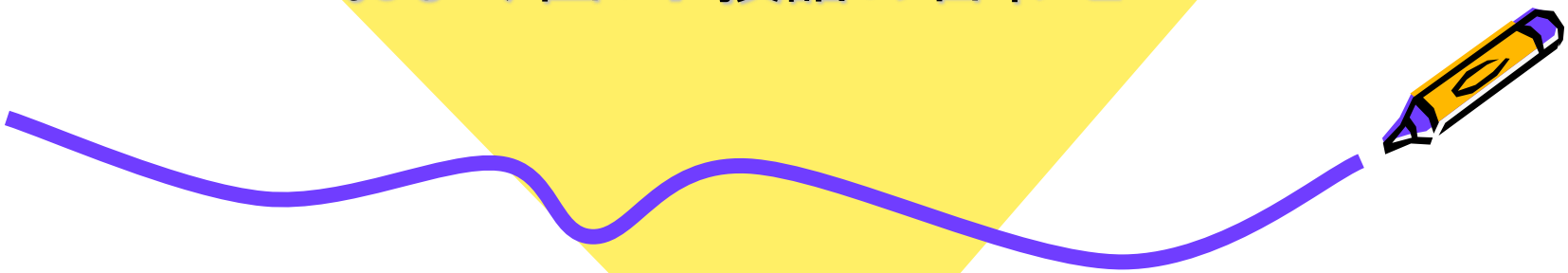
MATLABの使い方

おまけ回：小技詰め合わせ



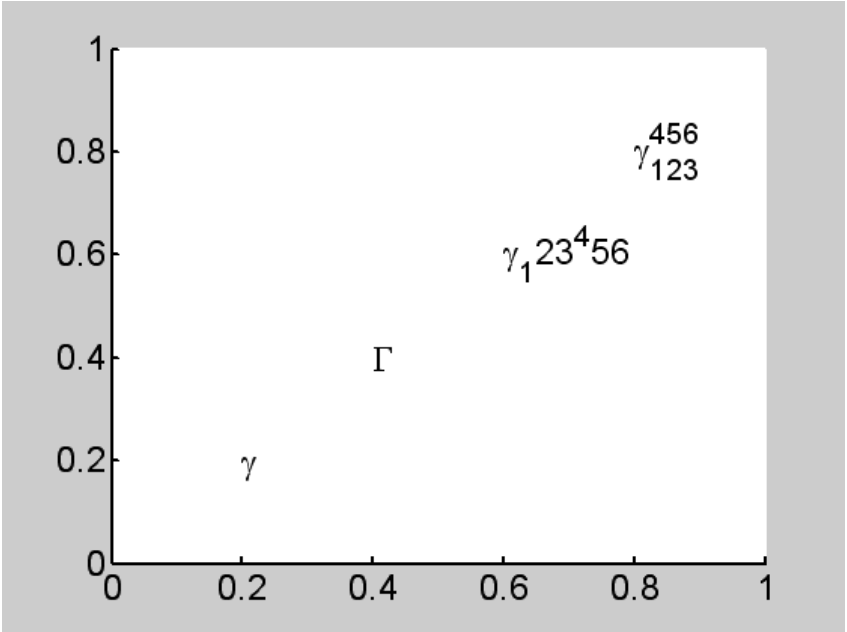
まとめページ：
http://www.nuee.nagoya-u.ac.jp/labs/plaene/koukai/purakaku85/tsukaikata/

☆2

図中の特殊文字

- 「￥（バックスラッシュ）」でギリシャ文字を入力可能
  ↑大文字小文字の変更は先頭文字で
- 「_（アンダーバー）」により下付き文字、「^（ハット）」により上付き文字を入力可能
  ↑{}でまとめて変更、「_」と「^」をそのまま表示したい場合は「￥_」,「￥^」とする

```
>> figure;
>> text(0.2,0.2,'¥gamma');
>> text(0.4,0.4,'¥Gamma');
>> text(0.6,0.6,'¥gamma_123^456');
>> text(0.8,0.8,'¥gamma_{123}^{456}');
```



