

SAPLAB

Structural Analysis Program Laboratory

Tool sperimentale di MATLAB (Matrix Laboratory) per il calcolo, la simulazione e l'analisi agli elementi finiti di semplici sistemi strutturali.

Annexo alla Tesi di Laurea Magistrale in Ingegneria Civile

IL METODO AGLI ELEMENTI FINITI
APPLICAZIONI DI MECCANICA COMPUTAZIONALE
PER L'ANALISI DI STRUTTURE PER L'INGEGNERIA CIVILE

Anno Accademico 2021 - 2022

SAPLAB 1.0 – Manuale d'uso

SAPLAB 1.0 – MANUALE D'USO

FIDUCIA GABRIELE – S265128



Indice degli argomenti

INDICE DEGLI ARGOMENTI	1
INTRODUZIONE.....	5
REQUISITI MINIMI DI SISTEMA.....	5
VERSIONE CONSIGLIATA DI MATLAB.....	6
TOOL AGGIUNTIVI.....	6
AVVIO DELL'APPLICAZIONE	7
CREARE UN MODELLO SAPLAB	7
SALVARE UN MODELLO SAPLAB.....	8
APRIRE UN MODELLO SAPLAB	8
VERIFICARE SE UN FILE .MAT CONTIENE UN MODELLO SAPLAB.....	9
ESECUZIONE DELL'INTERFACCIA UTENTE.....	9
DESCRIZIONE DELL'INTERFACCIA UTENTE	10
MODELLAZIONE DELLA GEOMETRIA	13
GEOMETRIA 1D.....	14
<i>AGGIUNGERE UN PUNTO ALLA GEOMETRIA.....</i>	<i>14</i>
<i>AGGIUNGERE UNA LINEA ALLA GEOMETRIA DEL MODELLO.....</i>	<i>15</i>
<i>DIVIDERE UNA LINEA IN PIÙ SEGMENTI.....</i>	<i>16</i>
<i>DIVIDERE UNA LINEA IN UN PUNTO SPECIFICO.....</i>	<i>17</i>
GEOMETRIA 2D.....	18
<i>AGGIUNGERE UNA SUPERFICIE RETTANGOLARE</i>	<i>19</i>
<i>AGGIUNGERE UNA SUPERFICIE CIRCOLARE</i>	<i>19</i>
<i>AGGIUNGERE UNA SUPERFICIE ELLITTICA.....</i>	<i>20</i>
<i>AGGIUNGERE UN POLIGONO GENERICO.....</i>	<i>21</i>
<i>DIVIDERE IL BORDO DI UNA SUPERFICIE IN PIÙ SEGMENTI.....</i>	<i>22</i>
<i>COMPOSIZIONE DELLE PRIMITIVE GEOMETRICHE</i>	<i>22</i>
<i>IMPORTARE LA GEOMETRIA 2D DI DEFAULT</i>	<i>24</i>
ASSEGNARE LE PROPRIETÀ STRUTTURALI	27

ASSEGNARE LE PROPRIETÀ MECCANICHE DEL MATERIALE	27
RICERCA DELLE PROPRIETÀ DEI MATERIALI ASSEGNATI	28
ASSEGNARE LE PROPRIETÀ GEOMETRICHE DELLE SEZIONI	29
RICERCA DELLE PROPRIETÀ GEOMETRICHE DELLE SEZIONI ASSEGNATE	30
ASSEGNARE I CARICHI AGLI ELEMENTI STRUTTURALI.....	31
APPLICARE CARICHI DI TIPO <i>DISTRIBUTED</i>	31
RICERCA DEI CARICHI DI TIPO <i>DISTRIBUTED</i> ASSEGNATI	34
APPLICARE I CARICHI DI TIPO <i>BODY</i>	34
RICERCA DEI CARICHI DI TIPO <i>BODY</i> ASSEGNATI	36
CONDIZIONI AL CONTORNO	37
ASSEGNARE LE CONDIZIONI AL CONTORNO.....	37
RICERCA DELLE CONDIZIONI AL CONTORNO ASSEGNATE	43
GLI END-RELEASE	45
ASSEGNARE GLI END-RELEASE	45
RICERCA DEGLI END-RELEASE.....	46
DISCRETIZZAZIONE DELLA GEOMETRIA	49
GENERAZIONE DELLA MESH.....	49
MESH REFINEMENT	50
RICERCA DELLE PROPRIETÀ DELLA MESH	52
FUNZIONE <i>saplab.findElementsInMesh()</i>	52
FUNZIONE <i>saplab.findNodesInMesh()</i>	53
FUNZIONE <i>saplab.findFRPInMesh()</i>	54
ANALISI FEM CON ELEMENTI FRP.....	55
CREARE UN RINFORZO STRUTTURALE.....	55
ASSEGNARE LE PROPRIETÀ DEL MATERIALE DEL FIBRORINFORZO	56
CREARE UNA INTERFACCIA DI COLLEGAMENTO.....	57
MODELLARE LE PROPRIETÀ MECCANICHE DELL'INTERFACCIA DI COLLEGAMENTO	59
VISUALIZZAZIONE DELLA GEOMETRIA.....	65
ESEMPIO 1 – PLOT GEOMETRIA (STATIC-PLANEFRAME/SPACEFRAME)	65
ESEMPIO 2 – PLOT LABELS ESEMPIO 1.....	67

