### Università degli Studi di Napoli *FEDERICO II*

A COUNTY DOLLARD OF THE PARTY O

Facoltà di Ingegneria

#### Corso di

Modelli e Metodi Numerici per l'Ingegneria

a.a. 2010/11

Progetto di fine corso:

Realizzazione di un' interfaccia grafica con GUI di Matlab.

Compilazione del file.m per ottenere l'eseguibile.

Prof.

Studenti:

Berardino D'Acunto

De Fusco Danilo M56/3

De Martino Francesco M56/59

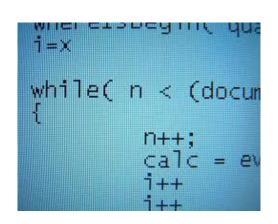
### **Sommario**

>	Introduzione.
>	Realizzazione di un'interfaccia utente con Matlab.
>	Elaborazione del codice contenente le istruzioni da eseguire dopo l'immissione dei dati.
>	Compilazione del file.m per ottenere il file.exe.
>	Realizzazione di un'applicazione WEB.



Nella realizzazione di un programma l'interfaccia grafica collegata al codice è di estrema utilità affinchè un qualsiasi utente possa utilizzare il programma.

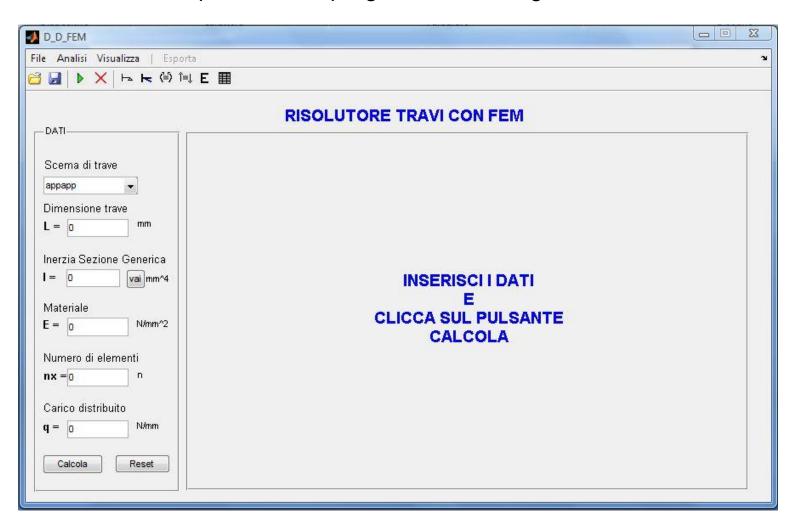
Nella maggior parte dei casi l'utente non è un programmatore e quindi non è interessato al codice ma solo al risultato finale.





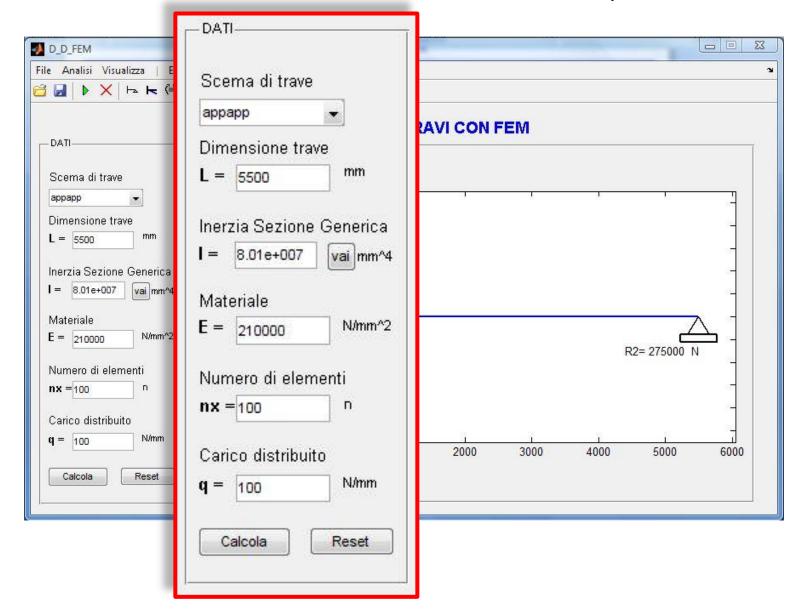
Per rendere più facile la vita dell'utente costruiremo, con Matlab, un'interfaccia grafica (GUI) mediante la quale l'utente inserirà i dati fisici e numerici, per risolvere diversi schemi di trave, e visualizzerà i risultati in termini di deformata, di diagrammi delle sollecitazioni e di diagramma dell'errore tra la soluzione esatta e quella approssimata.

L'interfaccia realizzata per il nostro programma è la seguente

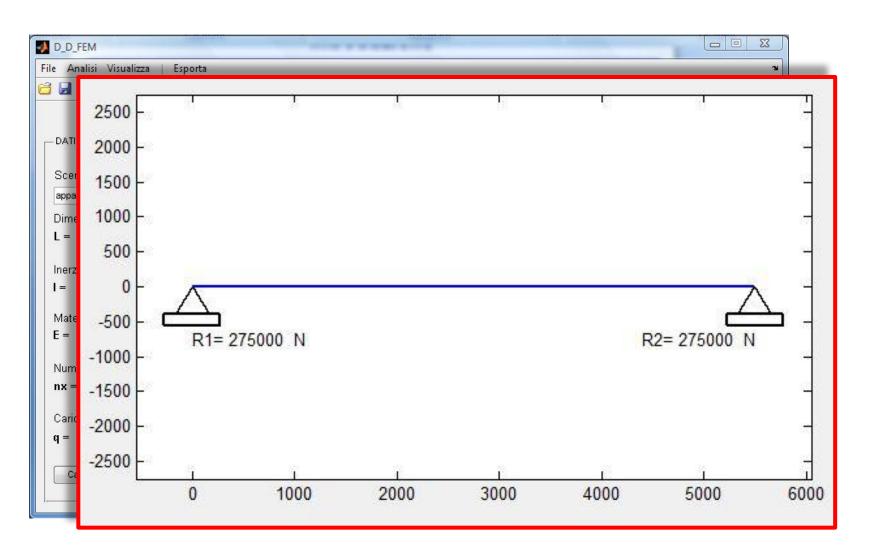


la quale è composta da:

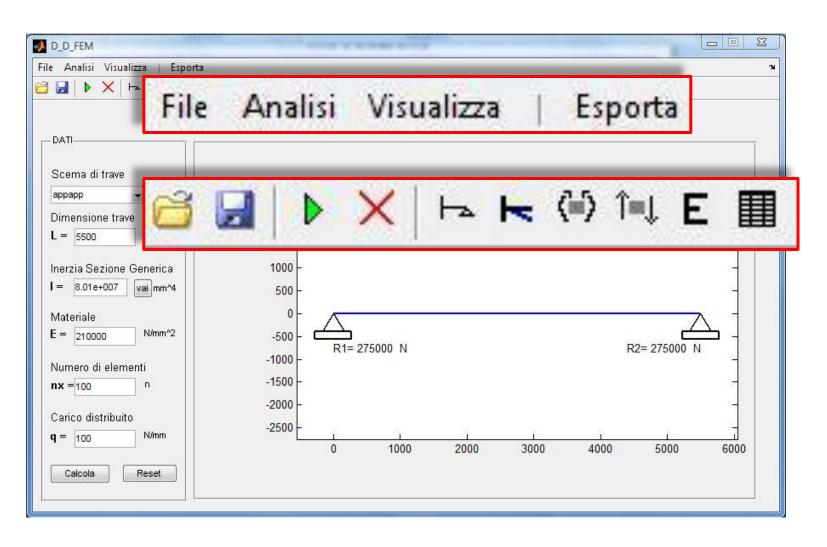
□ sezione dedicata all'immissione dei dati e alla scelta del tipo di struttura:



☐ sezione dedicata alla visualizzazione dei diagrammi di output



■ Barre dei menu e degli strumenti



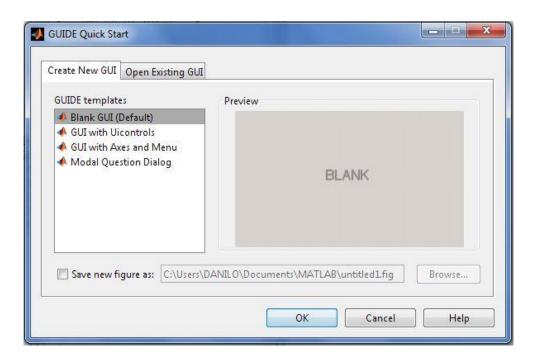
Dopo aver avviato Matlab per aprire l'editor di GUI possiamo procedere in tre modi:

- 1. clicchiamo dal menù (File -> New -> GUI);
- 2. dalla barra degli strumenti clicchiamo la seguente icona



3. digitiamo direttamente nella finestra dei comandi (guide).

Si avvia in questo modo la seguente finestra:

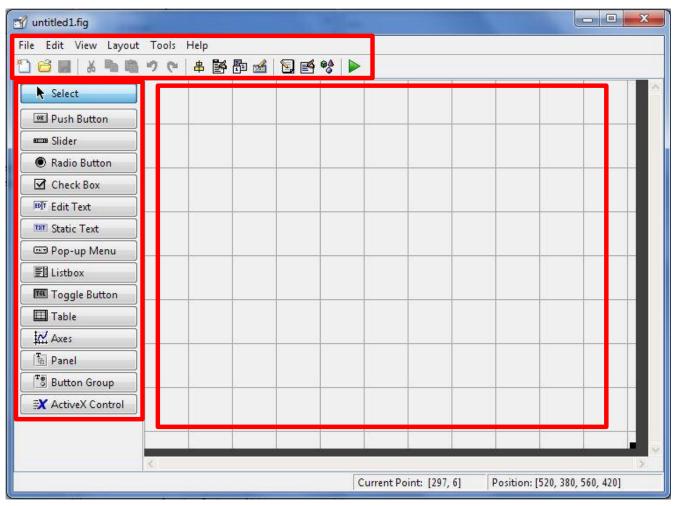


Selezioniamo la voce

Blank GUI (Default)

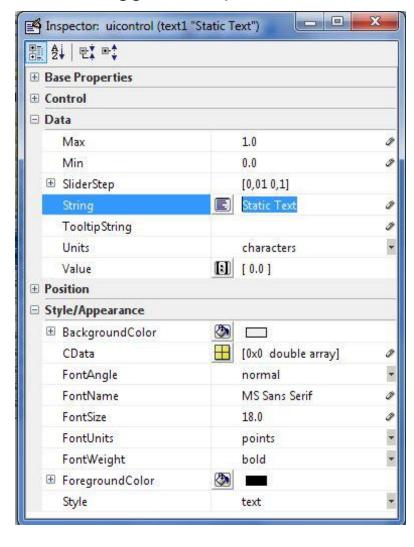
e clicchiamo sul pulsante **OK** 

Si apre l'editor di GUI che si presenta come segue:



L'editor presenta in alto una barra dei menù e una barra degli strumenti, a sinistra troviamo la barra degli oggetti che possiamo inserire nella nostra interfaccia e al centro troviamo l'area di lavoro.