☆1

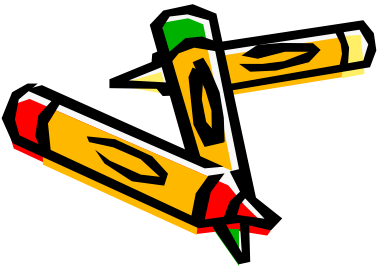
表示桁数を変更する

・MATLABは通常使う数値配列は倍精度で扱われるが、表示桁は小数点以下4桁となる

・表示される桁数を増やしたいときはformat関数を使う

```
disp(pi)
format long; disp(pi*2);
format short; disp(pi*3);
```

```
>> disp(pi)
    3.1416

>> format long; disp(pi*2);
   6.28318530717959

>> format short; disp(pi*3);
    9.4248
```



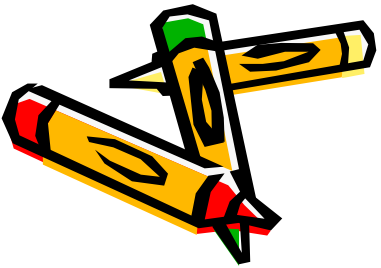
☆2

(')を文字として書く

・シングルクォーテーション(')をキャラクタ配列として変数に代入するとき、普通に「' ' '」と入力するとエラー

・シングルクォーテーションを連続して2つ置くことで1つのシングルクォーテーションとして扱われる

```
>> a='''
??? a='''
      |
エラー: 変数、関数または定数が見つかりません。"incomplete string"が見つかりました。

>> a''''

a =

'
```



☆1

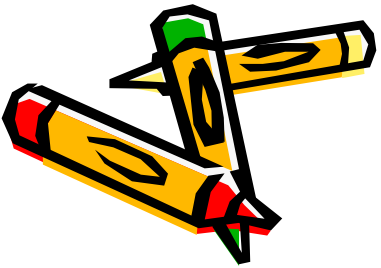
コマンド内容に変数を使用する

・evalコマンドを使うと引数となっている文字列が実行される
・通常の入力ではできない、例えば可変の変数名への代入処理などが実行可能
・シングルクォーテーションを文字列中で使う場合は数に注意
・関数名を可変にするときはfevalコマンドを使用

```
>> help eval

  EVAL    MATLAB表現をもつ文字列の実行

  EVAL(s) は、s が文字列のとき、文字列を式またはステートメントとして実行し
  ます。
```

evalコマンド使用の例

```
>> a='b';
>> eval([a '=1'])

b =
                    b=1
     1

>> eval(['''' a '=1'''])

ans =
                   'b=1'
b=1

>>
```



☆2

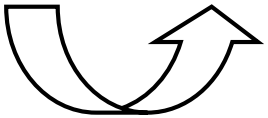
テキストファイルの読み書き

・文字と数字の混ざったテキストファイルを扱うときはtextread
が使えるが、低水準のI/Oコマンドを使用する方が低リスク
・テキスト内容の書き換えにはfscanfとfprintf

全読み込みと全書き込みの例

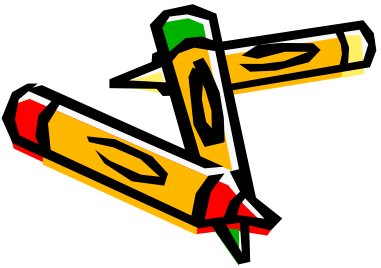
```
fnm='読みたいデータファイル名';
fid=fopen(fnm);
data=fscanf(fid,'%c');
fclose(fid);
```

```
fnm='書きたいデータファイル名';
fid=fopen(fnm,'w');
fprintf(fid,'%c',data);
fclose(fid);
```



ファイル操作

findstr, strcmp, strncmp, strcmpi,
strmatch: 文字列の検出、比較



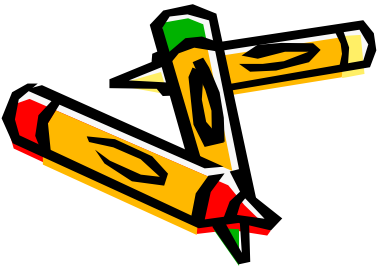
☆1

バイナリファイルの読み書き

・バイナリファイルの読み書きにはfreadとfwrite
・サイズや型式をヘッダファイルなどから予め読み出しておく
・最後はreshapeとpermuteで変数の形を整形する

