>meshgrid(x,y)</code>	2D および 3D グリッドを作成
<code>rand(m,n), randi</code>	一様分布の乱数の数値または整数を作成する
<code>randn(m,n)</code>	正規分布する乱数を作成

複素数

<code>i, j, 1i, 1j</code>	虚数単位
<code>real(z)</code>	複素数の実数部
<code>imag(z)</code>	複素数の虚数部
<code>angle(z)</code>	位相角 (ラジアン単位)
<code>conj(z)</code>	各要素の複素共役
<code>isreal(z)</code>	配列が実数かどうかを判別

初等関数

<code>sin(x)</code> , <code>asin</code>	正弦および逆正弦 (引数はラジアン単位)
<code>sind(x)</code> , <code>asind</code>	正弦および逆正弦 (引数は度単位)
<code>sinh(x)</code> , <code>asinh</code>	双曲線正弦および逆正弦 (引数はラジアン単位)
以下の三角関数についても同様です。 <code>cos</code> , <code>tan</code> , <code>csc</code> , <code>sec</code> , および <code>cot</code>	
<code>abs(x)</code>	x の絶対値、複素数の大きさ
<code>exp(x)</code>	x の指数
<code>sqrt(x)</code> , <code>nthroot(x,n)</code>	平方根、実数の n 乗根
<code>log(x)</code>	x の自然対数
<code>log2(x)</code> , <code>log10</code>	2 と 10 それぞれを底とする対数
<code>factorial(n)</code>	n の階乗
<code>sign(x)</code>	x の符号
<code>mod(x,d)</code>	除算後の剰余 (モジュロ)
<code>ceil(x)</code> , <code>fix</code> , <code>floor</code>	正の無限大方向、ゼロ方向、負の無限大方向への丸め
<code>round(x)</code>	最も近い小数または整数への丸め

Table

<code>table(var1,...,varN)</code>	変数 var1, ..., varN のデータから table を作成
<code>readtable("file")</code>	ファイルから table を作成
<code>array2table(A)</code>	数値配列を table に変換
<code>T.var</code>	変数 var からデータを抽出
<code>T(rows,columns)</code> , <code>T(rows,["col1","coln"])</code>	T から指定の行と列を持つ新しい table を作成
<code>T.varname=data</code>	T の (新しい) 列にデータを割り当てる
<code>T.Properties</code>	T のプロパティへのアクセス
<code>categorical(A)</code>	categorical 配列を作成
<code>summary(T)</code> , <code>groupsummary</code>	table の概要を出力
<code>join(T1, T2)</code>	共通の変数を使用して table を連結

タスク (ライブエディター)

ライブエディター タスクは、ライブスクリプトに追加して、特定の一連の操作を対話形式で実行できるアプリです。タスクは、一連の MATLAB コマンドを表します。タスクで実行されるコマンドを確認するには、生成されたコードを表示します。

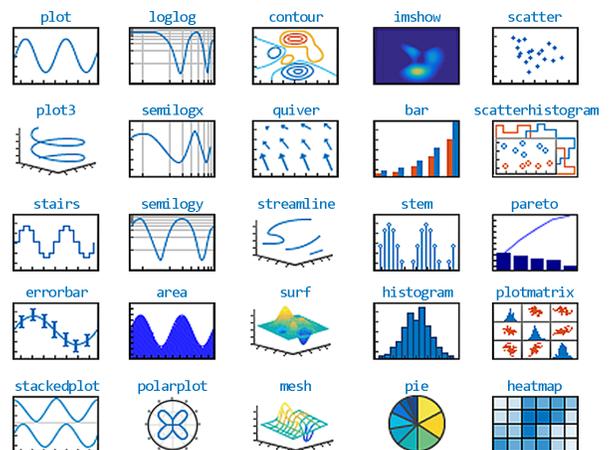
デスクトップでツールストリップの [ライブエディター] タブから利用できる一般的なタスクは、以下のとおりです。