Per realizzare il nostro programma inseriamo inizialmente un oggetto (**Static Text**) nell'area di lavoro. Per modificare il testo visualizzato facciamo doppio click sull'oggetto e apriamo la finestra delle proprietà (**Property Inspector**):



Portiamoci alla voce (*String*) e digitiamo il testo desiderato, nel nostro caso:

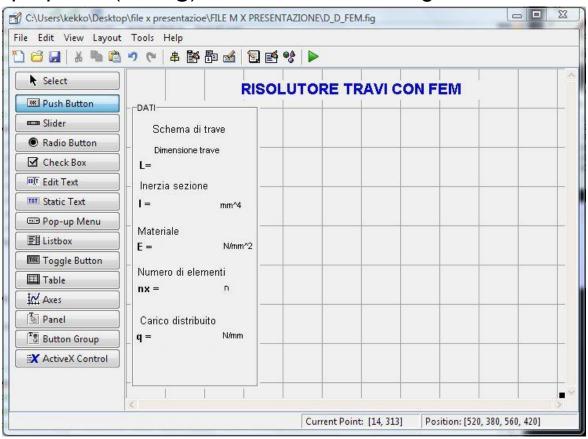
RISOLUTORE TRAVI CON FEM.

Per modificare le dimensioni e il formato del testo andiamo nel gruppo **Style/Apparence** e modifichiamo i campi **FontSize**, **FontWeight**, **FontName**.

Successivamente inseriamo un oggetto (*Panel*) che raggrupperà gli oggetti relativi all'inserimento dei dati.

Anche per quest'oggetto è possibile modificare le proprietà attraverso il (*Property Inspector*). Nel nostro caso modifichiamo il campo (*String*) digitando "**Dati**".

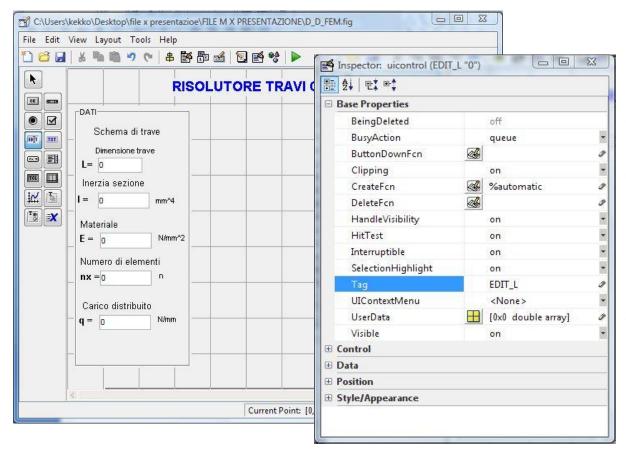
Andiamo ora ad inserire all'interno del Panel appena creato altri oggetti (*Static Text*) modificandone la proprietà (*String*) fino ad ottenere il seguente risultato.



A questo punto occorre inserire al fianco dei testi statici gli (*Edit Text*) che serviranno all'inserimento dei dati per la simulazione.

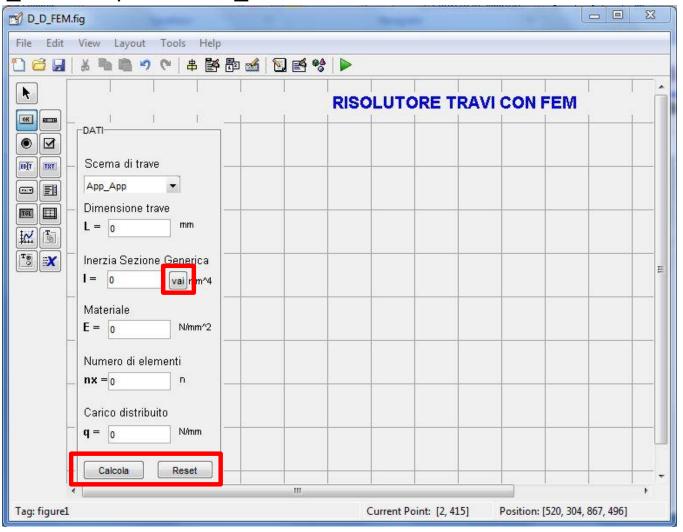
Inseriamo un primo (*Edit Text*) al fianco della (*Static Text*) "L =" e gli assegniamo il valore "0" come (*String*) e "EDIT L" come (*Tag*).

In modo analogo assegniamo agli altri (*Edit Text*) il valore "0" come (*String*) e "EDIT\_I", "EDIT\_E", "EDIT\_nx", "EDIT\_q" come (*Tag*).

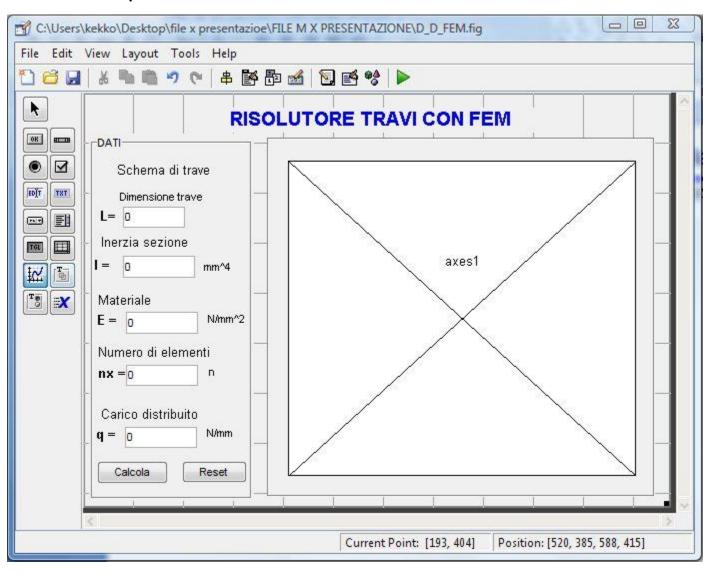


Il (*Tag*) è il nome che verrà associato all'oggetto per poterlo richiamare all'interno del codice. Il nome deve indicare la variabile contenuta nell'oggetto. Ad esempio la nostra "EDIT\_L" contiene la lunghezza dell'elemento strutturale.

Procediamo ora con l'inserimento di tre (*Push Button*) modificandone la (**String**) in "Calcola", "Reset" e "Vai" e il (*Tag*) rispettivamente in "pushbutton\_calcola", "pushbutton\_reset" e "pushbutton\_section"

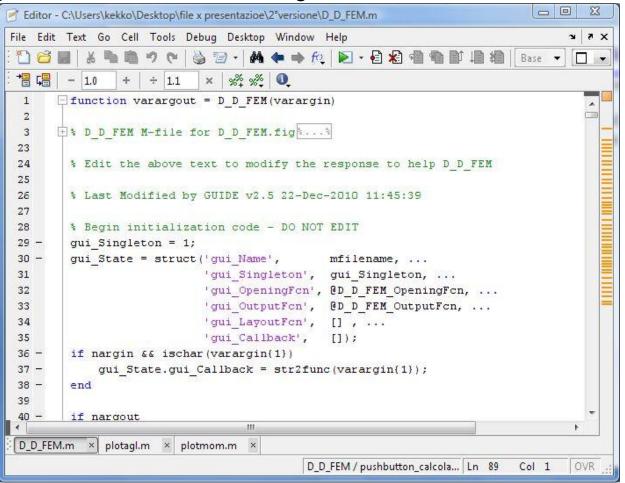


Passiamo ora all'inserimento del secondo (*Panel*) e di un oggetto (*Axes*) per la visualizzazione dell'output.

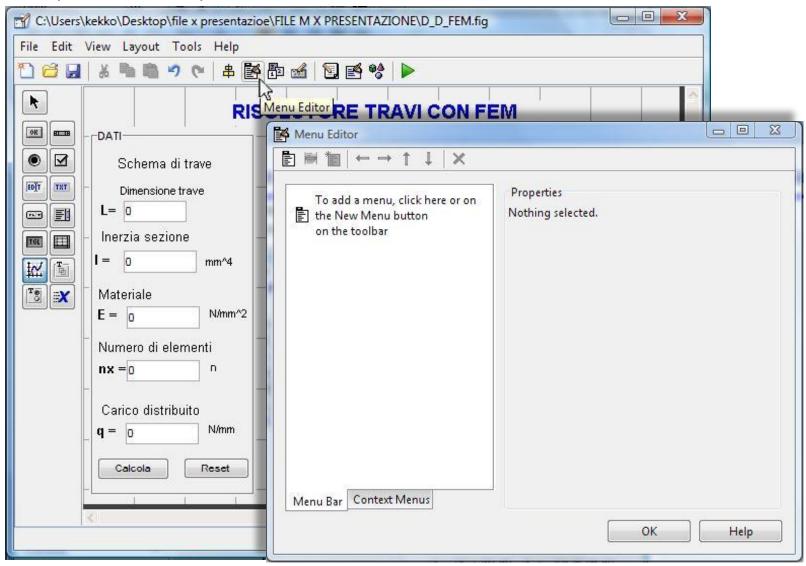


Salviamo il progetto ed assegniamogli il nome "**D\_D\_FEM**" creando quindi il file "**D\_D\_FEM.fig**".

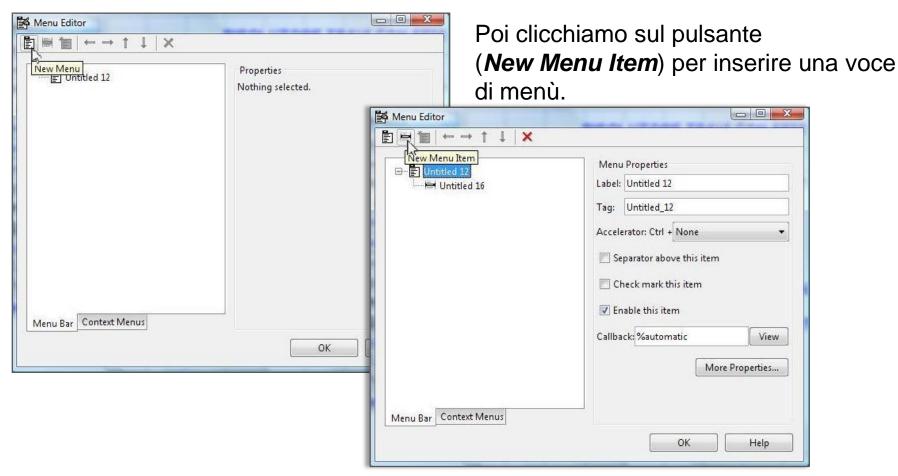
Una volta salvato, l'editor di GUI genera automaticamente il file "**D\_D\_FEM.m**" contenente il codice per gestire l'interfaccia e quindi tutti gli oggetti inseriti fino ad'ora. Si riporta di seguito uno stralcio del codice generato automaticamente.



Per creare la barra dei menu clicchiamo sul pulsante (*Menu Editor*) come in figura che apre la finestra (*Menu Editor*).