ESEMPIO 3 – PLOT GEOMETRIA (STATIC-PLANESTRESS/PLANESTRAIN).....	68
ESEMPIO 4 – PLOT LABELS ESEMPIO 3	69
ESEMPIO 5 – PLOTTARE UNA GEOMETRIA CSG	71
ESEMPIO 6 – PLOT GEOMETRIA (COUPLED-STATIC-PLANESTRESS).....	72
ESEMPIO 7 – PLOT LABELS ESEMPIO 6	73
VISUALIZZAZIONE DELLA MESH	75
PLOTTARE LA MESH DI UN MODELLO SAPLAB	75
<i>ESEMPIO DI VISUALIZZAZIONE DELLA MESH (GEOMETRIA 1D)</i>	76
<i>ESEMPIO DI VISUALIZZAZIONE DELLA MESH (GEOMETRIA 2D)</i>	78
<i>ESEMPIO DI VISUALIZZAZIONE DELLA MESH (GEOMETRIA 2D E RINFORZO STRUTTURALE)</i>	79
SOLUZIONE DEL MODELLO FEM.....	83
RISOLUTORE FEM: ELINSOLVER()	83
RISOLUTORE FEM: NLINSOLVER()	83
POST-PROCESSING DEI RISULTATI.....	87
LA FUNZIONE RPLOT().....	87
LA SOLUZIONE DEL MODELLO FEM – GLI SPOSTAMENTI NODALI	88
<i>PLOTTARE GLI SPOSTAMENTI NODALI</i>	89
<i>LE SOLUZIONI PER ELEMENTO FINITO</i>	90
I DIAGRAMMI DI SOLLECITAZIONE	92
<i>PLOTTARE IL DIAGRAMMA DI SFORZO NORMALE</i>	92
<i>PLOTTARE IL DIAGRAMMA DELLO SFORZO DI TAGLIO</i>	93
<i>PLOTTARE IL DIAGRAMMA DEL MOMENTO FLETTENTE</i>	93
<i>PLOTTARE IL DIAGRAMMA DEL MOMENTO TORCENTE</i>	93
STRESS/STRAIN PLOT: DIAGRAMMARE TENSIONI E DEFORMAZIONI.....	93
LE TENSIONI DI CONTATTO TRA RINFORZO STRUTTURALE IN FRP E TRAVE.....	94

Introduzione

SAPLAB (acronimo di *Structural Analysis Program Laboratory*) è un *tool* di MATLAB (Matrix Laboratory) per il calcolo, la simulazione e l'analisi agli elementi finiti di semplici sistemi strutturali.

Il software, scritto in linguaggio MATLAB®, è stato realizzato nell'ambito del progetto di Tesi di Laurea Magistrale in Ingegneria Civile "IL METODO AGLI ELEMENTI FINITI - APPLICAZIONI DI MECCANICA COMPUTAZIONALE PER L'ANALISI DI STRUTTURE PER L'INGEGNERIA CIVILE" nell'Anno Accademico 2021-2022 (Relatore: Prof. Ing. Giorgio Zavarise e Correlatore: Prof. Ing. Pietro Cornetti).

All'interno del presente capitolo introduttivo:

- [I requisiti minimi di sistema;](#)
- [I tool aggiuntivi;](#)
- [La versione consigliata di MATLAB®.](#)

Per un utilizzo ottimale dell'applicazione è necessario avere una sufficiente conoscenza dell'ambiente MATLAB®. Pertanto, si rimanda ad uno studio preventivo della documentazione fornita dalla MathWorks® (Vedi link: [Help Center for MATLAB, Simulink and other MathWorks products](#)).

Per poter utilizzare correttamente le funzioni implementate e descritte nel presente testo si consiglia la lettura della documentazione fornita al seguente link [Add optional name-value pair argument into input parser scheme - MATLAB addParameter - MathWorks Italia](#)).

REQUISITI MINIMI DI SISTEMA

I requisiti minimi di sistema per un corretto funzionamento del programma sono quelli specificati dalla MathWorks®. È possibile visualizzare tali informazioni ai link elencati di seguito:

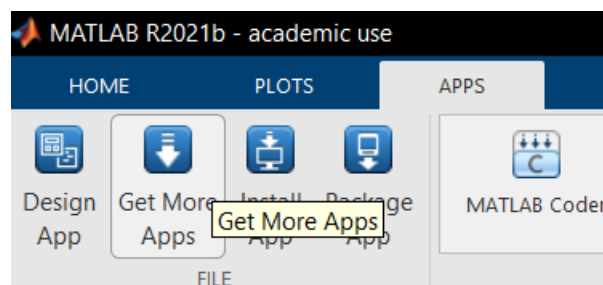
- OS: Windows® [system-requirements-release-2021a-windows.pdf \(mathworks.com\)](#);
- OS: Linux® [system-requirements-release-2021a-linux.pdf \(mathworks.com\)](#);
- OS: Mac OS® [system-requirements-release-2021a-macintosh.pdf \(mathworks.com\)](#).

VERSIONE CONSIGLIATA DI MATLAB

Si consiglia di eseguire l'applicazione SAPLAB utilizzando una versione di MATLAB® non inferiore a R2020a (Vedi link: [R2020a - Aggiornamenti alle famiglie dei prodotti MATLAB e Simulink - MATLAB & Simulink \(mathworks.com\)](#)) in quanto versioni più obsolete potrebbero essere in possesso di funzionalità indispensabili per la sua corretta esecuzione.

TOOL AGGIUNTIVI

Alcune funzionalità del programma SAPLAB richiamano delle *routine* del *tool* Partial Differential Equation Toolbox (PDE Toolbox - [Partial Differential Equation Toolbox - MATLAB \(mathworks.com\)](#)) scaricabile dall' *Add-On Explorer* (Disponibile nella Tab **APPS** alla voce **Get More Apps**) una volta installato MATLAB.



I - Apri Add-On Explorer cliccando su Get More Apps

Avvio dell'applicazione

Nel presente capitolo vengono esposte le operazioni introduttive all'utilizzo del *tool* SAPLAB. SAPLAB dispone di una interfaccia utente appositamente programmata per una gestione *user-friendly* dei codici di calcolo. Ciò non impedisce all'utente di utilizzare l'applicazione da riga di comando mediante *script*, *live-script* o *command window*.

All'interno del presente capitolo:

- [Creare un modello SAPLAB](#);
- [Salvare un modello SAPLAB](#);
- [Aprire un modello SAPLAB esistente](#);
- [Verificare se un file .mat contiene un modello SAPLAB](#);
- [Esecuzione dell'interfaccia utente](#);
- [Descrizione dell'interfaccia utente](#).