- 欠損データの削除
- 外れ値データの削除
- 変化点の検出
- 局所的極値の検出
- トレンドの除去
- データの平滑化

プロット

<code>plot(x,y,LineStyle)</code>	x に対して y をプロット (LineStyle はオプション) LineStyle は LineStyle 、 マーカー 、および 色 を文字列として組み合わせたものです。 例: <code>"-r"</code> = マーカーを使用しない赤色の実線
<code>title("Title")</code>	プロットタイトルを追加
<code>legend("1st", "2nd")</code>	座標軸に凡例を追加
<code>x/y/zlabel("label")</code>	x/y/z 軸ラベルを追加
<code>x/y/zticks(ticksvec)</code>	x/y/z 軸の目盛りを取得または設定
<code>x/y/zticklabels(labels)</code>	X/Y/Z 軸の目盛りラベルを取得または設定
<code>x/y/ztickangle(angle)</code>	x/y/z 軸の目盛りラベルを回転
<code>x/y/zlim</code>	X/Y/Z 軸の範囲を取得または設定
<code>axis(lim)</code> , <code>axis style</code>	軸の範囲とスタイルを設定
<code>text(x,y,"txt")</code>	テキストを追加
<code>grid on/off</code>	座標軸グリッドを表示
<code>hold on/off</code>	新しいプロットの追加時に現在のプロットを保持
<code>subplot(m,n,p)</code> , <code>tiledlayout(m,n)</code>	タイル表示の座標軸を作成
<code>yyaxis left/right</code>	2 本目の Y 軸を作成
<code>figure</code>	Figure ウィンドウの作成
<code>gcf</code> , <code>gca</code>	現在の Figure を取得、現在の軸を取得
<code>clf</code>	現在の Figure をクリア
<code>close all</code>	開いている Figure を閉じる

一般的なプロットタイプ



プロットギャラリー: mathworks.com/products/matlab/plot-gallery

プログラミング メソッド

関数

```
% 関数を関数ファイルまたはスクリプトファイルの末尾
% に保存します。関数ファイル名はファイル内の
% 最初の関数名と同じにする必要があります
function cavg = cumavg(x) %複数の 引数を指定可能
    cavg=cumsum(vec)./(1:length(vec));
end
```

無名関数

```
% 関数ハンドルを介して定義
fun = @(x) cos(x.^2)./abs(3*x);
```

制御構造

if, elseif 条件

```
if n<10
    disp("n は 10 未満")
elseif n<=20
    disp("n は 10 以上 20 以下")
else
    disp("n は 20 より大きい")
```

switch case

```
n = input("整数を入力: ");
switch n
    case -1
        disp("マイナス 1")
    case {0,1,2,3} % 4 つのケースを一緒にチェックする
        disp("0 以上 3 以下の整数")
    otherwise
        disp("[-1,3] の区間外の整数値")
end % 制御構造は end で終了
```

for ループ

```
% 特定回数のループを実行し
% インデックス変数をインクリメントして
% 各反復を追跡します
```

```
for i = 1:3
    disp("cool");
end % 制御構造は end で終了
```

while ループ

```
% 条件が真である限りループを繰り返します
n = 1;
nFactorial = 1;
while nFactorial < 1e100
    n = n + 1;
    nFactorial = nFactorial * n;
end % 制御構造は end で終了
```

その他のプログラミング/制御コマンド

break	for ループまたは while ループの実行を終了
continue	ループの次の反復に制御を渡す
try, catch	ステートメントを実行し、エラーを受け取る

数値的手法

fzero(fun,x0)	非線形関数の根
fminsearch(fun,x0)	関数の最小値を求める
fminbnd(fun,x1,x2)	区間 [x1, x2] で関数の最小値を求める
fft(x), ifft(x)	高速フーリエ変換と逆高速フーリエ変換

積分および微分

integral(f,a,b)	数値積分 (2 次元、3 次元も同様)
trapz(x,y)	台形則による数値積分
diff(X)	差分および近似微分
gradient(X)	数値勾配
curl(X,Y,Z,U,V,W)	回転および角速度
divergence(X,...,W)	ベクトル場の発散を計算
ode45(ode,tspan,y0)	ノンステップ ODE の求解
ode15s(ode,tspan,y0)	ステップ ODE の求解
deval(sol,x)	微分方程式の解を評価
pdepe(m,pde,ic,... bc,xm,ts)	1 次偏微分方程式を求解
pdeval(m,xmesh,... usol,xq)	PDE の数値解を内挿

内挿および多項式

interp1(x,v,xq)	1 次元内挿 (2 次元、3 次元も同様)
pchip(x,v,xq)	区分的 3 次エルミート多項式内挿
spline(x,v,xq)	3 次スプラインデータ内挿
ppval(pp,xq)	区分的多項式の計算
mkpp(breaks,coeffs)	区分的多項式の作成
unmkpp(pp)	区分的多項式の詳細を抽出
poly(x)	指定された根 x を持つ多項式
polyeig(A0,A1,...,Ap)	多項式固有値問題の固有値
polyfit(x,y,d)	多項式の曲線近似
residue(b,a)	部分分数展開/分解
roots(p)	多項式の根
polyval(p,x)	多項式 p を x の各点で評価
polyint(p,k)	多項式積分
polyder(p)	多項式微分