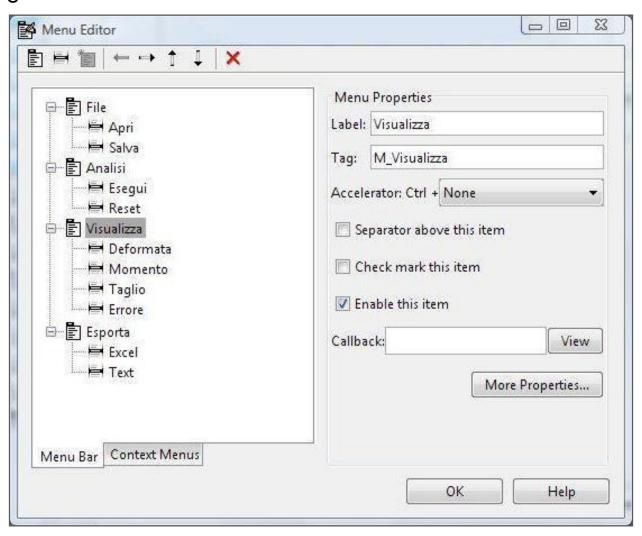


Nella finestra (*Menu Editor*) clicchiamo sul pulsante (*New Menu*) per inserire un nuovo menù.

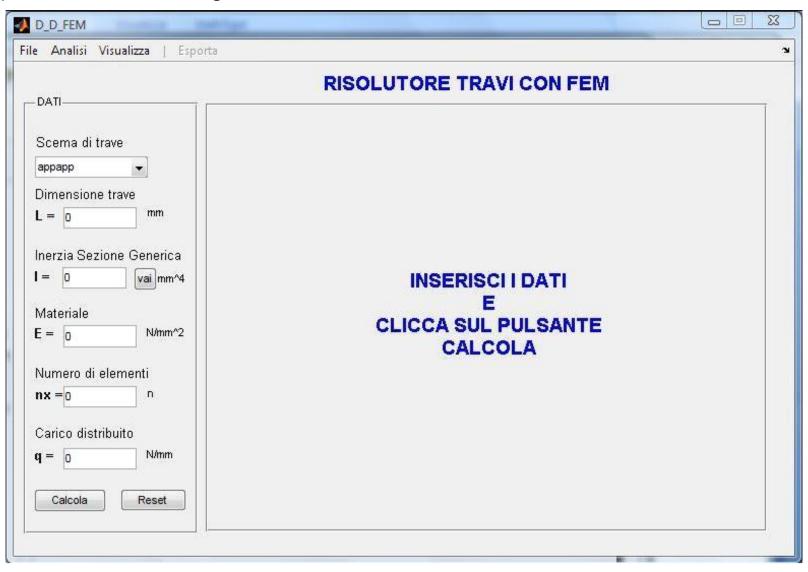


Nominiamo il menu modificando la proprietà (*Label*) in "File" e quindi assegniamo come (*Tag*) il nome "M\_File".

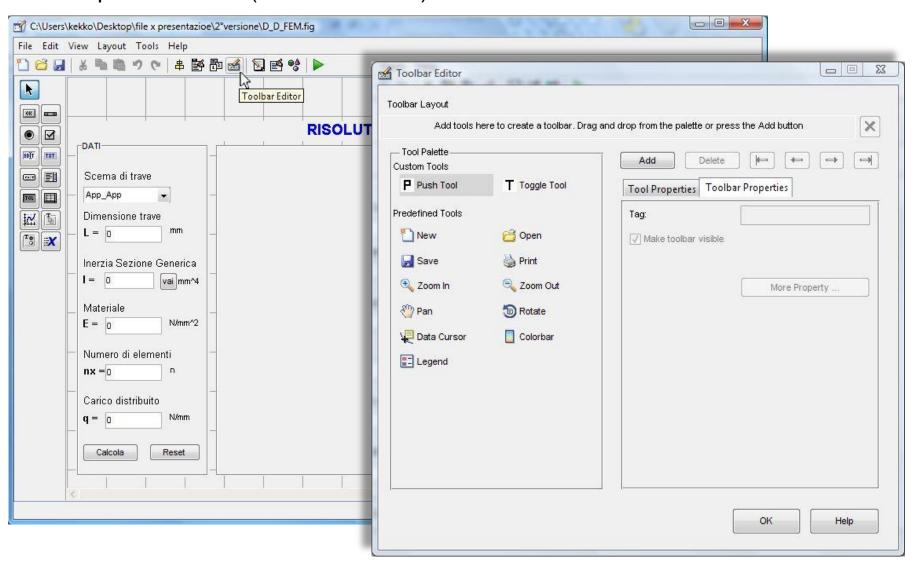
Per completare la barra procediamo allo stesso modo per tutti i menù e le voci di menù ottenendo il seguente risultato.



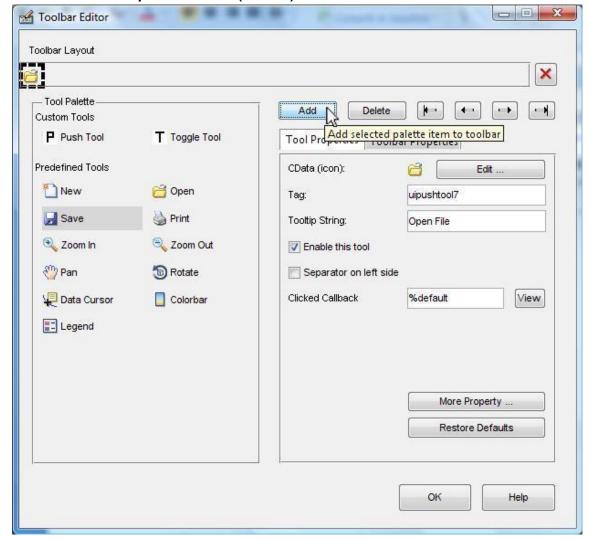
Per poter visualizzare il risultato finale dobbiamo avviare la figura cliccando sulla freccia verde posta sulla barra degli strumenti dell'editor di GUI.



Per creare la barra degli strumenti clicchiamo sull'icona (*Toolbar Editor*) nell'editor di GUI che apre la finestra (*Toolbar Editor*).

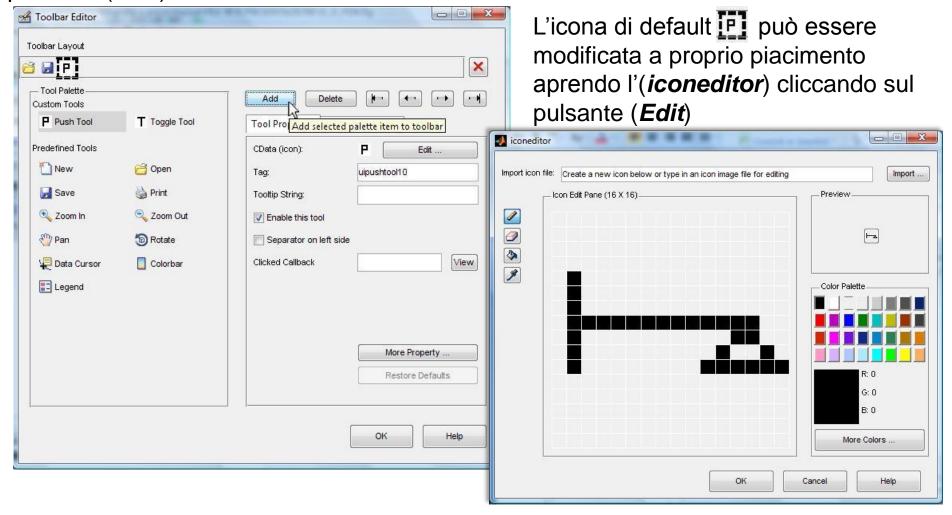


Inseriamo all'interno della nostra barra degli strumenti i pulsanti "Apri" e "Salva" scegliendo le relative icone all'interno di (*Tool Palette*) - (*Predefined Tools*) e cliccando sul pulsante (*Add*).

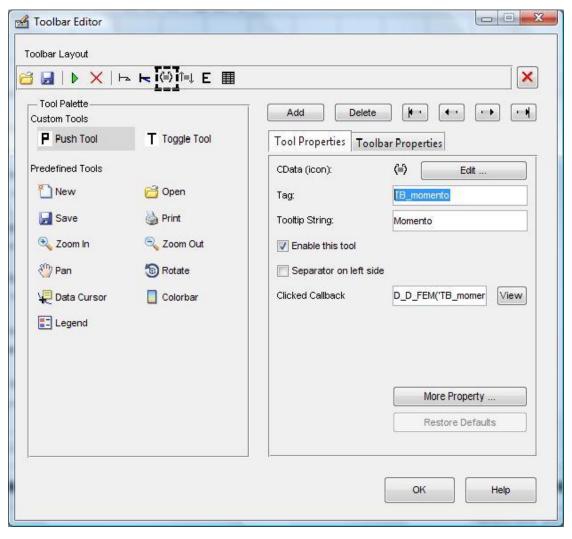


Modifichiamo le proprietà (*Tag*) scrivendo "TB\_apri" e "TB\_salva".

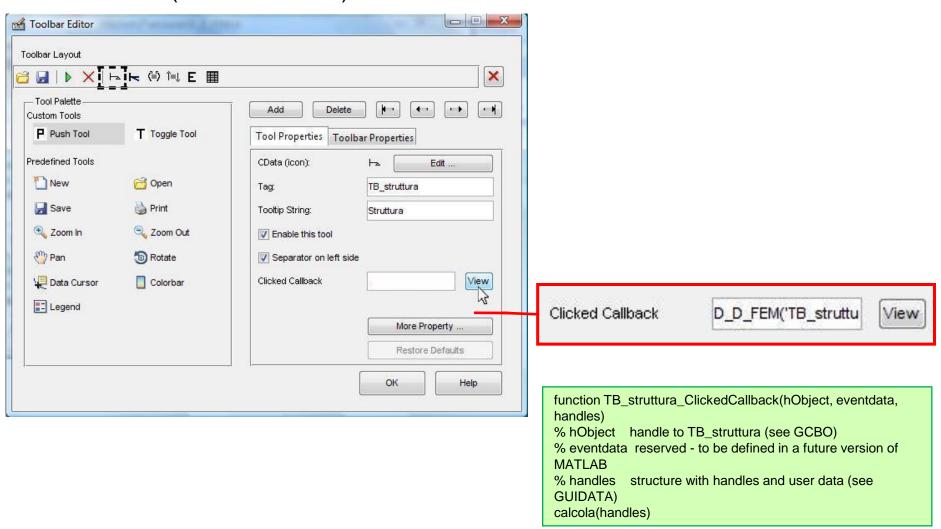
Completiamo la barra degli strumenti creando i pulsanti personalizzati: "Esegui", "Reset", "Struttura", "Deformata", "Momento", "Taglio", "Errore" e "Tabelle dati" scegliendo (*Push Tool*) all'interno di (*Tool Palette*) - (*Custom Tools*) e cliccando sul pulsante (*Add*).