CREARE UN MODELLO SAPLAB

Avviare MATLAB®. Aprire un nuovo *file script* (Ctrl+N) o *live script* (Disponibile nella Tab *HOME* alla voce *New Live Script*). L'esecuzione dell'applicazione SAPLAB da riga di comando viene effettuata attraverso la funzione:

```
sapLab.create(_).
```

Tale procedura crea un oggetto appartenente alla classe `sap.StructuralModel`. Per maggiori informazioni digitare il comando:

```
sapLab.example('createModelsExample')
```

Sintassi e Descrizione

SINTASSI 1	<pre>structuralModel = sapLab.create();</pre> <p>Crea un nuovo modello strutturale SAPLAB. non specificando alcun argomento di <i>input</i>, la funzione genera un modello SAPLAB per analizzare elementi trave alla Eulero-Bernoulli nel piano XY.</p>
SINTASSI 2	<pre>structuralModel = sapLab.create('attributes', analysisType);</pre> <p>Crea un nuovo modello strutturale SAPLAB con analisi specificata dall'argomento di <i>input</i> analysisType. La tipologia di analisi da effettuare viene specificata negli argomenti di <i>input</i> <i>Name-Value</i>.</p>

SALVARE UN MODELLO SAPLAB

Una volta creato un modello SAPLAB è possibile salvarlo all'interno di un file .mat (Vedi [MAT-File Versions - MATLAB & Simulink - MathWorks Italia](#)).

La funzione attraverso cui effettuare il salvataggio del modello SAPLAB è:

```
sapLab.save(_);
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('saveModelsExample')
```

Sintassi e Descrizione

SINTASSI 1	<pre>sapLab.save(structuralModel);</pre> <p>Salva il modello SAPLAB senza specificare il nome del file di destinazione .mat. Non effettuando quest'ultima operazione il file di destinazione .mat prenderà automaticamente il nome 'untitled.mat'.</p>
SINTASSI 2	<pre>sapLab.save(structuralModel, 'filename', filename);</pre> <p>Salva il modello SAPLAB structuralModel specificando il nome del file di destinazione .mat.</p>

APRIRE UN MODELLO SAPLAB

L'apertura di un modello SAPLAB salvato all'interno di un file .mat viene effettuata attraverso la funzione:

```
sapLab.open(_)
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('openModelsExample')
```

Sintassi e Descrizione

SINTASSI 1	<pre>Model = sapLab.open('filename.mat');</pre> <p>Apri un file SAPLAB esistente il cui nome è specificato nell'argomento di input filename. Se il file .mat non contiene un modello strutturale SAPLAB allora la funzione genererà in output un errore. Ricordarsi che oltre al nome del file è necessario specificare anche l'estensione .mat.</p>
------------	--

VERIFICARE SE UN FILE .MAT CONTIENE UN MODELLO SAPLAB

È possibile effettuare tale verifica digitando la funzione:

```
saplab.filechk(_)
```

Per maggiori informazioni digitare il comando:

```
saplab.example('checkInMatFile')
```

Sintassi e Descrizione

	<pre>tf = saplab.filechk('filename.mat')</pre>
	Verifica se il file .mat specificato nell'argomento di input filename contiene un modello strutturale SAPLAB. In caso affermativo la funzione ritornerà in output:
SINTASSI 1	<pre>tf = logical(1)</pre>
	viceversa, la funzione ritornerà in output il valore:
	<pre>tf = logical(0).</pre>
	N.B. È necessario specificare anche l'estensione .mat.

ESECUZIONE DELL'INTERFACCIA UTENTE

È possibile eseguire SAPLAB non solo da riga di comando ma anche attraverso l'interfaccia utente. Per eseguirla digitare la funzione:

```
saplab.ui(_)
```

Per maggiori informazioni digitare il comando:

```
saplab.example('executeSapLabUI')
```

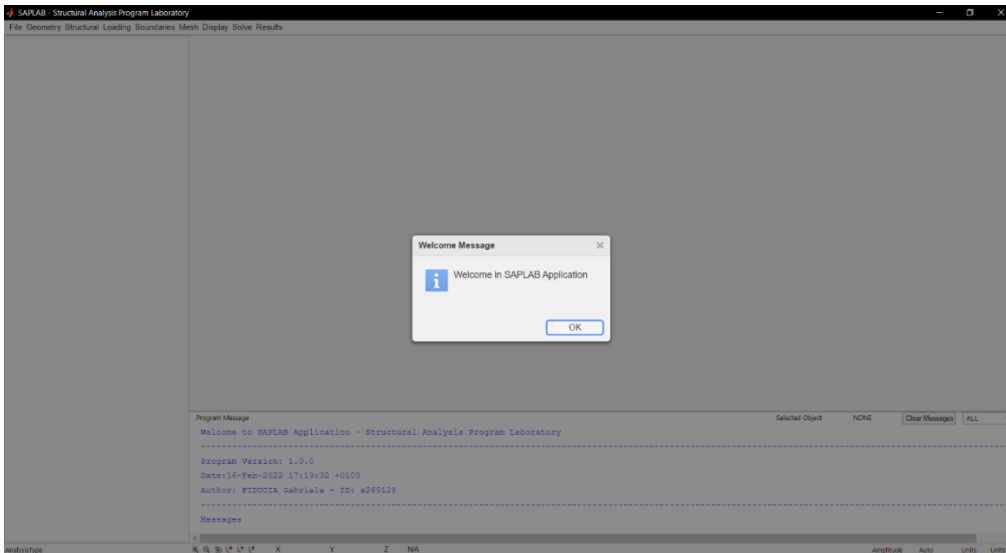
Sintassi e Descrizione

SINTASSI 1	<pre>saplab.ui();</pre> <p>Apri l'interfaccia utente SAPLAB.</p>
SINTASSI 2	<pre>saplab.ui(structuralmodel);</pre> <p>Apri l'interfaccia utente caricando il modello esistente specificato dall'argomento structuralmodel.</p>
SINTASSI 3	<pre>saplab.ui('filename.mat');</pre> <p>Visualizza il modello contenuto all'interno del file .mat filename nell'interfaccia utente SAPLAB.</p>