読み込みの例（高速カメラのrawwデータ）

```
sizea=[256,128]; % width & height
precision='uint16=>double'; % uint16をdouble型へ変換
fnm='読みたいデータファイル名';
fid=fopen(fnm);
img=fread(fid,prod(sizea),precision);
fclose(fid);
img=reshape(img,sizea); % sizeaのサイズに整形
img=permute(img,[2,1]); % 縦と横を入れ替え
```



☆1

dos,unixコマンドの使用

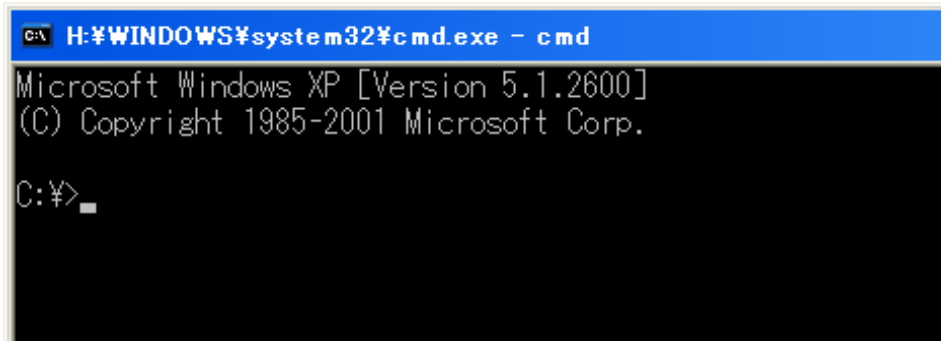
・dos, unix関数によりコマンドプロンプトで動くプログラムをMATLABから実行可能

・文末にアンド(&)を加えることでDOSウィンドウを起動(MATLABの処理は継続)

・アンド(&)を挟むことで複数行のコマンド入力が可能

dosコマンド使用の例

```
>> dos('cmd &');
>> dos('make');
make: *** No targets specified and no makefile found.  Stop.
>> dos('make & exit &');
```



☆1

cygwin環境をMATLABで!

・cygwin(http://www.cygwin.com/)をインストールするとwindowsマシンでunix環境が実現できる

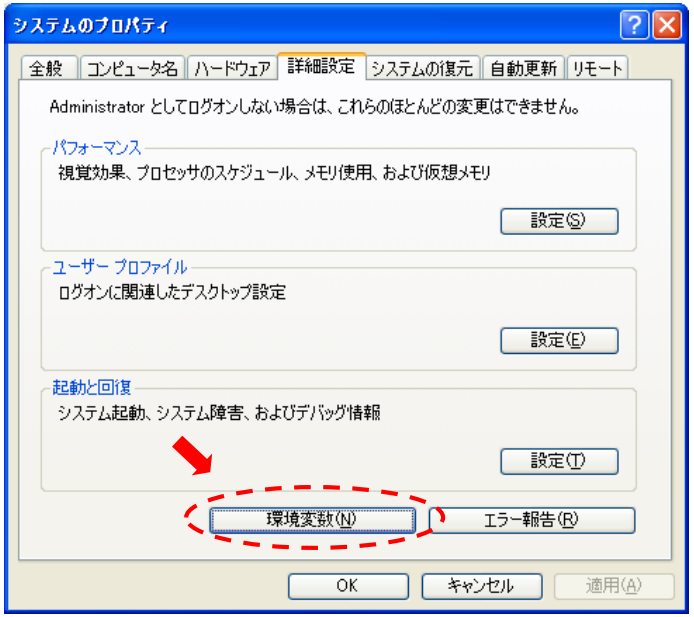
・MATLABのdos, unix関数でunixコマンドを使うときは、cygwinのインストールディレクトリを環境変数に登録しておく必要がある

例えば('C:¥cygwin¥bin')にunixの関数が格納されているときは

```
setenv('PATH',[getenv('PATH') …
';C:¥cygwin¥bin']);
```