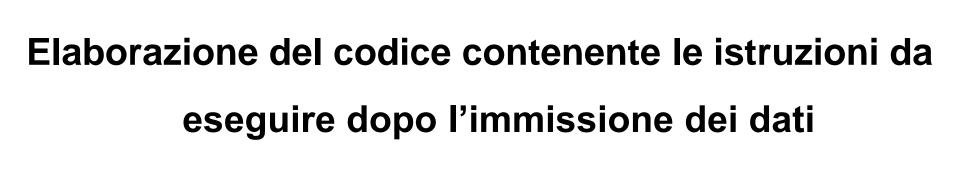


Per ogni pulsante personalizzato realizzato precedentemente modifichiamo, all'interno di (*Tool Properties*) il (*Tag*) scrivendo: "TB\_struttura", "TB\_deformata", "TB\_momento", "TB\_taglio", "TB\_errore", "TB\_tabelle".



Per ogni pulsante inserito all'interno dalla barra degli strumenti occorre generare il codice. Ciò viene fatto automaticamente dall'editor di GUI cliccando il pulsante (*View*) all'interno della (*Toolbar Editor*).





Dopo aver completato l'inserimento degli oggetti grafici dobbiamo scrivere il codice necessario sia alla gestione degli stessi, sia alla gestione delle funzioni esterne.

Come già detto precedentemente l'editor di GUI genera in automatico il file D\_D\_FEM.m associato alla nostra interfaccia. Le prime 44 righe di questo file non vanno modificate perché servono a stampare a video la finestra principale. Mentre nelle righe successive sono riportate tutte le funzioni associate agli oggetti grafici.

Per inizializzare la finestra principale prima che questa venga visualizzata viene creata la funzione D\_D\_FEM\_OpeningFcn.

Per ogni oggetto *Edit text* troviamo una funzione Createfon (create function) che inizializza l'oggetto e una funzione Callback (funzione risposta) che viene richiamata quando si modifica il contenuto dell'oggetto stesso.

Per gli oggetti push button, per i menu, per le voci di menu e per i pulsanti della barra degli strumenti troviamo la funzione Callback (funzione risposta) che viene richiamata quando clicchiamo su di essi.

Come primo passo scriviamo la funzione initialize\_gui che è formata dal seguente blocco di istruzioni:

```
function initialize_qui(handles)
                                                                          handles.vettori.vinc1v=[]:
                                                                          handles.vettori.vinc2x=[];
%creo la struttura (dati) che conterrà i dati del problema.
                                                                          handles.vettori.vinc2y=[];
                                                                          handles.vettori.strx=[];
handles.dati.L=0;
                                                                          handles.vettori.strv=[]:
handles.dati.E=0;
handles.dati.l=0;
                                                                          %salvo le strutture appena create (dati e vettori)
handles.dati.nx=0;
handles.dati.q=0;
                                                                          %nella struttura principale handles
handles.dati.strutt=1;
                                                                          guidata(handles.figure1,handles);
%fattore di scala per i diagrammi
handles.dati.su=10;
                                                                          %passiamo i dati alle edit_text
handles.dati.sm=0.000005;
                                                                          set(handles.popup_schematrave,'Value',1);
                                                                          set(handles.EDIT_L,'String',handles.dati.L);
handles.dati.sv=0.001;
handles.dati.funz=[];
                                                                          set(handles.EDIT_E,'String',handles.dati.E);
                                                                          set(handles.EDIT_I,'String',handles.dati.I);
%creo la struttura (vettori) che conterrà i vettori
                                                                          set(handles.TEX_sezione, 'String', 'Sezione Generica');
%restituiti dalle funzioni risolutrici
                                                                          set(handles.EDIT_nx,'String',handles.dati.nx);
                                                                          set(handles.EDIT_q,'String',handles.dati.q);
handles.vettori.x=[];
handles.vettori.u=[];
handles.vettori.U=[];
                                                                          %nascondo axes
handles.vettori.v=[];
                                                                          set(handles.axes1,'Visible','off');
handles.vettori.V=[];
                                                                          %visualizzo avviso
handles.vettori.mo=[];
                                                                          set(handles.TEXT_AVVISO,'Visible','on');
handles.vettori.Mo=[];
                                                                          %blocco menu esporta
handles.vettori.Er=[];
                                                                          set(handles.M_Esporta, 'Enable', 'off')
handles.vettori.vinc1x=[];
                                                                          cla;
```

Con tale funzione vengono create le strutture (dati e vettori) che contengono dati condivisi tra varie funzioni. Vengono inoltre settati i valori iniziali da visualizzare nelle Edit Text.

La funzione initialize\_gui sarà richiamata quando viene inizializzata la finestra principale, quando clicchiamo sul pulsante *Reset*, quando clicchiamo *Analisi->Reset* dalla barra dei menu e quando clicchiamo il pulsante *Reset* dalla barra degli strumenti.

Purché questo avvenga occorre richiamare la funzione initialize\_gui all'interno delle rispettive funzioni Callback come riportato di seguito:

```
function D_D_FEM_OpeningFcn(hObject, eventdata, handles, varargin)

% This function has no output args, see OutputFcn.
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
% varargin command line arguments to D_D_FEM (see VARARGIN)
% Choose default command line output for D_D_FEM
handles.output = hObject;
% Update handles structure
guidata(hObject, handles);
initialize_gui(handles);
```

```
function pushbutton_reset_Callback(hObject, eventdata, handles)

% hObject handle to pushbutton_reset (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%resetto la finestra avviando la funzione initialize_gui
initialize qui(handles)
```

```
function TB_reset_ClickedCallback(hObject, eventdata, handles)

% hObject handle to TB_reset (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

initialize_gui(handles)
```

```
function M_Reset_Callback(hObject, eventdata, handles)

% hObject handle to M_Reset (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

initialize_gui(handles)
```

Procediamo ora con la scrittura della funzione calcola che esegue le seguenti azioni:

- salva nella struttura "handles.dati" i valori numerici inseriti nelle Edit Text e la tipologia di struttura scelta attraverso il Popup Menu;
- esegue un controllo per verificare la compatibilità dei dati inseriti;
- richiama, attraverso il costrutto swich, la funzione adatta a risolvere il tipo di struttura selezionato nel Popup Menu;
- salva nella struttura handles.matrici i seguenti vettori (x',u,v,mo) restituiti dalla funzione risolutrice al fine di poterli visualizzare ed esportare in altri formati (ad esempio .xls);
- attiva il Menu Esporta inizialmente disabilitato.

La funzione calcola è formata dal seguente blocco di istruzioni:

```
function calcola(handles)
                                                                               case 2
                                                                                  [handles]=incapp(handles);
%salvo i dati inseriti nelle edit_text nella struttura (dati)
                                                                               case 3
handles.dati.L = str2double(get(handles.EDIT_L,'String'));
                                                                                  [handles]=incinc(handles);
handles.dati.E = str2double(get(handles.EDIT_E,'String'));
                                                                               case 4
handles.dati.l = str2double(get(handles.EDIT_I,'String'));
                                                                                  [handles]=inclib(handles);
handles.dati.nx = str2double(get(handles.EDIT_nx,'String'));
                                                                               otherwise
handles.dati.q = str2double(get(handles.EDIT_q,'String'));
                                                                                 [handles]=appapp(handles);
handles.dati.strutt = get(handles.popup_schematrave,'Value');
                                                                            end
                                                                            %salvo i vettori restituiti dalla funzione nella struttura (vettori)
%se L=0 la funzione si arresta
if handles.dati.L==0
                                                                            x = handles.vettori.x;
                                                                            u = handles.vettori.u;
  return
end
                                                                             v = handles.vettori.v;
                                                                            mo = handles.vettori.mo;
%visualizzo l'oggetto axes1
set(handles.axes1,'Visible','on');
                                                                            %creo la struttura (matrici)nella quale salvo la matrice Mdati che ha per
                                                                            %colonne i vettori (x' u mo v) necessaria per il riempimento e il salvataggio
                                                                            %dei dati numerici
%nascondo la static text
set(handles.TEXT_AVVISO,'Visible','off');
                                                                            handles.matrici.Mdati=[x',u,mo,v];
%avvio la funzione "appapp", "incapp", "incinc" o"inclib"
                                                                            %salvo la struttura (matrici) appena creata
%in base alla selezione fatta dal popupmenu
                                                                            quidata(handles.figure1,handles);
switch handles.dati.strutt
                                                                            %attivo il menù esporta
                                                                            set(handles.M_Esporta, 'Enable', 'on')
  case 1
     [handles]=appapp(handles);
```

E sarà richiamata quando clicchiamo i pulsanti *Calcola*, *Analisi->Esegui* dal menu, *Esegui* dalla barra degli strumenti e quando selezioniamo lo schema di trave dal *popupmenu*.

Si riportano di seguito i blocchi di codice relativi alle funzioni Callback degli oggetti sopra elencati:

function pushbutton\_calcola\_Callback(hObject, eventdata, handles) % hObject handle to pushbutton\_calcola (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) %richiamo la funzione calcola calcola(handles)

function M\_Esegui\_Callback(hObject, eventdata, handles)
% hObject handle to M\_Esegui (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
calcola(handles)

function TB\_esegui\_ClickedCallback(hObject, eventdata, handles) % hObject handle to TB\_esegui (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA) calcola(handles)

function popup\_schematrave\_Callback(hObject, eventdata, handles) % hObject handle to popup\_schematrave (see GCBO) % eventdata reserved - to be defined in a future version of MATLAB % handles structure with handles and user data (see GUIDATA)

calcola(handles)

Per il nostro progetto sono state compilate le seguenti funzioni risolutrici:

- 1. appapp.m per lo schema di trave appoggio-appoggio;
- 2. incapp.m per lo schema di trave incastro-appoggio;
- 3. **incinc.m** per lo schema di trave incastro-incastro;
- 4. **inclib.m** per lo schema di trave incastro-estremo libero.