DESCRIZIONE DELL'INTERFACCIA UTENTE

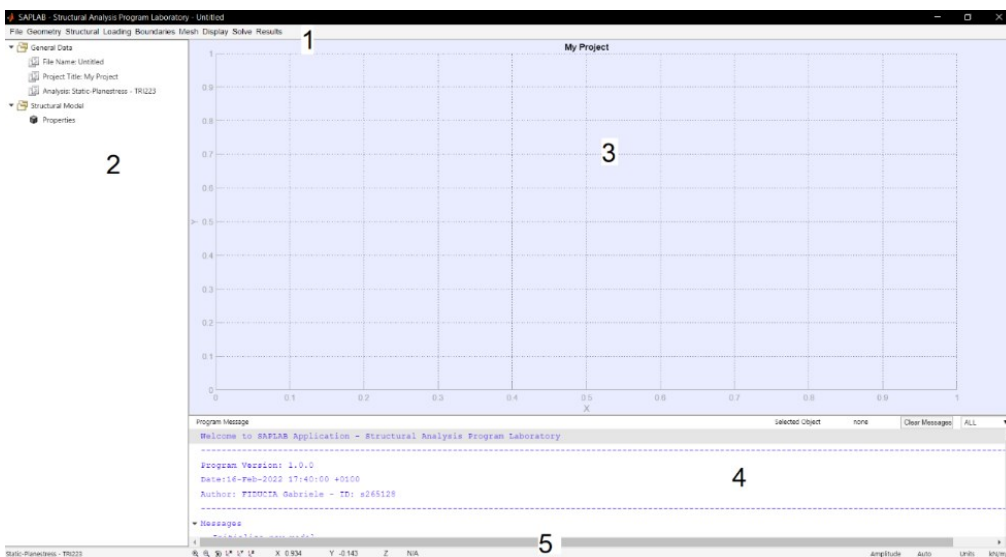
Una volta eseguito il comando [*saplab.ui\(\)*](#) attendere qualche secondo.

MATLAB® aprirà l'interfaccia utente di SAPLAB (Vedi figura) programmata attraverso il *tool* MATLAB App Designer (Vedi Link: [MATLAB App Designer - MATLAB & Simulink \(mathworks.com\)](#)) che consente di creare applicazioni trascinando e rilasciando i componenti presenti in libreria, programmandone rapidamente il comportamento attraverso l'editor integrato.



2 - Apertura dell'interfaccia utente di SAPLAB

L'interfaccia è composta da cinque componenti principali che sono la *Menu Bar* (1), il *Model Browser* (2), il *Canvas* (3), la *Program Message Area* (4) e la *Bottom Bar* (5).

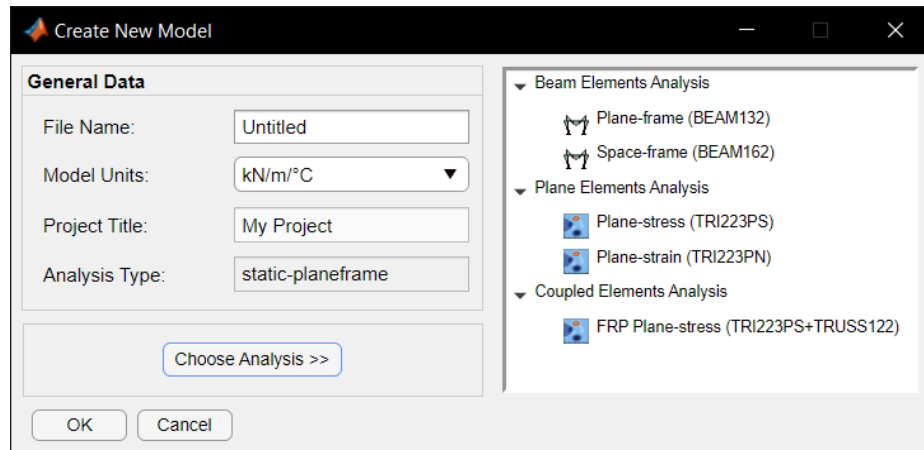


3 - Componenti principali dell'interfaccia utente

Le operazioni di creazione, apertura e salvataggio, descritte in precedenza, possono essere eseguite cliccando sulla voce *File* presente nella *Menu Bar*.

Cliccando su:

- *File* → *New* (Ctrl+N), l'interfaccia richiama la funzione [saplab.create\(\)](#) che viene tradotta dalla *form* **Create New Model** sotto raffigurata. La *form* consente di inserire il nome del file, le unità di misura del modello, il titolo del progetto e la tipologia di analisi (**Select Analysis**):



4 - Create New Model

- *File* → *Open* (Ctrl+O), l'interfaccia richiama la funzione [saplab.open\(\)](#);
- *File* → *Save* (Ctrl+S), l'interfaccia richiama la funzione [saplab.save\(\)](#).

Modellazione della geometria

La modellazione della geometria della struttura in SAPLAB viene eseguita attraverso l'utilizzo di apposite funzioni. Queste richiamano i *methods*¹ (Vedi link: [Methods - MATLAB & Simulink - MathWorks Italia](#)) delle due classi sottoelencate.

- *sap.GeometricModel1D*: Un oggetto corrispondente a tale classe consente la modellazione di geometrie per effettuare analisi su elementi finiti di tipo *beam* (*Plane-frame* e *Space-frame analysis*).
- *sap.GeometricModel2D*: Un oggetto corrispondente a tale class consente la modellazione di geometrie per effettuare analisi su elementi finiti piani (*Plane-stress*, *Plane-strain* e *Coupled-Plane-Stress analysis*).

Entrambi gli oggetti su citati descrivono la geometria della struttura secondo entità discrete denominate *regions*. Quest'ultime vengono definite nella tabella seguente.

Tabella I - Regioni Geometriche: Definizione e caratteristiche

Oggetto	Tipo	Dimensione	Regions	Label
<i>sap.GeometricModel1D</i>	Punto	0D	'Point'	'P'
	Linea	1D	'Line'	'L'
<i>sap.GeometricModel2D</i>	Vertice	0D	'Vertex'	'V'
	Linea	1D	'Edge'	'E'
	Superficie piana	2D	'Face'	'F'

All'interno del presente capitolo:

1. [Modellazione di una geometria 1D](#):
 - a) [Aggiungere una punto alla geometria](#);
 - b) [Aggiungere una linea alla geometria](#);
 - c) [Dividere una linea in più segmenti](#);
 - d) [Dividere una linea in un punto specifico](#).
2. [Modellazione di una geometria 2D](#):
 - a) [Aggiungere un rettangolo](#);
 - b) [Aggiungere un cerchio](#);
 - c) [Aggiungere una ellisse](#);

¹ Un metodo (o anche funzione membro), in informatica, è un termine che viene usato principalmente nel contesto della programmazione orientata agli oggetti per indicare un sottoprogramma associato in modo esclusivo ad una classe e che rappresenta (in genere) un'operazione eseguibile sugli oggetti e istanze di quella classe.

