

CENTRO BRASILEIRO DE PESQUISAS FÍSICAS

Laboratório de Processamento Digital de Sinais e Imagens
(LPDSI/CAT/CBPF)

Introdução a Interface Gráfica no MATLAB

Érica Marques da Silva Santos
Marcelo P. de Albuquerque
Márcio P. de Albuquerque
Aline da Rocha G. Mello
Eugênio S. Caner

RIO DE JANEIRO, DEZEMBRO DE 2004

Sumário

1	Introdução	2
2	Handle Graphics	3
2.1	Handle de objetos	4
3	Estilos de Controle	6
4	GUIDE	9
5	Exemplo - Criando uma GUI	12
5.1	Criando a GUI com o GUIDE	13
5.2	Salvando e interpretando a GUI	14
5.3	Acessando o arquivo.M	14
	Referências	15

Introdução a Interface Gráfica no MATLAB

Érica M. Silva, Marcelo P. de Albuquerque, Márcio P. de Albuquerque,
Aline da Rocha G. Mello, Eugênio S. Caner

21 de dezembro de 2004

Resumo

Esta nota técnica apresenta uma introdução a interface gráfica no MATLAB. A princípio é feita uma breve introdução sobre uma GUI e é também apresentado uma definição sobre *Handles Graphics*. Os conceitos básicos necessários para iniciar o desenvolvimento de uma interface gráfica no MATLAB, são vistos em detalhes. O desenvolvimento de uma interface simples é dado como exemplo, no qual é apresentado todas as etapas necessárias para o desenvolvimento de uma GUI.

1 Introdução

Uma interface gráfica com o usuário é o ponto de contato ou método de interação entre uma pessoa e um computador ou programa de computador. É o método usado pelo computador e pelo usuário para trocarem informações. O computador exibe texto e imagens na tela e pode gerar sons com um alto falante. O usuário se comunica com o computador utilizando um dispositivo de entrada, como um teclado, mouse ou micorfone. A interface com o usuário define a aparência e a facilidade de uso do computador. Muitas vezes, um computador ou programa é escolhido com base na beleza do design e na eficiência funcional de sua interface com o usuário.

Uma interface gráfica com o usuário, ou gui, incorpora objetos gráficos, como janelas, ícones, botões, menus e texto. A seleção ou ativação desses objetos normalmente provoca uma ação ou mudança. O método mais comum de ativação consiste em usar um mouse ou outro dispositivo de seleção para controlar o movimento de um ponteiro na tela e pressionar um botão do mouse para selecionar um objeto ou alguma outra ação.

Uma interface gráfica pode ser usada para criar ferramentas muito eficientes no MATLAB ou construir demonstrações interativas de seu trabalho. As razões mais comuns para se criar uma interface gráfica com o usuário são as seguintes:

1. Você está escrevendo uma função que vai usar repetidas vezes e menus, botões ou caixas de texto podem ser usados como métodos de inserção dos dados.
2. Você está escrevendo uma função ou desenvolvendo um aplicativo para outros usarem
3. Você quer criar uma demonstração interativista de um processo, técnica ou método de análise
4. Você acha que GUIs são ótimas e quer fazer experiências com elas.

Este trabalho apresenta os princípios básicos para se criar uma GUI, porém antes de iniciarmos, será feita uma breve definição de Handle Graphics, que é um pré-requisito para projetar e implementar uma GUI no MATLAB.

2 Handle Graphics

O que é handle Graphics? Handle Graphics é o nome dado a uma coleção de rotinas de baixo nível que cumprem a tarefa de gerar gráficos no MATLAB. Handle Graphics proporciona um alto grau de controle sobre os gráficos. Suas funções e opções normalmente estão ocultas dentro de arquivos M de nível mais alto. Handle Graphics permite que você personalize aspectos de gráficos que não podem ser gerados por meio dos comando e funções de alto-nível. A barra de ferramentas e os menus da janela Figure fornecem uma interface Gráfica com o usuário (GUI) para muitos recursos Handle graphics, tornando menos importante, para o usuário ocasional do MATLAB, a compreensão do funcionamento de Handle Graphics.