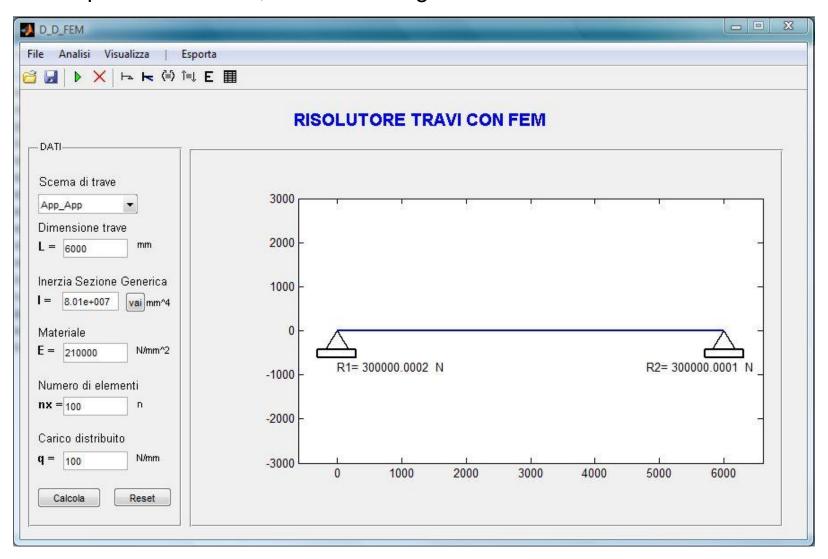
#### Esaminiamo ora il listato della funzione appapp.m:

```
function [handles]=appapp(handles)
                                                                       L=(k(1:2,1:2)+k(3:4,3:4));
                                                                       %------MATRICE DI RIGIDEZZA GLOBALE-----
%------DATI-----
                     %lunghezza della trave
                                                                       K=zeros(m,n);
L = handles.dati.L;
E = handles.dati.E;
                     %modulo elastico del materiale
                                                                       K(1:1,1:1)=[A];
                     %inerzia della sezione scelta
                                                                       K(1:1,2:3)=[B]:
I = handles.dati.I;
nx = handles.dati.nx; %numero di elementi finiti
                                                                       K(2:3,1:1)=[B'];
g = handles.dati.g;
                     %carico distribuito
                                                                       K((m-2):(m-1),n:n)=[C];
                                                                       K(m:m,(n-2):(n-1))=[C'];
I=L;
                                                                       K(m:m,n:n)=[D];
EI=E*I;
                                                                       if nx > = 3
          %lunghezza elemento
                                                                         K(2:3,2:5)=[L M]:
h=l/nx;
                                                                         K((m-2):(m-1),(n-4):(n-1))=[M'L];
          %Dimensione matrice K:m identifica il numero di righe
m=nx*2;
          %Dimensione matrice K:n identifica il numero di colonne
n=nx*2;
                                                                         z=4;y=2;
x=linspace(0,l,nx+1);
                                                                         for i=1:nx-3
              %matrice di rigidezza del singolo elemento finito
                                                                           K(z:(z+1),(y:y+5))=[M' L M];
k=Getk(EI,h);
                                                                           z=z+2;
%Si può quindi definire la matrice di rigidezza K dell'intera trave
                                                                           y=y+2;
%suddividendola in alcune sottomatrici
                                                                         end
A=k(2:2,2:2);
                                                                       else
B=k(2:2,3:4);
                                                                         K(2:3,2:3)=[L];
C=k(1:2,4:4);
                                                                       end
D=k(4:4,4:4);
M=k(1:2,3:4);
```

```
%-----Si definisce il vettore dei carichi------
                                                                                                                                                                                                 xc(nx)).^3/h^2;
                                                                                                                                                                                                 N4=-(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^3/h^2;
fd=zeros(m.1):
fd(1)=-q*h^2/12;
                                                                                                                                                                                                  %------Deformata approssimata-----
fd(m)=q*h^2/12;
                                                                                                                                                                                                 u=zeros(nx+1,1);
                                                                                                                                                                                                       %valore dello spostamento nell'estremo iniziale
fd(m-1)=0;
                                                                                                                                                                                                 u(1:2,1)=N21*us(1)+N31*us(2)+N41*us(3);
z=2;
                                                                                                                                                                                                       %valore dello spostamento nell'estremo finale
for i=1:nx-1
      fd(z:z+1)=[-q*h 0]';
                                                                                                                                                                                                 u(nx:nx+1,1)=N1*us(m-2)+N2*us(m-1)+N4*us(m);
      z=z+2;
                                                                                                                                                                                                j=2;p=2;
end
                                                                                                                                                                                                 for i=1:nx-2
  %------FEM------
                                                                                                                                                                                                       u(j:j+1,1)=N1*us(p)+N2*us(p+1)+N3*us(p+2)+N4*us(p+3);
us=K\fd; %coefficienti approssimati
                                                                                                                                                                                                      j=j+1;
 U=-q^{(x'.^4-2^*l^*x'.^3+l^3^*x')/24/EI}; %deformata esatta di una trave app-app
                                                                                                                                                                                                      p=p+2;
xc=zeros(nx+1.1): %vettore colonna delle coordinate dei vari elementi
                                                                                                                                                                                                 end
xc(1)=0;
                                                                                                                                                                                                 Er=U-u:
xc(nx+1)=1;
                                                                                                                                                                                                 %------REAZIONI------
                                                                                                                                                                                                 F1=k(1,2)*us(1)+k(1,3)*us(2)+k(1,4)*us(3)+q*h/2;
i=2:
                                                                                                                                                                                                 F2=k(3,1)*us(m-2)+k(3,2)*us(m-1)+k(3,4)*us(m)+q*h/2;
for i=1:nx-1
      xc(j,1)=i*h;
                                                                                                                                                                                                  %------TAGLIO------
                                                                                                                                                                                                 V=q*I/2-(q*x'); %Taglio esatto di una trave app-app
      j=j+1;
                                                                                                                                                                                                 v=zeros(nx+1,1);
end
                                                                                                                                                                                                v(1)=F1;
                                                                                                                                                                                                                                         %valore del taglio approssimato nel primo estremo
%Valori dei coefficienti hermitiani per il primo elemento[0,h]
                                                                                                                                                                                                v(nx+1)=-F2;
                                                                                                                                                                                                                                              %valore del taglio approssimato nell'ultimo estremo
N21=xc(1:2)-2*xc(1:2).^2/h+xc(1:2).^3/h^2;
N31=3*xc(1:2).^2/h^2-2*xc(1:2).^3/h^3;
                                                                                                                                                                                                 if nx>3
N41=-xc(1:2).^2/h+xc(1:2).^3/h^2;
                                                                                                                                                                                                       %valore del taglio approssimato nel primo nodo interno,calcolato come
                                                                                                                                                                                                       %appartente al primo elemento
%Valori dei coefficienti hermitiani per l'i-esimo elemento
                                                                                                                                                                                                       v(2)=(k(1,1)*us(2)+k(1,2)*us(3)+k(1,3)*us(4)+k(1,4)*us(5))+q*h/2;
N1=1-3*(xc(nx:nx+1)-xc(nx)).^2/h^2+2*(xc(nx:nx+1)-xc(nx)).^3/h^3;
                                                                                                                                                                                                 else
N2=(xc(nx:nx+1)-xc(nx))-2*(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx
                                                                                                                                                                                                       v(2)=(k(1,1)*us(2)+k(1,2)*us(3)+k(1,4)*us(4))+q*h/2;
xc(nx)).^3/h^2;
                                                                                                                                                                                                 end
N3=3*xc(1:2).^2/h^2-2*xc(1:2).^3/h^3;
                                                                                                                                                                                                i=3;p=2;
N4=-(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^3/h^2;
                                                                                                                                                                                                 for i=1:nx-2
                                                                                                                                                                                                       v(i)=-(k(3,1)*us(p)+k(3,2)*us(p+1)+k(3,3)*us(p+2)+k(3,4)*us(p+3))-q*h/2;
%Valori dei coefficienti hermitiani per l'ultimo elemento[nx,nx+1]
                                                                                                                                                                                                      j=j+1;
N1=1-3*(xc(nx:nx+1)-xc(nx)).^2/h^2+2*(xc(nx:nx+1)-xc(nx)).^3/h^3;
                                                                                                                                                                                                       p=p+2;
N2=(xc(nx:nx+1)-xc(nx))-2*(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx:nx+1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx)).^2/h+(xc(nx-1)-xc(nx
                                                                                                                                                                                                 end
```

```
%------MOMENTO------
Mo=(q^*I/2)^*x'-((q^*(x').^2)/2); %Momento esatto di una trave app-app
                                                                      %-----PLOTTAGGIO DELLE REAZIONI VINCOLARI-----
mo=zeros(nx+1,1);
                           %Momento approssimato
                                                                      ff1=num2str(F1);
mo(1)=0;
                       %momento approssimato nel primo estremo
                                                                      ff2=num2str(F2);
mo(nx+1)=0;
                         %momento approssimato nell'ultimo estremo
                                                                      text(0,-14*s,['R1= ',ff1,' N'],'HorizontalAlignment','left','FontSize',10);
                                                                      text((L-20*s),-14*s,['R2=',ff2,' N'],'HorizontalAlignment','left',...
if nx > = 3
                                                                                                          'FontSize',10);
    %momento approssimato nel primo nodo interno
                                                                      %----salvo i vettori restituiti dalla funzione nella struttura (vettori)
  mo(2)=-(k(2,1)*us(2)+k(2,2)*us(3)+k(2,3)*us(4)+k(2,4)*us(5))-q*h^2/12; handles.vettori.x=x;
                                                                      handles.vettori.u=u:
else
  mo(2)=-(k(2,1)*us(2)+k(2,2)*us(3)+k(2,4)*us(4))-q*h^2/12;
                                                                      handles.vettori.U=U:
                                                                      handles.vettori.v=v;
end
                                                                      handles.vettori.V=V:
                                                                      handles.vettori.mo=mo:
j=3;p=2;
                                                                      handles.vettori.Mo=Mo;
for i=1:nx-2
    %valore del momento per tutti i nodi interni
                                                                      handles.vettori.Er=Er:
  mo(i)=(k(4,1)^*us(p)+k(4,2)^*us(p+1)+k(4,3)^*us(p+2)+k(4,4)^*us(p+3))-
                                                                      handles.vettori.vinc1x=vinc1x:
q*h^2/12;
                                                                      handles.vettori.vinc1y=vinc1y;
                                                                      handles.vettori.vinc2x=vinc2x:
  j=j+1;
                                                                      handles.vettori.vinc2y=vinc2y;
  p=p+2;
                                                                      handles.vettori.strx=strx;
end
%------DISEGNO DELLA STRUTTURA------
                                                                      handles.vettori.stry=stry;
                                                                      %-----subfunction-----subfunction-----
L=I:
s=L/100;
                                                                      %----definizione della matrice di rigidezza del singolo elemento finito
vinc1x=[0,-3,-5,-5,5,5,-3,3,0].*s;
                                                                      function k=Getk(EI,h)
vinc1y=[0, -7, -7, -10, -10, -7, -7, -7, 0].*s;
                                                                      k(1,1)=12; k(1,2)=6*h; k(1,3)=-12;
                                                                                                           k(1,4)=6*h;
strx=[0,L];
                                                                      k(2,1)=6*h; k(2,2)=4*h^2; k(2,3)=-6*h; k(2,4)=2*h^2;
stry=[0,0];
                                                                      k(3,1)=-12; k(3,2)=-6*h; k(3,3)=12; k(3,4)=-6*h;
carx1=[L,L,L,L,L,L,L,L,L];
                                                                      k(4,1)=6*h; k(4,2)=2*h^2; k(4,3)=-6*h; k(4,4)=4*h^2;
carx2=[0,-3,-5,-5,5,-3,3,0].*s;
                                                                      k=EI/h^3*k;
vinc2x=carx1+carx2:
vinc2y=[0, -7, -7,-10,-10, -7, -7, -7,0].*s;
%-----PLOTTAGGIO DELLA STRUTTURA------
plot(vinc1x, vinc1y, 'k', vinc2x, vinc2y, 'k', strx, stry, 'b', 'LineWidth', 2);
axis([-10*s L+10*s -L/2 L/2]);
```

Scegliendo quindi dal popupmenu lo schema di trave App-App e lanciando la soluzione cliccando sul pulsante calcola, si otterrà il seguente risultato:



Dopo aver risolto la struttura con la funzione vista precedentemente dobbiamo creare le funzioni necessarie alla visualizzazione dei diagrammi delle sollecitazioni e della deformata.

Sono state scritte le seguenti funzioni:

- Plotdef.m per il plottaggio della deformata;
- Plotmom.m per il plottaggio della funzione momento;
- Plottagl.m per il plottaggio della funzione taglio;
- **Ploterr.m** per il plottaggio dell'errore tra la funzione spostamento esatta e quella approssimata.