- d) [Aggiungere un poligono;](#)
- e) [Dividere il bordo di una superficie in più segmenti;](#)
- f) [Comporre le geometrie;](#)
- g) [Importare la geometria di default.](#)

GEOMETRIA ID

Oggetti geometrici appartenenti alla classe *sap.GeometricModelID* vengono utilizzati ogni volta che l'utente voglia modellare elementi di tipo *beam*. A tal proposito, vengono di seguito discusse e descritte le sintassi delle funzioni che ne consentono la modellazione.

Si descriveranno, inoltre, le operazioni da effettuare dall'interfaccia utente per svolgere le operazioni su citate.

AGGIUNGERE UN PUNTO ALLA GEOMETRIA

L'inserimento di un punto nella geometria del modello strutturale viene effettuato digitando la funzione:

```
sapLab.addpoint(_)
```

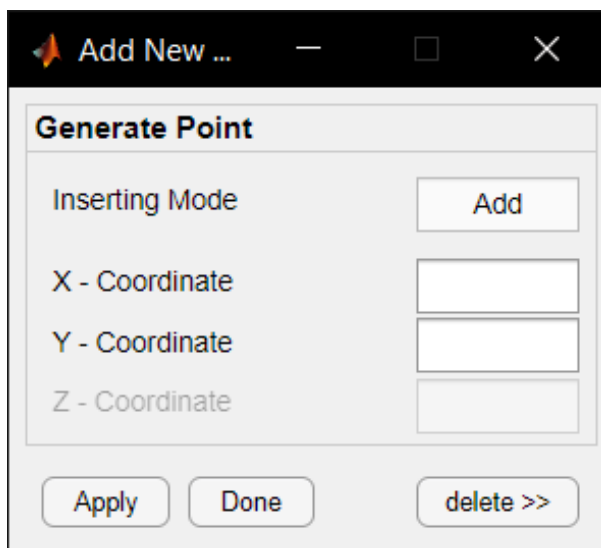
Per maggiori informazioni digitare il comando:

```
sapLab.example('addPointsExample')
```

Sintassi e Descrizione

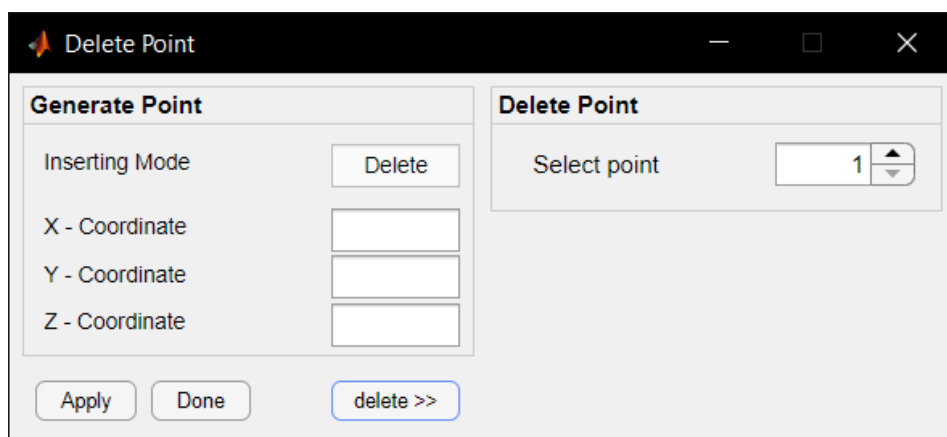
SINTASSI I	<pre>pointID = sapLab.addpoint(structuralmodel, coordinates);</pre> <p>aggiunge un punto alla geometria del modello strutturale structuralmodel. Le coordinate del punto, riferite al sistema di riferimento globale (GCS), vengono definite dalla variabile di <i>input</i> coordinates. La variabile coordinates viene specificata come una matrice Nx2 o Nx3 (con N pari al numero di punti che si vogliono inserire). La funzione fornisce in output l'ID del punto inserito.</p>
------------	--

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Add Point* della *Menu Bar*. Tale operazione apre la *form Add New Point* che richiama la funzione `sapLab.addpoint(_)` dalla quale sarà possibile digitare le coordinate del punto da inserire (**Inserting Mode: Add**).



5 - Inserimento di un punto da interfaccia utente

Dalla *form* in esame, cliccando sul pulsante ***delete*** si attiva la funzione (**Inserting Mode: Delete**). Ciò consentirà all'utente di eliminare il punto avente l'identificativo indicato nello *spinner box*.



6 - Eliminazione del punto

AGGIUNGERE UNA LINEA ALLA GEOMETRIA DEL MODELLO

L'inserimento di un segmento lineare nella geometria del modello strutturale viene effettuato digitando la funzione:

`saplab.addLine(_)`

Sintassi e Descrizione

SINTASSI I	<pre>saplab.addLine(model, 'Coordinates', start, end)</pre> <p>Aggiunge un segmento lineare congiungente i punti di coordinate start ed end. Le coordinate dei punti vanno specificate nel formato [X,Y,Z].</p>
------------	---

```
sapLab.addLine(model, 'Connect', [p1,p2])
```

SINTASSI 2

Aggiunge un segmento lineare congiungente i punti aventi identificativo pari a **p1** e **p2**. Gli identificativi dovranno indicare dei punti già esistenti all'interno della geometria del modello. Se tale condizione non fosse verificata, allora la funzione genererà un errore.

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Add Line* della *Menu Bar*. Tale operazione apre la *form Add New Straight Line* che richiama la funzione [saplab.addline\(\)](#) dalla quale sarà possibile digitare le coordinate del punto da inserire (**Inserting Mode**: Add). Dalla *form* in esame, cliccando sul pulsante **delete** si attiva la funzione (**Inserting Mode**: Delete). Ciò consentirà all'utente di eliminare il punto avente l'identificativo indicato nello *spinner box*.

The screenshot shows a window titled 'Add New Straight Line...'. It has a 'Generate Line' section with 'Inserting Mode' set to 'Coordinates'. Below this are two input fields: 'Starting Point Location (x,y,z)' and 'Ending Point Location (x,y,z)'. At the bottom, there is an 'Inserting Mode' section with radio buttons for 'Coordinates' (selected) and 'Connect'. To the right of these are three buttons: 'Apply', 'Done', and 'Delete >>'.