Handle Graphics se baseia na idéia de que cada componente de um aspecto visual do MATLAB é um objeto, que cada objeto possui um único identificador, ou handle, associado a ele e que cada objeto possui propriedades que podem ser modificadas quando se desejar.

2.1 Handle de objetos

Suponha que você tenha três figuras abertas, com gráficos em duas delas, você quer alterar a cor de uma linha em um dos eixos dos gráficos. Como você identificaria a linha que deseja alterar. No MATLAB, cada objeto possui um identificador associado, chamado Handle, que é um número em precisão dupla. Cada vez que um objeto é criado, um Handle único é criado para ele. Todas as funções do MATLAB que criam objetos fornecem um handle ou um vetor coluna de Handles para cada objeto criado. Alguns gráficos são compostos por mais de um objeto. Por exemplo, um gráfico *mesh* consiste em um único objeto *surface*, enquanto um gráfico polar consiste em diversos objetos *lines* com handles individuais associados a cada linha. A tabela a seguir documenta as funções de Handle Graphics do MATLAB:

Função	Descrição
get	Obtém as propriedades do objeto
set	Define as propriedades do objeto
gcf	Obtém a <i>figura</i> atual
gca	Obtém os eixos atuais
gco	Obtém o objeto atual
findobj	Encontra os objetos com propriedades especificadas
findall	Encontra objetos ocultos e não-ocultos com propriedades especificadas
allchild	Retorna <i>Handles</i> de descendentes ocultos e não-ocultos de um objeto
copyobj	Faz cópia de objetos
root	Objeto <i>raiz(root)</i> do computador, <i>handle</i> = 0
figure	Cria uma janela <i>Figure</i>
axes	Cria <i>eixos</i>
line	Cria <i>linhas</i>
text	Cria <i>textos</i>
patch	Cria um <i>retalho</i>
rectangle	Cria um <i>retângulo</i>
surface	Cria uma <i>superfície</i>
image	Cria uma <i>imagem</i>
light	Adiciona efeitos de luz
uicontrol	Cria controle de interface com usuário
uimenu	Cria menu de interface com usuário
uicontextmenu	Cria menu de interface contextual com usuário
reset	Restabelece propriedades de objeto aos valores originais
clf	Apaga a <i>figura</i> atual
cla	Apaga os <i>eixos</i> atuais
ishandle	Verdadeiro para argumentos que são <i>handles</i> de objetos
delete	Remove um objeto
close	Fecha a janela <i>Figure</i>
refresh	Redesenha a <i>Figura</i> atual
gcbo	Ativa a manipulação do objeto atual
gcbf	Ativa a manipulação da figura atual
closereq	Fecha a figura
newplot	Cria eixos a partir da propriedade <i>NextPlot</i>

3 Estilos de Controle

O MATLAB aceita dez tipos de *uicontrol*, vejamos quais são:

1. *Botões simples (pushbuttons)*. Os *pushbuttons*, às vezes chamados de botões de comando ou simplesmente botões, são objetos de tela pequenos, retangulares que normalmente contêm um rótulo de texto. Selecionado um botão simples com o mouse movendo-se o ponteiro para cima o objeto e pressionando-se o botão do mouse, faz-se com que o MATLAB execute a ação definida pela callback string do objeto. Imediatamente depois que um botão simples é acionado, ele retorna a seu estado inicial. Os botões simples costumam ser usados para executar uma ação e não para mudar um estado ou fixar um atributo.
2. *Botões de chave (toggle buttons)*. Os botões de chave são idênticos aos botões simples exceto pelo fato de alternarem entre dois estados, para cima e para baixo, quando pressionados. A propriedade “value” de um botão de chave é fixada como igual ao valor especificado pela propriedade “Max” quando o botão está para baixo ou pressionado. É fixada como igual ao valor da propriedade “Min” quando o botão está para cima.
3. *Botões de rádio (radio buttons)*. Os botões de rádio consistem em botões contendo um rótulo e um pequeno círculo ou diamante à esquerda do texto de rótulo. Quando selecionado, o círculo ou diamante é preenchido e a propriedade “Value” é fixada como o valor especificado pela propriedade “Max”, que por padrão, é 1; quando não-selecionado, o indicador é removido e a propriedade “Value” é fixada como o valor especificado pela propriedade “Min”, que é 0 por padrão. Radio buttons costumam ser usados para selecionar uma de um grupo de opções mutuamente exclusivas. Para reforçar essa exclusividade, todavia, a callback string para cada botão de rádio deve desativar todos os outros botões de grupo fixando o “Value” de cada um deles em 0 ou com o valor atribuído à propriedade “Min”. Isso, no entanto, é apenas uma convenção. Os botões de radio podem ser usados em alternância com as caixas de controle, caso haja interesse.
4. *Caixas de controle*. As caixas de controle consistem em botões com um rótulo e uma pequena caixa quadrada à esquerda do texto do rótulo. Quando ativado, o controle alterna entre preenchido e limpo. No primeiro caso, a