#### Si riporta il listato della funzione **plotdef.m**:

```
function plotdef(handles)
%-----DATI-----
                                                                         %-----PLOTTAGGIO DELLA DEFORMATA------
                                                                         plot(x,u.*su,'g',x,U.*su,'m*:',vinc1x,vinc1y,'k',vinc2x,vinc2y,'k',strx,stry,'b');
x=handles.vettori.x;
u=handles.vettori.u:
                                                                         %settaggi del plot
U=handles.vettori.U;
                                                                         s=L/100:
                                                                         axis([-10*s L+10*s -L/2 L/2]);
L=handles.dati.L:
                                                                         lab=linspace(-L/2,L/2,9)/su;
su=handles.dati.su;
vinc1x=handles.vettori.vinc1x;
                                                                         lab1=linspace(-L/2,L/2,9);
                                                                         set(handles.axes1,'YTickLabel',lab');
vinc1y=handles.vettori.vinc1y;
                                                                         set(handles.axes1,'YTick',lab1');
vinc2x=handles.vettori.vinc2x;
vinc2y=handles.vettori.vinc2y;
                                                                        xlabel('x [mm]');
strx=handles.vettori.strx;
                                                                         vlabel('U [mm]');
stry=handles.vettori.stry;
                                                                         title('Deformata');
                                                                         legend('FEM','Exact',0);
%elimina il precedente plottaggio
axes(handles.axes1)
cla
```

Come si nota alla funzione plotdef (handles) viene passata la struttura handles nella quale sono state salvate le strutture (dati e vettori) contenenti i valori necessari al plottaggio della funzione spostamento.

In modo analogo sono state scritte le altre funzioni.

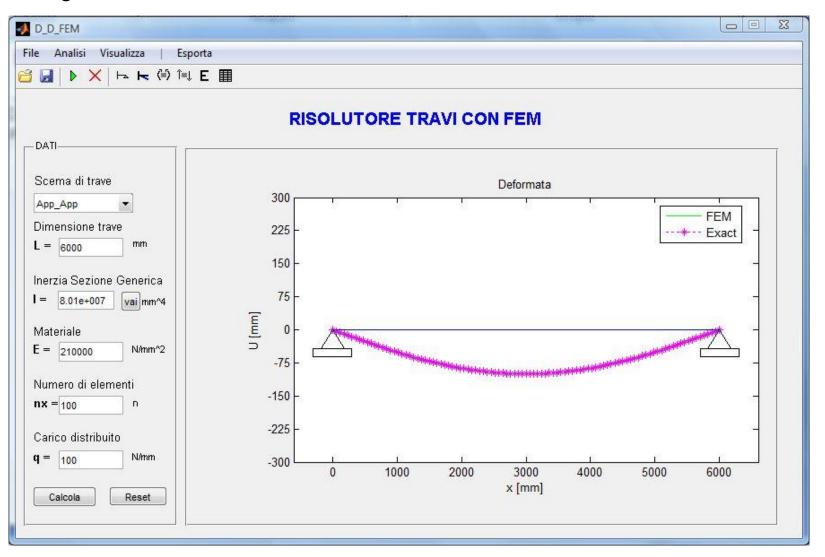
Ora non ci resta che associare ai pulsanti **Deformata, Momento, Taglio** ed **Errore** presenti nella barra degli strumenti e alle voci di **menu visualizza** le relative funzioni.

#### Si riporta il listato per la sola funzione plotdef:

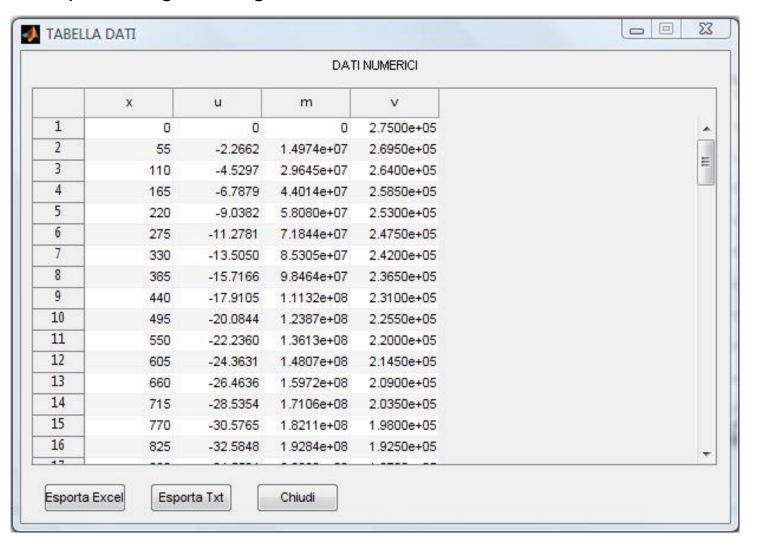
```
function TB_deformata_ClickedCallback(hObject, eventdata, handles)
% hObject handle to TB_deformata (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%richiamo la funzioe esterna (plotdef) e plotto la deformata della struttura
plotdef(handles);
```

```
function M_Deformata_Callback(hObject, eventdata, handles)
% hObject handle to M_Deformata (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)
%richiamo la funzioe esterna (plotdef) e plotto la deformata della struttura plotdef(handles);
```

Cliccando sul pulsante Deformata, ad esempio, per la trave vista precedentemente si ottiene il seguente risultato:



Per visualizzare la tabella dati clicchiamo sul pulsante **Tabella** della barra degli strumenti che apre la seguente figura:



La figura vista prima è stata realizzata senza l'ausilio dell'editor di GUI scrivendo manualmente la funzione **tabella dati.m**.

Come si può vedere la figura presenta il pulsante **Chiudi** che chiude questa finestra e ci riporta sulla finestra principale, ed i pulsanti **Esporta Txt** ed **Esporta Excel** che permettono l'esportazione dei dati nei rispettivi formati.

Si riporta di seguito il listato della funzione tabella\_dati.m:

```
function tabella dati(handles)
                                                                          chiudi = uicontrol(form1, 'Style', 'pushbutton', 'Position', [200, 10, 70, 25],...
%come si può vedere alla funzione deve essere passato il valore
                                                                                    'String', 'Chiudi', 'Callback', @chiudi plot);
%(handles)che contiene le indicazioni sulla struttura della figura
                                                                                    FUNZIONE PULSANTE ESPORTA EXCEL
%principale per poter riconosce gli oggetti contenuti in essa.
                                                                            function esporta_excel_plot(hObject,eventdata)
Mdati = handles.matrici.Mdati;
                                                                             [file,path] = uiputfile('animinit.xls','Save file name');
% INIZIALIZZAZIONE DEGLI OGGETTI GRAFICI
                                                                               if file==0
% FIGURA PRINCIPALE
                                                                                  return
form1=figure('Position',[400,300,600,400],'Name','TABELLA DATI'....
                                                                               end
  'NumberTitle', 'off', 'Color', [0.941 0.941 0.941],...
                                                                             pathfile=joinseq(path,file);
  'Resize', 'off', 'WindowStyle', 'modal');
                                                                             xlswrite(pathfile, Mdati);
% STATIC TEXT
                                                                            end
testo = uicontrol(form1,'Style','text','Position',[250,370,100,25],...
                                                                                       FUNZIONE PULSANTE ESPORTA TXT
                 'String','DATI NUMERICI');
                                                                            function esporta_txt_plot(hObject,eventdata)
                                                                             [file,path] = uiputfile('animinit.txt','Save file name');
              TABLE
cnames = {'x','u','m','v'};
                                                                               if file==0
table = uitable('Data',Mdati,'ColumnName',cnames,...
                                                                                  return
       'Parent', form 1, 'Position', [10 50 580 320]);
                                                                               end
                    PULSANTI
                                                                             pathfile=joinseq(path,file);
Esporta_Excel = uicontrol(form1,'Style','pushbutton',...
                                                                             dlmwrite(pathfile, Mdati);
          'Position',[20,10,70,25],'String','Esporta Excel',...
                                                                            end
          'Callback', @esporta_excel_plot);
                                                                                       FUNZIONE PULSANTE CHIUDI
Esporta_Txt = uicontrol(form1,'Style','pushbutton',...
                                                                            function chiudi_plot(hObject,eventdata)
          'Position',[110,10,70,25],...
                                                                              close
          'String', 'Esporta Txt', 'Callback', @esporta_txt_plot);
                                                                            end
                                                                          end
```

L'esportazione dei dati nei formati .xls e .txt può essere eseguita anche dal menu *Esporta* assegnando i seguenti blocchi di istruzioni alle funzioni Callback delle rispettive voci di menu *Excel* e *Text*.

Si riporta di seguito il listato delle due funzioni:

#### Esportazione Excel:

```
function M_Excel_Callback(hObject, eventdata, handles)
% hObject handle to M_Excel (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

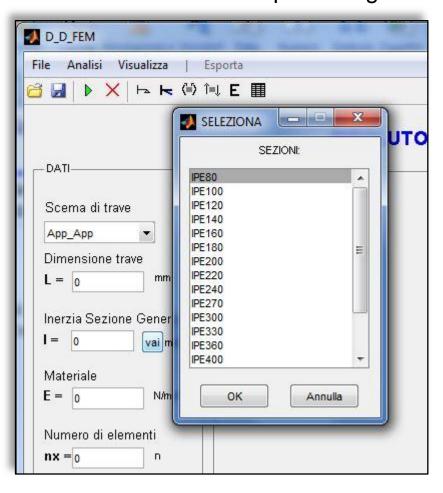
%esporto i dati in un file excel
[file,path] = uiputfile('animinit.xls','Save file name');
if file==0
return
end
pathfile=joinseq(path,file);
xlswrite(pathfile, handles.matrici.Mdati);
```

#### Esportazione Text:

```
function M_Text_Callback(hObject, eventdata, handles)
% hObject handle to M_Text (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%esporto i dati in un file di testo txt
[file,path] = uiputfile('animinit.txt','Save file name');
if file==0
return
end
pathfile=joinseq(path,file);
dlmwrite(pathfile, handles.matrici.Mdati);
```

Vediamo ora la funzione che è stata assegnata al pulsante vai. Cliccando su di esso si apre la seguente figura:



La finestra presenta una listbox contenente diversi tipi di sezioni ed i pulsanti OK ed Annulla

Cliccando sul pulsante OK si assegna alla edit text *Edit\_I* il valore dell'inerzia della sezione selezionata all'interno della listbox e si chiude la finestra.