↑

コマンド・手動どちらから
設定してもOK →



http://www.mathworks.co.jp/help/ja_JP/techdoc/ref/setenv.html

☆2

可変入出力引数関数の作成

・varargin, varargoutにより入出力引数の数が可変のユーザー定義関数を作成可能

・nargin, nargoutは引数の数決定の参考に使える

可変出力引数をもつ関数の作成例

```
1   function [varargout]=slice_matrix(matrix,dim)
2   % [A,B,...]=SLICE_MATRIX(MATRIX,DIM)は、配列MATRIX=CAT(DIM,A,B,...)を
3   % 次元DIMの方向に分割して(A,B,...)を返します。
4   len=size(matrix,dim);
5   if nargout<len, error('出力引数の数が一致しません'); end
6   for ii=1:len
7       if dim==1
8           varargout{ii}=matrix(ii,:);
9       else
10          eval(['varargout{ii}=matrix(' repmat(':,',1,dim-1) 'ii);']);
11      end
12  end
```

varargin, varargoutはセル配列として取り扱う

```
>> [a,b]=slice_matrix(rand(2,2),2)

a =

    0.6405
    0.2091


b =

    0.3798
    0.7833
```

☆1

サブ関数(subfunction)

・function M-file内にその関数内だけで使う専用関数を定義できる

・使い方は通常の記述の後にfunctionを定義するだけ

sample_subfunction.mの名前で保存

```
function b=sample_subfunction(a)
b=test_subfunction(a);

function d=test_subfunction(c)
d=c^2;
```

↑この部分がサブ関数

```
>> sample_subfunction(2)

ans =

     4

>> exist('sample_subfunction')

ans =

     2

>> exist('test_subfunction')

ans =

     0
```

サブ関数は外からは
見えない

http://www.mathworks.co.jp/help/ja_JP/techdoc/matlab_prog/f4-70666.html

☆2

グローバル(global)変数

・複数のM-ファイル内でたくさんの変数を共有したいとき、グローバル変数として定義すると便利

script M-file

```
script;
global p1 p2 p3;
p1=1; p2=2; p3=3;
c=sample_global(2);
```

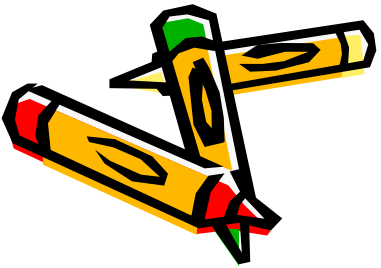
function M-file (sample_global.m)

```
function b=sample_global(a);
global p1 p2 p3;
b=p1*p2*p3*a;
return;
```

双方でglobal宣言された変数は値が共有される

・who('global')によりグローバル変数のリスト化

・isglobalによりグローバル変数の検出



http://www.mathworks.co.jp/help/ja_JP/techdoc/ref/global.html

☆1

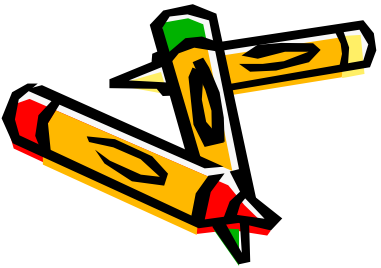
永続(persistent)変数

実行回数をカウントするサンプルプログラム

```
1   function num=sample_persistent
2
3   persistent p
4   if isempty(p), p=0;
5   else p=p+1;
6   end
7   num=p;
8
9   return
```

・M-ファイル内でpersistent宣言された変数はclearされるまで「そのfunction内」でのみ値を保持する

・最初の宣言時は空行列が代入される



```
>> clear all
>> sample_persistent

ans =

     0

>> sample_persistent

ans =

     1

>> sample_persistent

ans =

     2

>> clear all
>> sample_persistent

ans =

     0
```