caixa é preenchida ou contém um “X”, dependendo da plataforma e a propriedade “Value” recebe o valor especificado pela propriedade “Max”, que por padrão, é 1; quando limpo, a caixa fica vazia e a propriedade “Value” recebe o valor especificado pela propriedade “Min”, que por padrão, é 0. As caixas de controle costumam ser usadas para indicar o estado de uma opção ou atributo. Normalmente são objetos independentes, mas podem ser usados em alternância com botões de rádio, caso haja interesse.

5. *Caixas de edição.* Caixas de texto editáveis exibem texto em uma caixa, de maneira que você possa modificar ou substituir a string de texto dinamicamente, do mesmo como você faria em um editor de texto ou em um processador de palavras. A nova string de texto se torna, então, disponível na propriedade “string” do uicontrol. Caixas de texto editáveis costumam permitir que o usuário introduza textos ou um valor, e podem conter uma ou mais linhas de texto. Uma caixa de texto editável de linha única aceita uma linha de texto do usuário enquanto uma caixa de texto é concluída pressionando-se a tecla *Return*. A entrada de textos de múltiplas linhas é concluída com as teclas *Control-Return*. As caixas de texto de múltiplas linhas são criadas fixando-se os valores das propriedades “Max” e “Min” em números tais que $Max - Min > 1$. A propriedade “Max” não especifica o número máximo de linhas. As caixas de texto de múltiplas linhas podem ter um número ilimitado de linhas. Strings de múltiplas linhas podem ser especificadas como vetores celulares de texto ou vetor de caracteres.
6. *Caixas de texto.* Caixas de texto estatísticas são controles que simplesmente exibem uma string de texto como determinado pela propriedade “String”. Caixas de texto estatísticas costumam ser usadas para exibir rótulos, informação ao usuário ou valores atuais. Caixas de texto estatísticas são estatísticas; o usuário não pode mudar dinamicamente o texto exibido. O texto só pode ser alterado mudando-se a propriedade “String”. Strings de texto são centradas no topo da caixa de texto. Strings de texto maiores que a largura da caixa de texto passam por *word-wrap*; ou seja, linhas múltiplas são exibidas com as linhas quebradas entre palavras quando possível. Se a altura da caixa de texto é pequena demais para a string de texto, parte do texto não será visível. Strings de múltiplas linhas podem ser especificadas como um vetor celular de texto ou com um vetor de caracteres.
7. *Sliders.* Sliders, ou controles deslizantes, consistem em três partes distintas: o canal, ou a área retangular representando o intervalo de valores de

objetos válidos; o indicador dentro do canal representando o valor atual do slider; e setas em cada extremidade do canal. Sliders costumam ser usados para selecionar um valor de um intervalo de valores. Valores de slider podem ser estabelecidos de três maneiras. Na primeira, o indicador pode ser movido posicionando-se o ponteiro do mouse sobre o indicador, mantendo o botão do mouse pressionado enquanto se move o mouse, e soltando-se o botão quando o indicador estiver no local desejado. O segundo método é pressionar o botão do mouse enquanto o ponteiro estiver no canal, mas para um lado do indicador. O indicador se move naquela direção por alguma distância padrão igual a cerca de 10 do intervalo total do slider. Finalmente, clicando-se em uma das setas nas extremidades do slider o indicador, se nada diferente for especificado, se move cerca de 1 do intervalo do slider na direção da seta. Muitas vezes os sliders são acompanhados de objetos de texto separados usados para exibir rótulos, o valor atual do slider e limites de intervalo.

