

# 主専攻実験 数理モデリングとアルゴリズム

## 発展課題：GUIの構築

### 1 はじめに

本実験ではMATLABを用いて簡単なGUIを作成する。例題として、ボタンを押すとsin関数を描くGUIとスクロールバーを移動させることで関数の線が太くなるGUIを作成する。例題を応用することで様々なGUIが作成可能となる。

### 2 GUIの作成1 (sin関数を描くボタンの作成)

今回はMatlabで簡単なGUIを作成する。以下にsin関数を描くボタンのGUIの作成手順を示す。

1. Matlabを立ち上げる。
2. コマンドラインにguideと入力。
3. Blank GUIを選択。
4. axesとpushbuttonを配置(図1)。

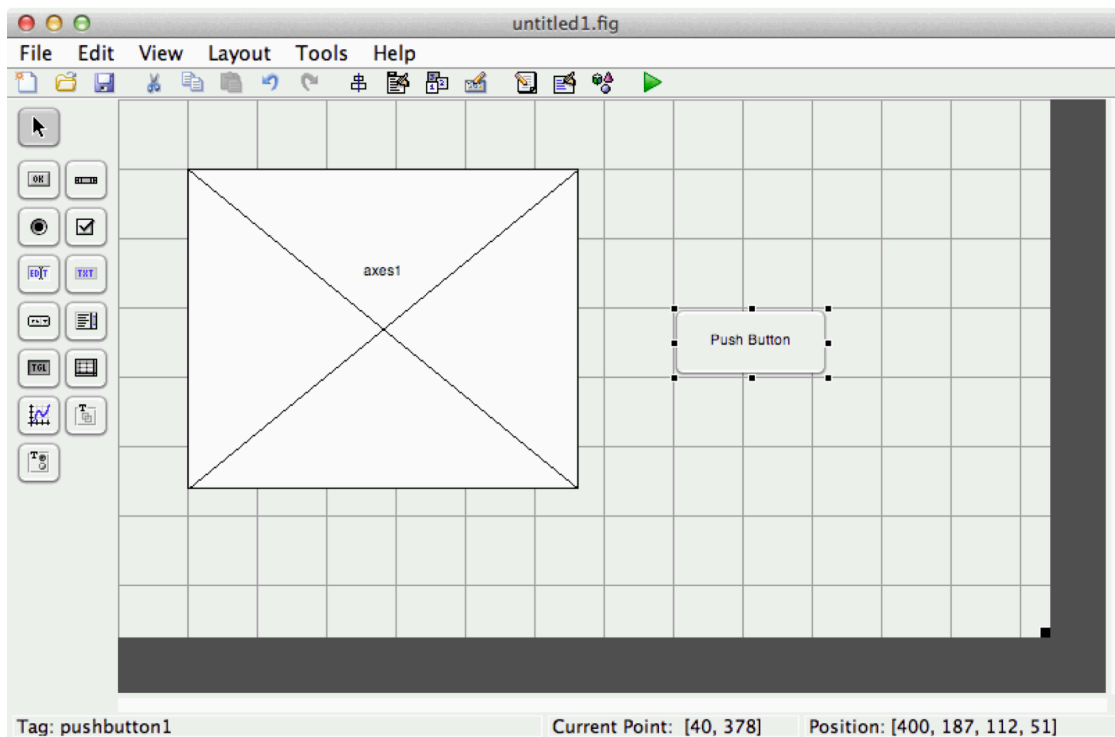


図 1: GUIの配置例

- 名前を付けて保存. 今回は sample.fig として保存.  
それにより, エディタに sample.m が出現.
- sample.m の function pushbutton1\_Callback(hObject, eventdata, handles) の下に axes 上に表示させたい関数を書き保存する.  
今回は,  $x = 0 \sim 10\pi$  までの  $\sin(x)$  関数を描画する.

```
x = 0:0.01:10*pi;
y = sin(x);
plot(x,y)
```

- コマンドラインに sample と入力するか, sample.fig の  $\Delta$  ボタンを押す.
- sample の GUI が出力され, pushbutton を押すと axes 上に  $\sin$  関数が表示される.

### 3 GUIの作成2 (sin関数の線を太くするスクロールバーの作成)

今回は前章と同様 Matlab を用いて GUI を作成する. 前章のプログラムをさらに発展させ  $\sin$  関数の線を太くするスクロールバーを作成する. 本章では GUI のプログラムでの関数間のデータの受け渡し方を示す.