7 - Add New Straight Line In Modalità Coordinates

The screenshot shows the same 'Add New Straight Line...' window, but with 'Inserting Mode' set to 'Connect'. The 'Starting Point Location' and 'Ending Point Location' fields are replaced by 'Starting Point Identifier' and 'Ending Point Identifier' fields. The 'Connect' radio button is selected in the 'Inserting Mode' section. The 'Apply', 'Done', and 'Delete >>' buttons remain.

8 - Add New Straight Line In Modalità Connect

Per maggiori informazioni digitare il comando:

```
sapLab.example('addLinesExample')
```

DIVIDERE UNA LINEA IN PIÙ SEGMENTI

Per dividere una linea della geometria del modello in più segmenti digitare la funzione:

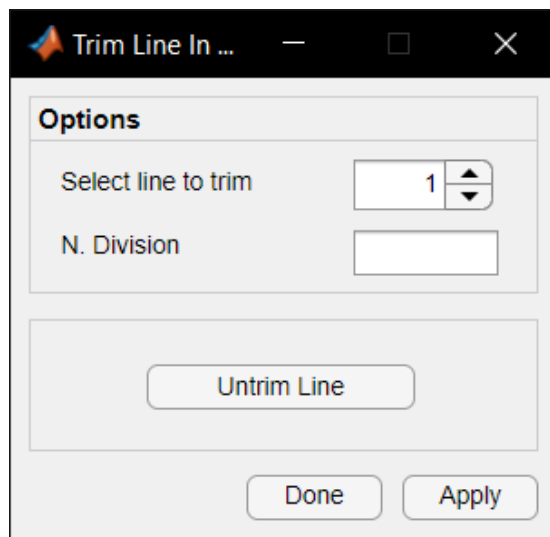
```
sapLab.trimLine(_)
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('trimLinesExample')
```

SINTASSI I	<pre>saplab.trimLine(model, 'Line', value)</pre> <p>Divide una linea della geometria in segmenti di egual lunghezza. L'argomento value è una matrice di dimensione Nx2 (con N pari al numero di linee da dividere) dove la prima colonna contiene l'identificativo della linea da dividere e la seconda contiene il numero di divisioni da effettuare.</p>
------------	---

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Trim Line* della *Menu Bar*. Tale operazione apre la *form Trim Line In Equal Segments* che richiama la funzione [saplab.trimLine\(_\)](#).



9 - Form Trim Line In Equal Segments

Sempre dalla stessa *form*, cliccando sul pulsante **Untrim Line**, è possibile ripristinare la linea al suo stato originale.

DIVIDERE UNA LINEA IN UN PUNTO SPECIFICO

Per dividere una linea della geometria in due segmenti ad una distanza specificata digitare la funzione:

```
saplab.trimLineAtLocation(_)
```

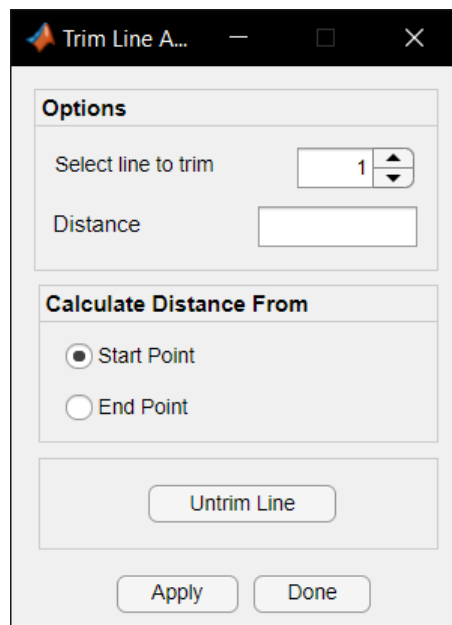
Per maggiori informazioni digitare il comando:

```
saplab.example('trimLinesAtLocationExample')
```

SINTASSI I	<pre>saplab.trimLineAtLocation(model, 'Line', value1, 'Origin', value2)</pre> <p>Divide una linea della geometria in due segmenti ad una distanza specificata dall'utente conteggiata a partire da uno dei due punti di estremità. L'argomento <i>value1</i> contiene l'identificativo della linea da dividere e la distanza alla quale applicare il nuovo punto. L'argomento <i>value2</i> contiene il punto da cui contare la distanza di cui sopra. Quest'ultimo argomento può assumere valore pari a 'start' e 'end'. Qualora non specificato esso assume valore pari a 'start'.</p>
------------	--

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Trim Line At Location* della *Menu Bar*.

Tale operazione apre la *form Trim Line At Specific Location* che richiama la funzione `saplab.trimLineAtLocation(_)`.



10 - Trim Line At Specific Location

GEOMETRIA 2D

Oggetti geometrici appartenenti alla classe *sap.GeometricModel2D* vengono utilizzati ogni volta che l'utente voglia modellare elementi finiti piani. La modellazione delle geometrie viene fatta attraverso la combinazione di primitive geometriche (Rettangolo, Cerchio, Ellisse).

A tal proposito, vengono di seguito discusse e descritte le sintassi delle funzioni che ne consentono la modellazione.

Si descriveranno, inoltre, le operazioni da effettuare dall'interfaccia utente per svolgere le operazioni su citate.

AGGIUNGERE UNA SUPERFICIE RETTANGOLARE

L'inserimento di una superficie rettangolare viene effettuato digitando la funzione:

```
sapLab.addrect(_)
```

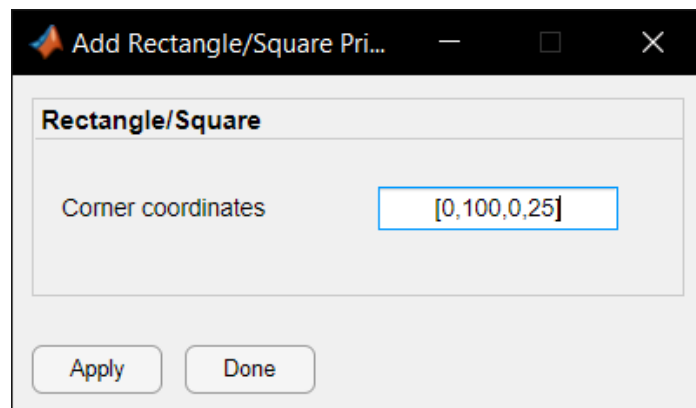
Per maggiori informazioni digitare il comando:

```
sapLab.example('addPrimitivesExample')
```

Sintassi e Descrizione

SINTASSI I	<pre>gtag = sapLab.addrect(structuralmodel, dim)</pre> <p>Genera una superficie rettangolare. Gli argomenti di <i>input</i> della funzione sono il modello strutturale <i>structuralmodel</i> e le dimensioni del rettangolo <i>dim</i>. La funzione ritorna in output il tag della superficie.</p>
------------	---

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Add Rectangle* della *Menu Bar*.