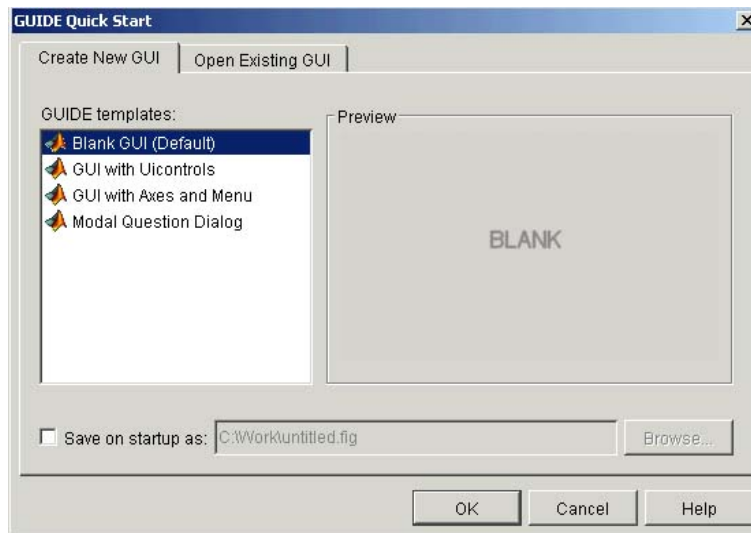
8. *Frame*. Objetos *uicontrol frame* são simplesmente regiões retangulares opacas, sombreadas, com uma margem. As frames são análogas à propriedade “Separator” dos objetos *uimenu*: definem a separação visual e costumam ser usadas para agrupar logicamente botões de rádio ou outros objetos *uicontrol*. As *frames* devem ser definidas antes que outros objetos sejam colocados dentro delas. Caso contrário, a *frame* pode encobrir os outros *uicontrol*.
9. *Caixas de listagem*. As caixas de listagem parecem caixas de texto de múltiplas linhas que permitem que os usuários selecionem itens individuais ou múltiplos de uma lista com um clique de mouse. Itens de lista individuais são especificados por vetores celulares de texto, por um vetor de string com espaços em branco, como separações, ou por uma única string com um caractere de barra vertical ‘|’ usado para separar as entradas da lista. Quando um item de lista é selecionado com um clique de mouse, a propriedade “value” é atualizada com o índice do item selecionado.
10. *Menu popup*. Os menus popup costumam ser usados para apresentar uma lista de escolhas mutuamente exclusivas ao usuário. Esses menus podem ser colocados em qualquer lugar da janela figure. Quando fechado, um menu popup aparece um retângulo ou um botão contendo o rótulo da seleção atual com um pequeno retângulo ou seta apontando para baixo saliente ao lado do rótulo para indicar que o objeto é um menu popup. Quando o ponteiro está colocado sobre um controle popup e o botão do mouse é pressionado,

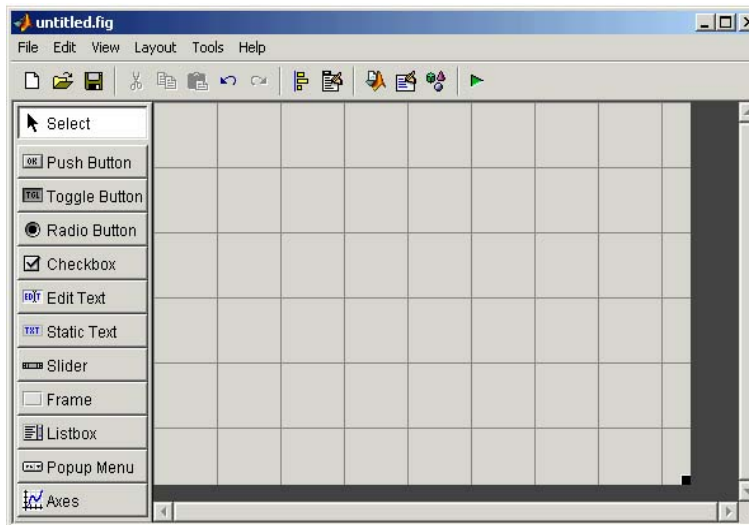
outras escolhas aparecem. Movendo-se o ponteiro para uma escolha diferente e soltando-se o botão do mouse, o menu popup se fecha e exibe uma nova seleção. As plataformas MS Windows e algumas X-Window System permitem que o usuário dê um clique em um menu popup para abri-lo e depois um outro clique em outra escolha para selecioná-la.