- 前章で作成した sample.fig を編集するためコマンドラインに guide sample.fig と入力し, GUI を編集できるようにする.
- 前章で作成した GUI に対し, slider を追加. (図 2)

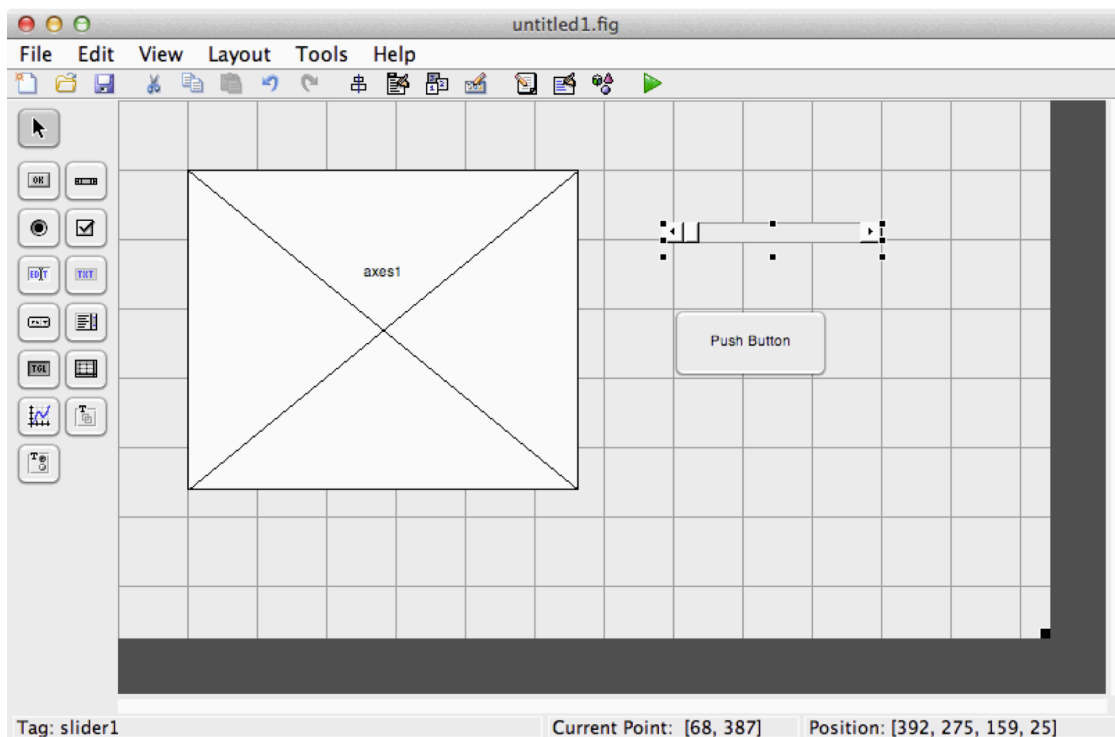


図 2: GUI の配置例 2

3. sample.m を開き, function pushbutton1\_Callback(hObject, eventdata, handles) に以下のプログラムを追加.

```
x = 0:0.01:10*pi;
y = sin(x);
handles.sinPlot = plot(x,y);
% handles.sinPlot = findobj('type','line') と同義
% 新しい handles 構造体のメンバー handles.sinPlot に
% オブジェクト Line (コマンド plot で作成されるオブジェクト) の
% データを格納
guidata(hObject, handles);
% 構造体の値を保存

set(handles.slider1, 'max', 30, 'min', 1, 'value', 1);
% slider の最大値を 30, 最小値を 1, 初期値を 1 に設定
set(findobj('type','axes'), 'xlim', [0 10*pi], 'ylim', [-1 1]);
% axes の X 軸を 0 ~ 10π, Y 軸を -1 ~ 1 に設定
```

4. function slider1\_Callback(hObject, eventdata, handles) に以下のプログラムを追加.

```
set(handles.sinPlot, 'linewidth', get(hObject, 'value'));
% handles.sinPlot の linewidth (線幅) プロパティの値を,
% slider の現在位置の値に設定
```

5. sample.m を保存し, 実行することで線の太さを操作するスクロールバーが作成される.

### 3.1 用語, 変数, 関数の説明

例題で用いられている用語や, 変数, 関数について説明する. 詳細な意味, 使い方やその他の関数については各自で調べること.

- 用語

1. オブジェクト: GUI が作成される際に作成されるもの (例: pushbutton, slider, axes, figure)
2. ハンドル: 各オブジェクトに設定されている番号
3. プロパティ: オブジェクトの属性値 (例: オブジェクト line なら linewidth, color, linestyle 等) GUI 編集時にオブジェクトをダブルクリックすると現れるダイアログで編集可能

- 変数

1. hObject: 現在実行されているオブジェクトのハンドル
2. handles: handles 構造体

- 関数

1. guidata(hObject, handles): handles 構造体の値を保存し, 他の関数内でも扱えるようにする関数

2. `set()` : オブジェクトのプロパティを設定する関数
3. `get()` : オブジェクトのプロパティを取得する関数
4. `findobj()` : オブジェクトのハンドルを取得する関数

関数間で受け渡しをしたい変数, 配列は新しい `handles` 構造体のメンバー `handles.***` (\*\*\*) は任意) に対して格納し, `guidata(hObject, handles)` により `handles` 構造体を保存すれば, 他の関数内でも `handles.***` の値が扱える.

## 4 実験課題

Matlab の GUI を用いてオリジナルの GUI を作成せよ. 課題 1, 2, 3 で学んだことを参考にしてもよい.

- 例 1. マウスを用いて Matlab のグラフ上に数字の絵を作成し, 画像認識を行う GUI.
- 例 2. 人工衛星の位置や速度をスクロールバーで設定し, 軌道と速度を求める GUI.