II - Add Rectangle/Square Primitive

AGGIUNGERE UNA SUPERFICIE CIRCOLARE

L'inserimento di una superficie circolare viene effettuato digitando la funzione:

```
sapLab.addcirc(_)
```

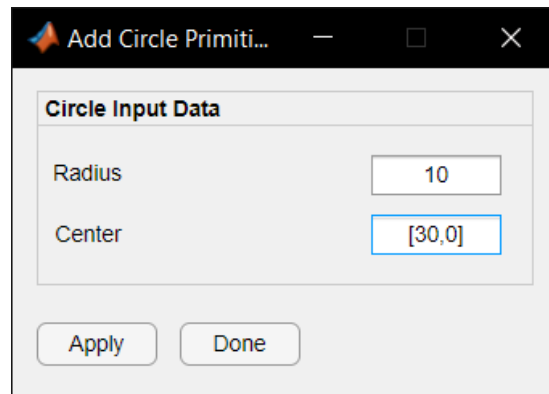
Per maggiori informazioni digitare il comando:

```
sapLab.example('addPrimitivesExample')
```

Sintassi e Descrizione

SINTASSI I	<pre>gtag = saplab.addcirc(structuralmodel, center, radius)</pre> <p>Genera una superficie circolare. Gli argomenti di input della funzione sono il modello strutturale <i>structuralmodel</i>, la posizione del centro del cerchio <i>center</i> e la dimensione del raggio. La funzione ritorna in output il tag della superficie.</p>
------------	--

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Add Circle* della *Menu Bar*.



I2 - Add Circle Primitive

AGGIUNGERE UNA SUPERFICIE ELLITTICA

L'inserimento di una superficie ellittica viene effettuato digitando la funzione:

```
saplab.addellip(_)
```

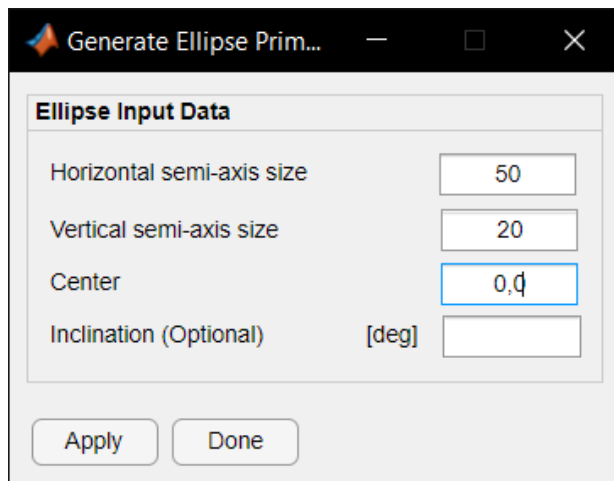
Per maggiori informazioni digitare il comando:

```
saplab.example('addPrimitivesExample')
```

Sintassi e Descrizione

SINTASSI I	<pre>gtag = saplab.addellip(structuralmodel, center, radius, phi)</pre> <p>Genera una superficie ellittica. Gli argomenti di input della funzione sono il modello strutturale <i>structuralmodel</i>, la posizione del centro dell'ellisse <i>center</i>, la dimensione dei semiassi <i>radius</i> e l'inclinazione rispetto all'asse X del GCS <i>phi</i> (Opzionale).</p> <p>La funzione ritorna in <i>output</i> il tag della superficie.</p>
------------	--

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Add Circle* della *Menu Bar*.



I3 - Add Ellipse Primitive

AGGIUNGERE UN POLIGONO GENERICO

L'inserimento di poligono viene effettuato digitando la funzione:

`sapLab.addpoly(_)`

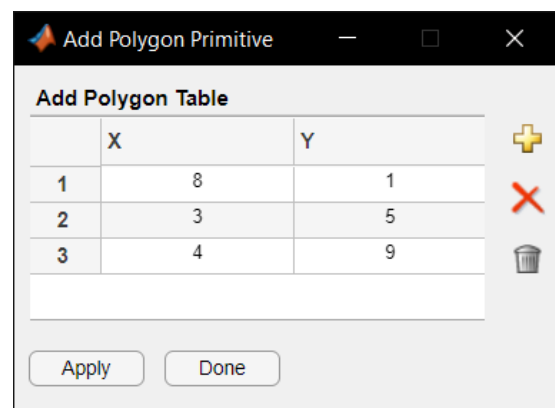
Per maggiori informazioni digitare il comando:

`sapLab.example('addPrimitivesExample')`

Sintassi e Descrizione

SINTASSI I	<p><code>gtag = sapLab.addpoly(structuralmodel, x, y)</code></p> <p>Genera un poligono. Gli argomenti di input della funzione sono il modello strutturale <i>structuralmodel</i> e i vertici che compongono il poligono x e y. I vertici devono costituire un percorso chiuso. Se questa condizione non viene soddisfatta la funzione genera un errore.</p>
------------	---

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Add Polygon* della *Menu Bar*. All'interno della tabella è possibile inserire le coordinate dei vertici del poligono da creare.