4 GUIDE

GUIDE é uma ferramenta GUI fornecida pelo MATLAB que foi projetada para que você construa suas próprias GUIs com maior facilidade e rapidez. A função *guide* contém ferramentas para criar, instalar, alinhar e alterar o tamanho de objetos *uicontrol*, um editor de propriedades e inspetor que lista propriedades de objetos e que permite ao usuário modificar essas propriedade interativamente; um menu editor para a edição interativa e para modificar os menus *pull-down* e contextuais definidos pelo usuário. GUIDE proporciona uma abordagem interativa ao desenvolvimento de GUIs e é ótima para ajustar a geometria de uma GUI. As versões anteriores dessa ferramenta não eram tão produtivas quanto escrever os seus próprios arquivos M GUI utilizando o editor.

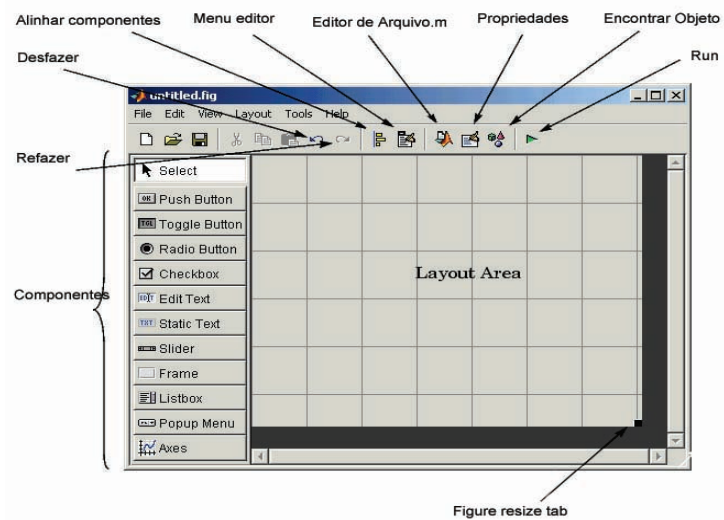
Para criar uma GUI através do GUIDE, você deve digitar o comando *guide* no prompt do MATLAB, em seguida aparece uma tela na qual você deve escolher dentre as opções a que é de seu interesse, após a seleção confirme e então irá aparecer a tela que corresponde a opção escolhida.





O GUIDE armazena uma GUI em dois arquivos, que são gerados quando você salva e interpreta uma GUI:

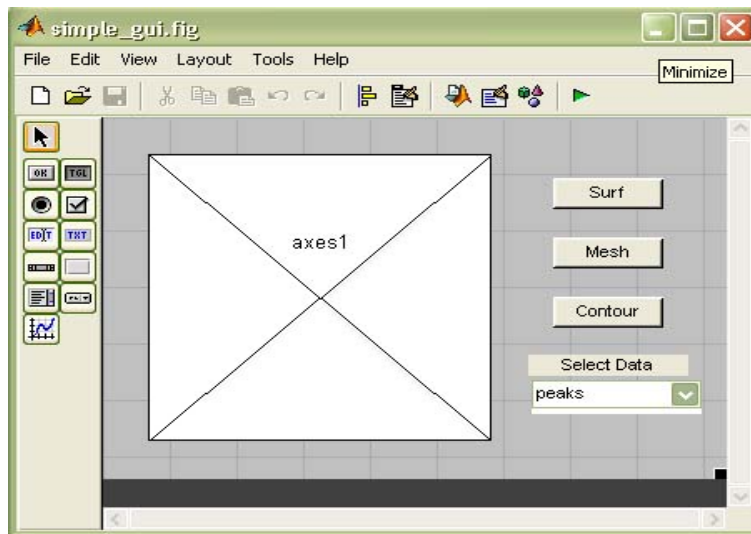
- Arquivo-FIG : um arquivo com a extensão *.fig*, contém uma descrição completa sobre as figuras presentes na GUI, tais como sliders, caixas de texto, menu popup e assim por diante. Quando você deseja fazer alguma modificação na aparência da GUI, esta deve ser feita no arquivo *.fig*.
- Arquivo-M : um arquivo com a extensão *.m*, contém o código que controla a GUI, incluindo as callbacks dos componentes. A princípio este arquivo está em branco, e cabe ao usuário inserir os comandos que sejam necessários para o funcionamento da GUI.