行列および配列

<code>length(A)</code>	配列の最大次元長
<code>size(A)</code>	配列の次元
<code>numel(A)</code>	配列の要素数
<code>sort(A)</code>	配列要素の並べ替え
<code>sortrows(A)</code>	配列または table の行の並べ替え
<code>flip(A)</code>	配列要素の順序を反転
<code>squeeze(A)</code>	長さ 1 の次元を削除
<code>reshape(A,sz)</code>	配列の形状変更
<code>repmat(A,n)</code>	配列のコピーを繰り返す
<code>any(A), all</code>	任意/すべての要素が非ゼロかどうかを確認
<code>nnz(A)</code>	非ゼロの配列要素の数
<code>find(A)</code>	非ゼロ要素のインデックスと値

線形代数

<code>rank(A)</code>	行列のランク
<code>trace(A)</code>	行列の対角要素の和
<code>det(A)</code>	行列の行列式
<code>poly(A)</code>	行列の特性多項式
<code>eig(A), eigs</code>	行列の固有値およびベクトル (サブセット)
<code>inv(A), pinv</code>	行列の逆行列と擬似逆行列
<code>norm(x)</code>	ベクトルまたは行列のノルム
<code>expm(A)</code>	行列指数
<code>cross(A,B)</code>	クロス積
<code>dot(A,B)</code>	ドット積
<code>kron(A,B)</code>	クロネッカーテンソル積
<code>null(A)</code>	行列のヌル空間
<code>orth(A)</code>	行列の範囲に対する正規直交基底
<code>tril(A), triu</code>	行列の上三角部分および下三角部分
<code>linsolve(A,B)</code>	$AX=B$ 形式の線形システムの求解
<code>lsqminnorm(A,B)</code>	線形方程式の最小二乗解
<code>qr(A), lu, chol</code>	行列の分解
<code>svd(A)</code>	特異値分解
<code>gsvd(A,B)</code>	一般化特異値分解
<code>rref(A)</code>	行列の行簡約階段形

記述統計

<code>sum(A), prod</code>	列に沿った和または積
<code>max(A), min, bounds</code>	最大要素および最小要素
<code>mean(A), median, mode</code>	統計演算
<code>std(A), var</code>	標準偏差および分散
<code>movsum(A,n), movprod, movmax, movmin, movmean, movmedian, movstd, movvar</code>	移動統計関数 n = 移動ウィンドウの長さ
<code>cumsum(A), cumprod, cummax, cummin</code>	累積統計関数
<code>smoothdata(A)</code>	ノイズを含むデータの平滑化
<code>histcounts(X)</code>	ヒストグラムのビンのカウント数を計算
<code>corrcoef(A), cov</code>	相関係数、共分散
<code>xcorr(x,y), xcov</code>	相互相関、相互共分散
<code>normalize(A)</code>	データの正規化
<code>detrend(x)</code>	多項式トレンドの削除
<code>isoutlier(A)</code>	データの外れ値を検出

シンボリック数式*

<code>sym x, syms x y z</code>	シンボリック変数を宣言
<code>eqn = y == 2*a + b</code>	シンボリック方程式を定義
<code>solve(eqns,vars)</code>	指定した変数でシンボリック式を求解
<code>subs(expr,var,val)</code>	式の変数を置き換える
<code>expand(expr)</code>	シンボリック式を展開
<code>factor(expr)</code>	シンボリック式を因数分解
<code>simplify(expr)</code>	シンボリック式を単純化
<code>assume(var,assumption)</code>	変数について仮定を設定
<code>assumptions(z)</code>	シンボリック オブジェクトの仮定を表示
<code>fplot(expr), fcontour, fsurf, fmesh, fimplicit</code>	シンボリック式の関数をプロット
<code>diff(expr,var,n)</code>	シンボリック式を微分
<code>dsolve(deqn,cond)</code>	微分方程式をシンボリックに求解
<code>int(expr,var,[a, b])</code>	シンボリック式を積分
<code>taylor(fun,var,z0)</code>	関数のテイラー展開

*Symbolic Math Toolbox が必要です