←isempty(p)==1のためp=0

←p==0のためp=p+1

←p==1のためp=p+1

←clear all/functions/sample_persistent(関数名)で変数消去

←isempty(p)==1のためp=0

☆1

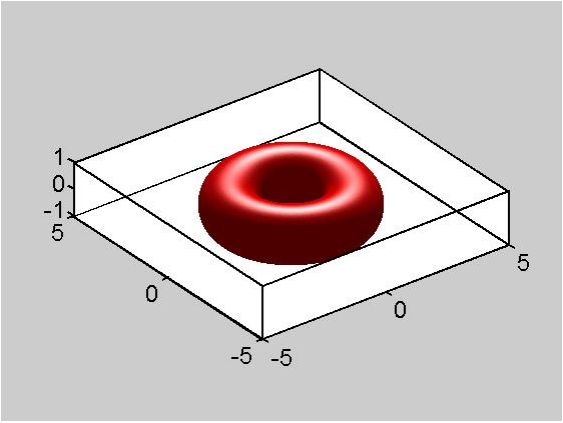
非等間隔データの3次元可視化

ドーナツ形状を描画するプログラム

```
% sample_torus

%%% パラメータの指定
a=1;    % 小半径
r0=2;   % 大半径
theta=(0:360)'; % ポロイダル角
phi=(0:360)';   % トロイダル角

lent=length(theta);
lenp=length(phi);

%%% 頂点の指定
[r,z]=pol2cart(theta*pi/180,a);
r=r+r0;
vertex=zeros(lent*lenp,3);
for ii=1:lenp
    phiii=phi(ii);
    [xii,yii]=pol2cart(phiii*pi/180,r);
    vertex((1:lent)+(ii-1)*lent,:)=[xii,yii,z];
end

%%% 面の指定
faces0=[(1:lent-1)',(2:lent)',[(2:lent)',(1:lent-1)']+lent];
faces=zeros((lent-1)*(lenp-1),4);
for ii=1:lenp-1
    faces((1:lent-1)+(ii-1)*(lent-1),:)=faces0+lent*(ii-1);
end

%%% 描画
figure;
p=patch('Vertices',vertex,'Faces',faces);
set(p,'FaceColor','red','EdgeColor','none');
daspect([1,1,1]); box on; camlight; lighting phong;
view(3);
```