Anche questa figura è stata realizzata senza l'ausilio dell'editor di GUI scrivendo manualmente la funzione **sezioni.m**. Al pulsante Vai sarà quindi associata la funzione sezioni.m come di seguito riportato:

```
function pushbutton_section_Callback(hObject, eventdata, handles)
% hObject handle to pushbutton_section (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

%_____
%richiamo il file.m che contiene l'elenco delle sezioni
sezioni(handles);
```

#### Vediamo ora il listato della funzione **sezioni.m**:

```
function sezioni(handles)
%come si può vedere alla funzione deve essere passato il valore
                                                                              %PULSANTI
%(handles)che contiene le indicazioni sulla struttura della figura
                                                                             B ok = uicontrol(form1,'Style','pushbutton',...
%principale per poter riconosce gli oggetti contenuti in essa.
                                                                                        'Position',[20,10,70,25],...
                                                                                        'String','OK',...
                                                                                        'Callback', @OK plot);
%carico la lista delle sezioni da un file di testo .txt creato
%precedentemente
                                                                             B annulla = uicontrol(form1,'Style','pushbutton',...
[Vsez1,Vsez2] = textread('SEZ.txt','%s %s');
                                                                                        'Position',[110,10,70,25],...
                                                                                        'String','Annulla',...
                                                                                        'Callback', @annulla plot);
%definizione della figura che contine gli oggetti
form1=figure(...
  'Position',[600,300,200,280],...
                                                                               function OK plot(hObject, eventdata)
  'Name', 'SELEZIONA',...
                                                                                  %otteniamo la posizione dell'elemento selezionato
  'NumberTitle','off',...
                                                                                 sez=get(lista sez,'Value');
                                                                                  %assegniamo alla cella della figura principale EDIT I il valore
  'Color',[0.941 0.941 0.941],...
  'Resize','off',...
                                                                                  %numerico corrispondente.
                                                                                  %come si può vedere è stato richiamato l'oggetto handles
  'WindowStyle', 'modal');
                                                                                 set(handles.TEX_sezione,'String',Vsez1(sez));
                                                                                 set(handles.EDIT I,'String', Vsez2(sez));
%STATIC TEXT
                                                                                 close
ST TEXT = uicontrol(form1,'Style','text',...
                                                                               end
          'Position',[10,250,180,25],...
          'String', 'SEZIONI:');
                                                                               function annulla plot(hObject, eventdata)
%LIST BOX
                                                                                 close
lista sez = uicontrol(form1,'Style','listbox',...
                                                                               end
          'Position',[10 50 180 200],...
          'Backgroundcolor',[1.0 1.0 1.0],...
                                                                             end
          'String', Vsez1);
```

Cliccando sul pulsante *Salva*, presente nella barra degli strumenti, è possibile salvare i dati inseriti dall'utente in un file .txt grazie alla seguente finestra:



Nella quale possiamo nominare il file e salvarlo nella directory desiderata.

Tale finestra viene attivata tramite la funzione

```
[file,path] =
uiputfile('animinit.tx
t','Save file name').
```

Si riporta il listato della funzione salva:

```
function salva(handles)
%salvataggio dei valori
Vdati=[handles.dati.L;...
     handles.dati.E;...
     handles.dati.l:...
     handles.dati.nx:...
     handles.dati.q;...
     handles.dati.su:...
     handles.dati.sm:...
     handles.dati.sv;...
     handles.dati.strutt];
%apro la classica finestra di windows (salva file)per ottenere la stringa
%(file=nomefile.txt e path=percorso completo)
[file,path] = uiputfile('animinit.txt','Save file name');
%se la finestra viene chiusa si blocca la funzione
if file==0
  return
end
%ottengo il percorso completo più nome con la funzione joinseq
pathfile=joinseq(path,file);
%scrivo il vettore colonna (Vdati) nel file.txt delimitato con (dlmwrite)
dlmwrite(pathfile, Vdati);
```

Cliccando sul pulsante *Apri*, presente nella barra degli strumenti, è possibile aprire file precedentemente salvati grazie alla seguente finestra:



Nella quale possiamo selezionare il file.txt ed aprirlo.

### Tale finestra viene attivata tramite la funzione

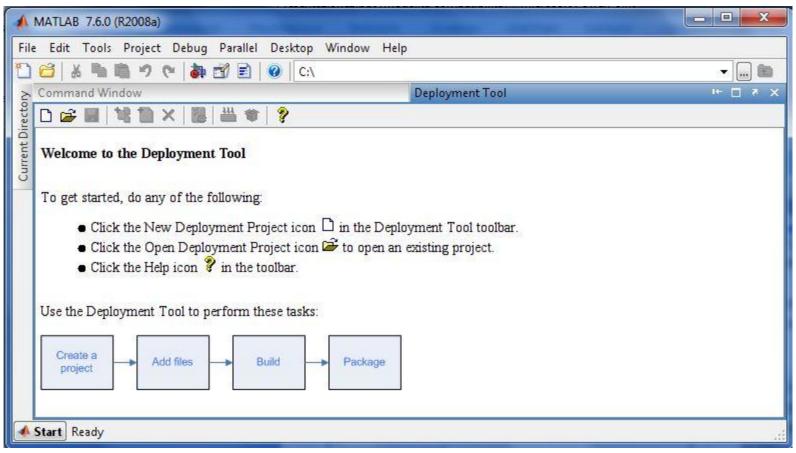
[FileName, PathName] =
uigetfile('\*.txt', 'Sel
eziona file').

Si riporta di seguito il listato della funzione Apri:

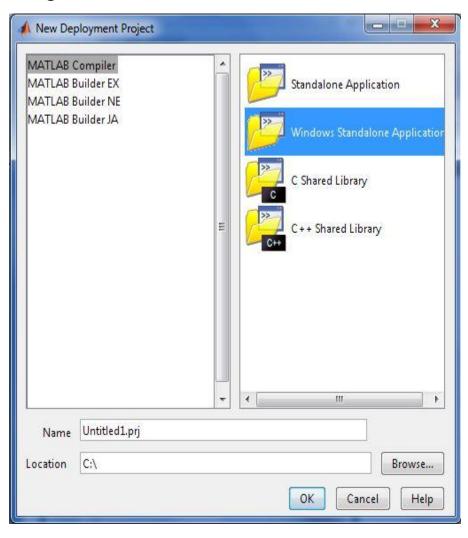
```
function apri(handles)
%apro la classica finestra (apri file) e ottengo la due stringhe contenenti
%il nome del file e il percorso completo
[FileName, PathName] = uigetfile('*.txt', 'Seleziona file');
%unisco le due stringhe in un unica stringa con il comando join
pathfile=joinseq(PathName,FileName);
%copio i dati scritti nel file nella variabile (inVdati)
inVdati = dlmread(pathfile);
%assegno i valori letti dal file alle corrispondenti variabili interne
handles.dati.L=inVdati(1,1);
handles.dati.E=inVdati(2,1);
handles.dati.l=inVdati(3,1);
handles.dati.nx=inVdati(4,1);
handles.dati.q=inVdati(5,1);
%fattore di scala per i diagrammi
handles.dati.su=inVdati(6,1);
handles.dati.sm=inVdati(7,1);
handles.dati.sv=inVdati(8,1);
handles.dati.strutt=inVdati(9,1);
guidata(handles.figure1,handles);
%assegno alle edit text il valore caricato
set(handles.popup schematrave, 'Value', handles.dati.strutt);
set(handles.EDIT L,'String',handles.dati.L);
set(handles.EDIT_E,'String',handles.dati.E);
set(handles.EDIT I,'String',handles.dati.I);
set(handles.EDIT nx,'String',handles.dati.nx);
set(handles.EDIT q,'String',handles.dati.q);
```

Dopo aver completato la struttura del del programma la cosa che più ci interessa è quella di poter distribuire il nostro progetto a utenti che non hanno il programma Matlab installato sul proprio computer.

Per realizzare il file eseguibile (.exe) da Matlab clicchiamo il pulsante (<u>Start -></u> <u>MATLAB -> MATLAB Compiler -> Deployment Tool</u>). Si avvia in questo modo la seguente finestra:



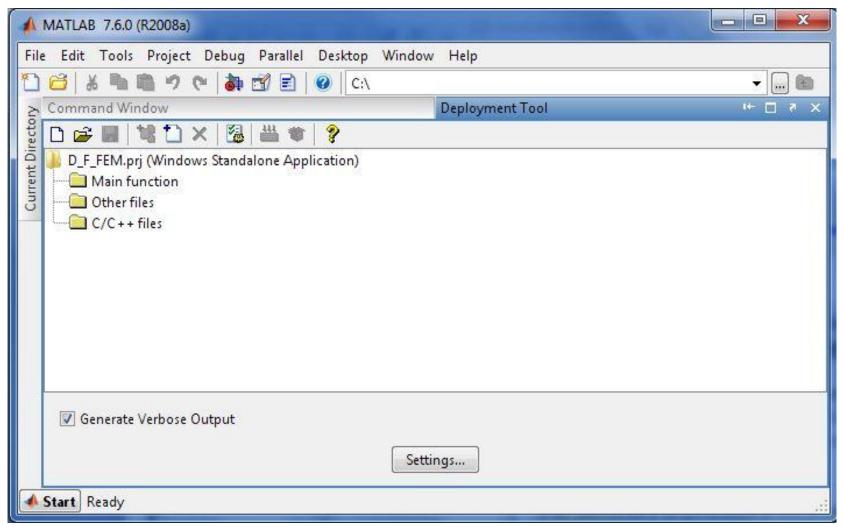
Dalla barra dei menù clicchiamo (*File -> New Deployment Project*). Si aprirà la seguente finestra.



Nella finestra a destra selezioniamo (*Windows Standalone Application*).

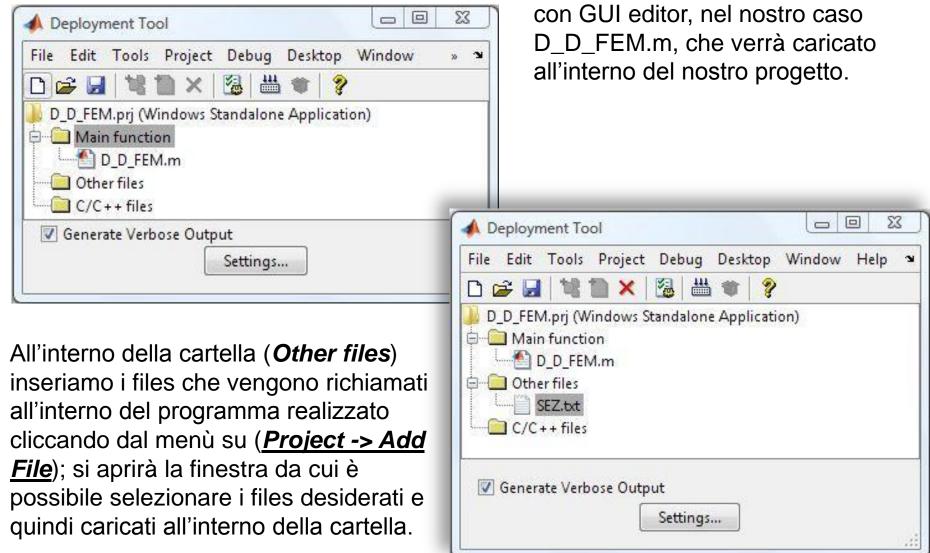
Digitiamo il nome del nostro progetto nella cella (*Name*) e selezioniamo il percorso di salvataggio cliccando sul pulsante (**Browse...**).