5 Exemplo - Criando uma GUI

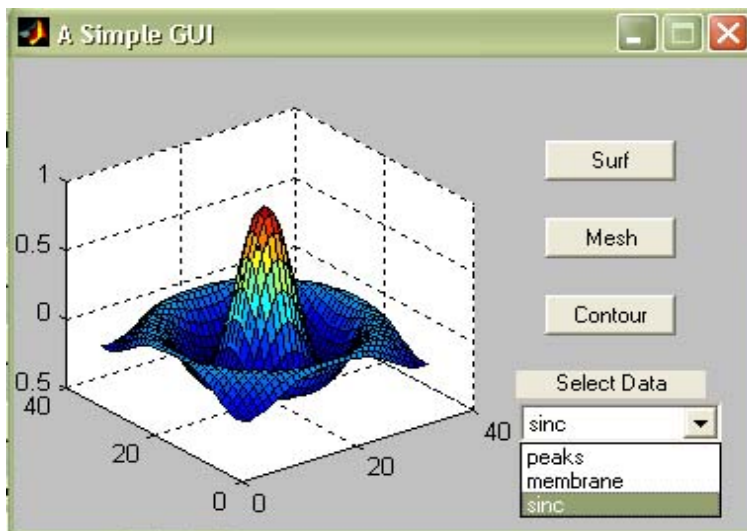
A GUI usada neste exemplo contém um gráfico que indica superfície, mesh ou o contorno dos dados selecionados no menu pop-up.

5.1 Criando a GUI com o GUIDE



5.2 Salvando e interpretando a GUI

Antes de escrever as callbacks, é necessário interpretar e salvar a GUI, para isto pressione o botão *Run*, no menu, uma caixa de diálogo aparece, para que você possa salvar sua GUI. Em seguida a seguinte figura é apresentada como resultado:



5.3 Acessando o arquivo.M

Após salvar a GUI, é possível escrever no *.m* as callbacks de cada componente da GUI. Uma maneira de visualizar o *.m* é através do próprio *.fig*, para isto leve o mouse até a opção *View* no menu e dentro desta opção escolha *M-file editor*.

Um editor de texto do MATLAB será aberto com o *.m* referente a GUI. Toda e qualquer modificação feita neste arquivo será enviado ao *.fig* após este ter sido salvo e interpretado.

Uma interface gráfica construída no MATLAB, pode ser tornar uma ferramenta extraordinária na demonstração e exploração de uma ampla variedade de dados técnicos, e isto se deve ao fato que o MATLAB dispõe de poderosos recursos numéricos e gráficos.

```
D:\Work\simple_gui.m
File Edit View Text Debug Breakpoints Web Window Help
Stack: Base
1 function varargout = simple_gui
2 % SIMPLE_GUI Application M-file
3 % OUT = SIMPLE_GUI Create
4 %
5 % SIMPLE_GUI Create the GUI
6 %
7 % SIMPLE_GUI('PropertyName',value2)
8 % PropertyName/PropertyValue pair arguments override defaults saved in
9 % simple_gui.fig. Unrecognized PropertyName/PropertyValue pairs are
10 % passed to simple_gui_OpeningFcn using varargin.
11 %
12 % See also: GUIDE, GCBO, OPENFIG, GUIDATA, GUIHANDLES, FEVAL
13
14 % Edit the above text to modify the response to help simple_gui
simple_gui
contour_pushbutton_Callback
mesh_pushbutton_Callback arguments.
select_plot_popup_Callback
simple_gui_OpeningFcn
simple_gui_OutputFcn
surf_pushbutton_Callback
simple_gui/select_pl... Ln 97 Col 21
```

Referências

- [1] Hanselman. Duane, Bruce Littlefield., "MATLAB 6 - Curso Completo", Editora Prentice Hall, Capítulo 31, 2004.