・MATLABの可視化関数はデカルト座標(x,y,z)内でメッシュ状に配置できるデータを扱うものばかり

・例えば座標(r,z,phi)で並ぶデータを可視化する場合にはpatch関数を使うと良い



http://www.mathworks.co.jp/help/ja_JP/techdoc/visualize/f2-11758.html

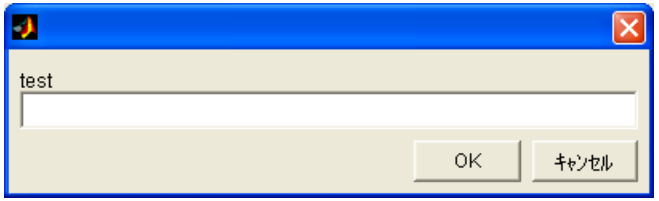
ダイアログボックスによる入出力

・ダイアログボックスを出現させての入出力が可能

waitbar

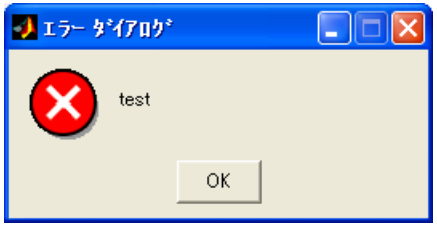
uigetfile

inputdlg

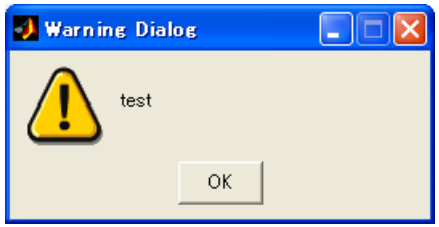


uisetcolor

errordlg



warndlg



uicontrol

☆1

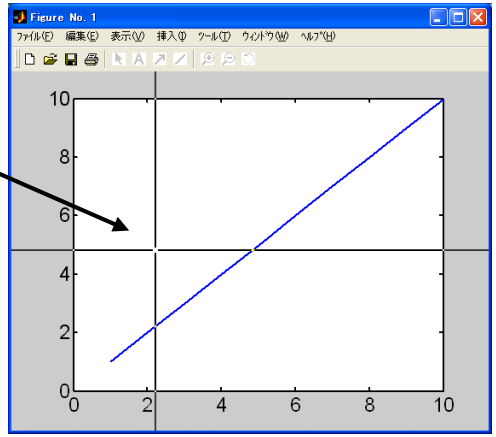
マウスを使った操作

・ginput関数によりマウスを使って図の座標値を取得できる

・gtext関数によりマウスを使って図上にテキストを配置できる

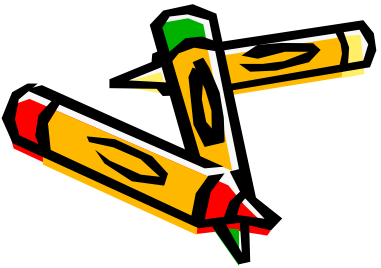
```
>> help ginput

  GINPUT   マウスを使ってグラフィックスへの入力

  [X,Y] = GINPUT(N) は、カレントのaxesから N 点を選択し、長さ N のベクトル X
  と Y にX座標とY座標を出力します。カーソルは、マウスを使って位置を決めるこ
  とができます(または、矢印キーを使うシステムもあります)。データ点は、マウス
  ボタンを押すか、キャリッジリターン以外のキーにより入力されます。キャリッジ
  リターンは、N 点が入力される前に、入力を終了します。
```

ターゲット



ginputの使用例

```
>> figure;
>> plot(1:10);
>> [x,y]=ginput(1)

x =

    2.2235


y =

    4.8099
```