Per terminare la procedura clicchiamo sul pulsante (**OK**) che chiude questa finestra e ci riporta sull'area di lavoro (**Deployment Tool**) che si presenta come segue:

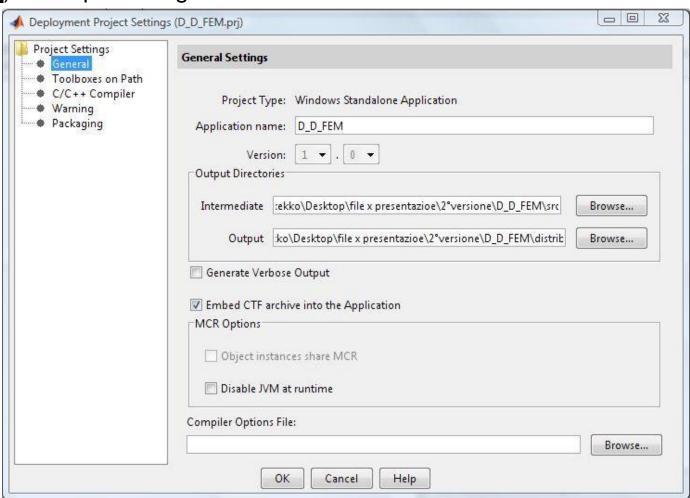


Come si può notare Matlab ha predisposto diverse cartelle che conterranno i file necessari per la realizzazione dell'eseguibile.

Selezioniamo la cartella (*Main function*) e dal menù clicchiamo su (*Project -> Add File*) e si apre la classica finestra di apertura. Selezioniamo il file.m realizzato

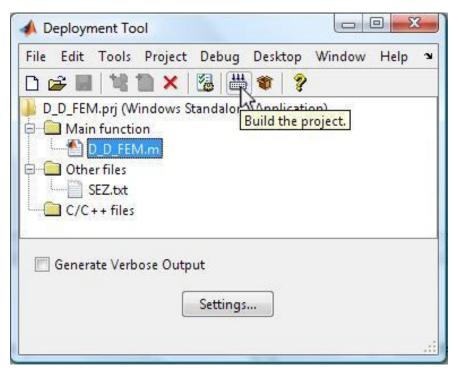


Prima di procedere con la compilazione del progetto clicchiamo su (*Project -> Settings*) che apre la seguente finestra:

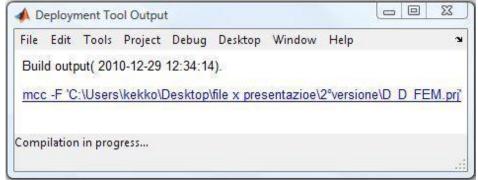


all'interno della quale è possibile impostare diversi parametri. Per il momento lasciamo i parametri di default.

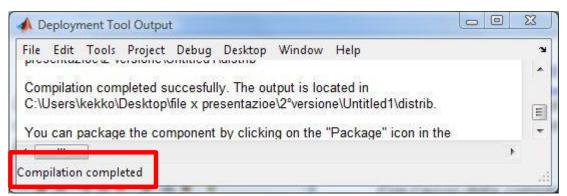
Procediamo ora alla compilazione del file eseguibile. Nella finestra *Deployment Tool* clicchiamo sull'icona *Build the project* presente nella barra degli strumenti.



Con l'avvio della compilazione, la quale durerà alcuni minuti, si apre la finestra *Deployment Tool Output* 



Al termine della compilazione, la finestra **Deployment Tool Output** apparirà come in figura



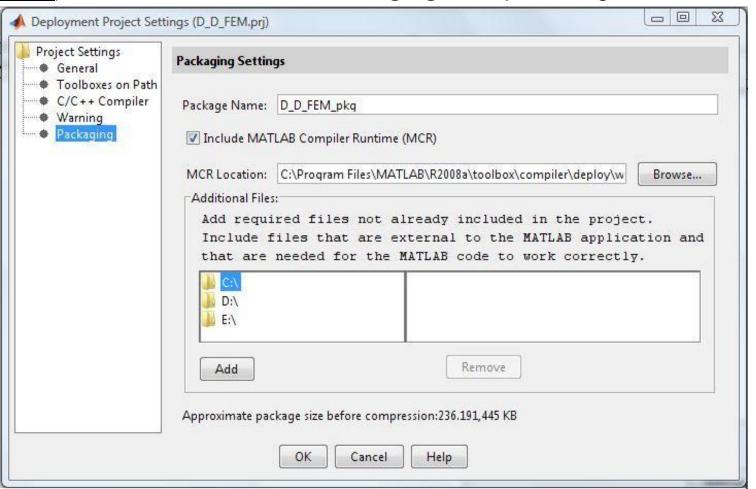
Per poter distribuire la nostra applicazione e quindi eseguirla su computer in cui non è istallato Matlab bisogna creare un pacchetto di istallazione che comprenda oltre al nostro progetto anche il file MCR.exe.

Il Matlab Compiler Runtime è la versione ridotta di Matlab necessaria per l'esecuzione dei calcoli. Questo file è fornito con MATLAB Compiler e può essere distribuito a titolo gratuito.

Per maggiori informazioni visitare il sito:

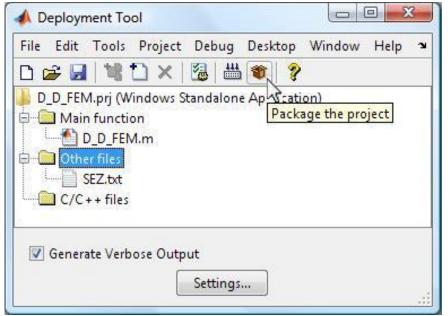
http://www.mathworks.com/products/compiler/?BB=1.

Al fine di includere l'MCR.exe nel pacchetto di distribuzione clicchiamo su (*Project* -> *Settings*) e selezioniamo la voce *Packaging* che apre la seguente finestra:

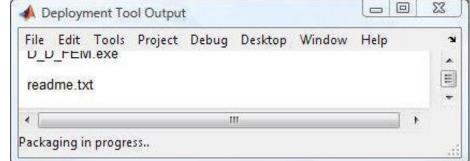


Nella parte destra spuntiamo la voce *Include MATLAB Compiler Runtime (MCR)* e controlliamo che il percorso riportato nella cella *MCR Location* sia corretto.

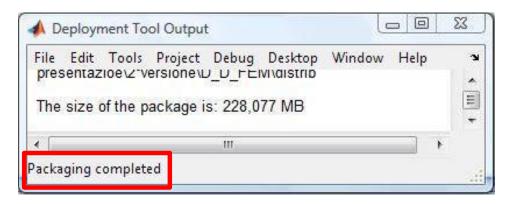
Ora possiamo creare il nostro pacchetto di istallazione cliccando sull'icona *Package the project* nella barra degli strumenti del *Deployment Tool*.



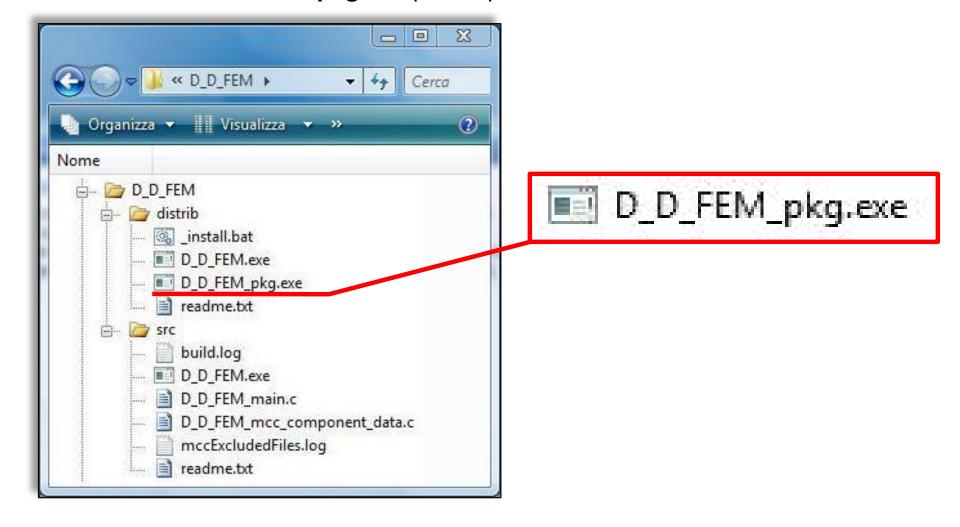
Anche questo processo può durare alcuni minuti.



Al termine della creazione del pacchetto, la finestra **Deployment Tool Output** apparirà come in figura



Durante le operazioni di compilazione Matlab crea la cartella **D\_D\_FEM** nella quale troviamo altre due cartelle denominate **distrib** e **src.** All'interno della cartella **distrib** troveremo il file **D\_D\_FEM\_pkg.exe** pronto per essere distribuito.

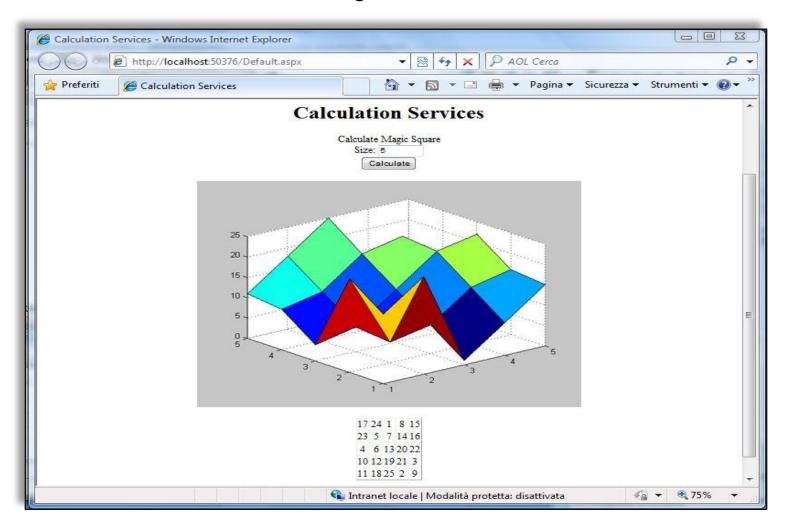




### Realizzazione di un'applicazione WEB

Grazie al compiler di Matlab si possono realizzare dei file.dll (dinamic link library) a partire da un file.m.

Queste librerie contengono delle classi utilizzabili per la realizzazione di pagine WEB dinamiche in formato ASPX come la seguente.



### Realizzazione di un'applicazione WEB

La pagina vista prima è stata realizzata con il compilatore Visual Studio 2008 in linguaggio C# il quale utilizza una dll realizzata con Matlab.

Il codice necessario per realizzare la pagina vista prima è reperibile nella guida in linea di Matlab ricercando "Matlab Builder NE" oppure seguendo il link <a href="http://www.mathworks.com/access/helpdesk/help/pdf\_doc/compiler/example\_guide.pdf">http://www.mathworks.com/access/helpdesk/help/pdf\_doc/compiler/example\_guide.pdf</a>.

### **Bibliografia**

Berardino D'Acunto. Computational Methods for PDE in Mechanics
Thomas Holland, Patrick Marchand. Graphics and GUIs with MATLAB (third edition)
www.mathworks.it

### **Appendice**

Si riporta di seguito i file.m che compongono il programma D\_D\_FEM:

- 1. D\_D\_FEM.m
- 2. tabella\_dati.m
- 3. sezioni.m
- 4. appapp.m
- 5. incapp.m
- 6. incinc.m
- 7. inclib.m
- 8. plotdef.m
- 9. plotmom.m
- 10. plotagl.m
- 11. ploterr.m