☆o

Persistent変数を使った高度目なプログラム

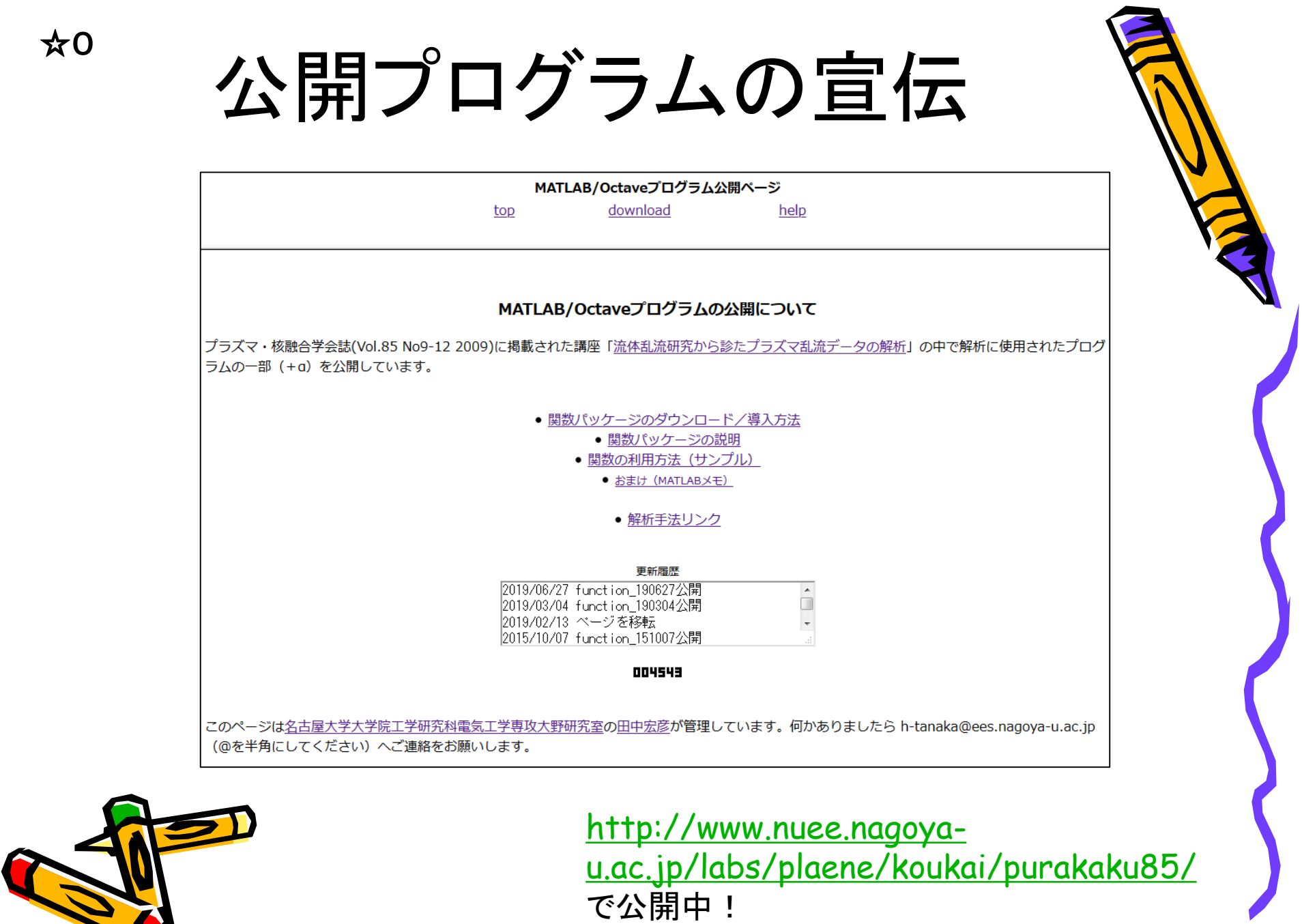
```
function num=call_times(s,lock)
%-------------------------------------------------
%   Remember and output a number of call times
%       呼び出し回数の記憶と出力
%                           Data: 2012/09/19
%                       Modified: 2012/09/19
%-------------------------------------------------
%
% NUM=CALL_TIMES(S)は、文字列・数値配列あるいは関数ハンドルS
% を入力とした「CALL_TIMES(S)」のコマンドが、過去に実行された
% 回数NUMを返します。NUMのカウントはCLEAR
ALL/FUNCTIONS/CALL_TIMES
% コマンドにより0に初期化されます。MATLAB終了時もカウントは
% 初期化されます。
%
% NUM=CALL_TIMES(S,LOCK)はLOCK==1のとき、CLEAR関数によるカウ
% ントの初期化を防ぎます。LOCK==0により、再度初期化を可能に
% します。
%
% 参考：PERSISTENT, MLOCK, MUNLOCK, CLEAR, ISA

% Author(s): H. Tanaka (National Institute for Fusion Science)
```

```
if nargin<2, lock=0; end
if ischar(s)
elseif isnumeric(s), s=num2str(s);
elseif isa(s,'function_handle'), s=func2str(s);
else, error('Unsupported input!');
end

val_name=['times_' s];
eval(['persistent ' val_name ';']); % persistent変数宣言

if isempty(eval(val_name))  % 初回
    eval([val_name '=0;']);
else                        % 2回目以降
    eval([val_name '=' val_name '+1;']);
end
num=eval(val_name);

switch upper(lock)  % カレントM-ファイルのロック
    case {1 'ON'}, mlock;
    case {0 'OFF'}, munlock;
end

return
```

コピペ可

☆o

公開プログラムの宣伝



MATLAB/Octaveプログラム公開ページ

top　download　help

MATLAB/Octaveプログラムの公開について

プラズマ・核融合学会誌(Vol.85 No9-12 2009)に掲載された講座「流体乱流研究から診たプラズマ乱流データの解析」の中で解析に使用されたプログラムの一部（＋α）を公開しています。

- 関数パッケージのダウンロード／導入方法
- 関数パッケージの説明
- 関数の利用方法（サンプル）
- おまけ（MATLABメモ）

- 解析手法リンク

更新履歴

2019/06/27 function_190627公開
2019/03/04 function_190304公開
2019/02/13 ページを移転
2015/10/07 function_151007公開

004543

このページは名古屋大学大学院工学研究科電気工学専攻大野研究室の田中宏彦が管理しています。何かありましたら h-tanaka@ees.nagoya-u.ac.jp（@を半角にしてください）へご連絡をお願いします。

http://www.nuee.nagoya-u.ac.jp/labs/plaene/koukai/purakaku85/
で公開中！