I4 - Add Polygon

DIVIDERE IL BORDO DI UNA SUPERFICIE IN PIÙ SEGMENTI

È possibile dividere i bordi di una superficie mediante la funzione:

```
sapLab.trimEdge(_)
```

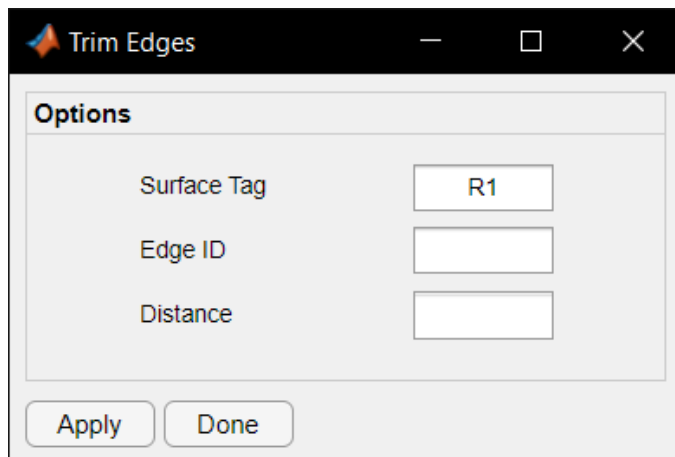
Per maggiori informazioni digitare il comando:

```
sapLab.example('trimEdgeExample')
```

Sintassi e Descrizione

SINTASSI I	<pre>gtag = sapLab.trimEdge(model.Geometry, Tag, 'Edge', value)</pre> <p>Divide il bordo avente identificativo <u>Eid</u> della superficie Tag alla distanza <u>Dist</u> dal vertice iniziale. La funzione fornisce in output il tag della nuova superficie. <u>Eid</u> e <u>Dist</u> vengono specificati all'interno della variabile <u>value</u>. (I parametri relativi all'identificativo e alla distanza vanno inseriti nell'ordine [<u>Eid</u>, <u>Dist</u>]). La funzione, per il momento, consente di effettuare l'operazione in oggetto solo per i bordi lineari della superficie.</p>
------------	--

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Trim Edge* della *Menu Bar*.



15 - Trim Linear Edges

COMPOSIZIONE DELLE PRIMITIVE GEOMETRICHE

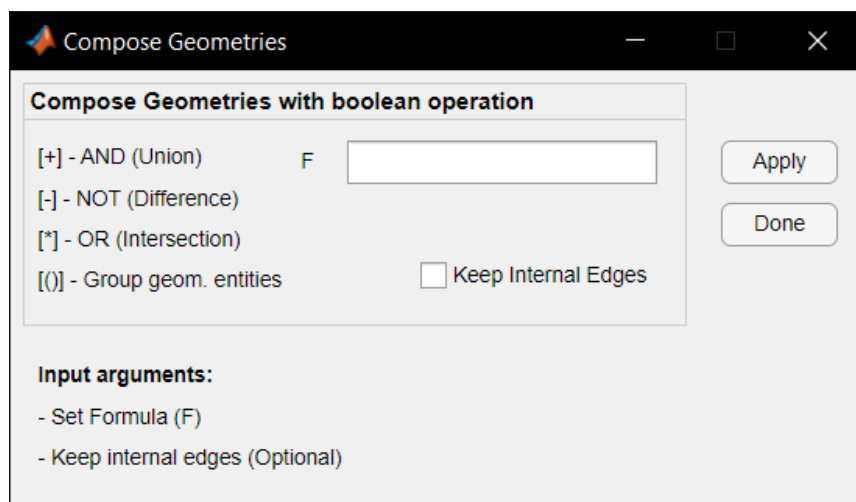
La composizione delle geometrie può essere fatta digitando la funzione:

```
sapLab.boolOper(_)
```


SINTASSI 1	<p><code>gtag = sapLab.booloper(model, formula, Name, Value)</code></p> <p>Tale funzione consente all'utente di comporre le primitive geometriche, fino a qui analizzate, in modo da formare geometrie più o meno complesse. La composizione è consentita grazie alle operazioni booleane che permettono di creare geometrie complesse mediante tre operazioni di trasformazione:</p> <ul style="list-style-type: none"> - AND (Unione - indicata dal simbolo "+") - OR (Intersezione - Indicata dal simbolo "*") - NOT (Sottrazione - Indicata dal simbolo "-") <p>Per comporre geometrie di diversa forma è necessario indicare nell'argomento di input <i>formula</i> il tag delle superfici piane da comporre accompagnato dall'operazione di trasformazione che si vuole eseguire. (Vedi esempio sotto).</p> <p><code>gtag = sapLab.booloper(model, 'R1+C2')</code></p>
SINTASSI 2	<p>Qualora le superfici piane, a seguito di una operazione di combinazione AND, dovessero intersecarsi è possibile decidere se mantenere attivi o meno i bordi in comune tra le due superfici sfruttando gli argomenti di input <i>Name-Value</i> (Vedi Add optional name-value pair argument into input parser scheme - MATLAB addParameter - MathWorks Italia). Si riporta di seguito un breve esempio.</p> <p><code>gtag = sapLab.booloper(model, 'R1+C2', 'KeepInternalEdges', 'on')</code></p>

Per maggiori informazioni digitare il comando:

`sapLab.example('geomCompositionExamples')`



La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Geometry* → *Boolean Operation* della *Menu Bar* (Figura 16).

IMPORTARE LA GEOMETRIA 2D DI DEFAULT

La creazione di una superficie, secondo le modalità viste fino ad ora, non equivale ad applicare la geometria del modello strutturale. Ciò è possibile evidenziarlo attraverso l'esempio riportato di seguito.

```
%creare un nuovo modello strutturale
structuralmodel = saplab.create('attributes','static-planestress')
structuralmodel =
```

StructuralModel with properties:

```
    AnalysisType: "static-planestress"
      Geometry: []
MaterialProperties: []
      BodyLoads: []
BoundaryConditions: []
      Mesh: []
      Results: []
```

```
%genera una superficie rettangolare
t = saplab.addrect(structuralmodel,[0,100,0,20])
t =
```

```
'R1'
```

```
%osserva la geometria del modello strutturale
gm = structuralmodel.Geometry
gm =
```

GeometricModel2D with properties:

```
NumVertices: []
NumEdges: []
NumFaces: []
Vertices: []
```

È possibile osservare come la geometria del modello, nonostante la creazione di una nuova superficie sia vuota. Per assegnare la superficie al modello strutturale bisognerà associare la superficie rettangolare creata alla sua geometria. Per effettuare tale operazione basterà digitare il comando:

```
structuralmodel.Geometry = t;
```

Richiamando, ora, la geometria del modello strutturale si ottiene:

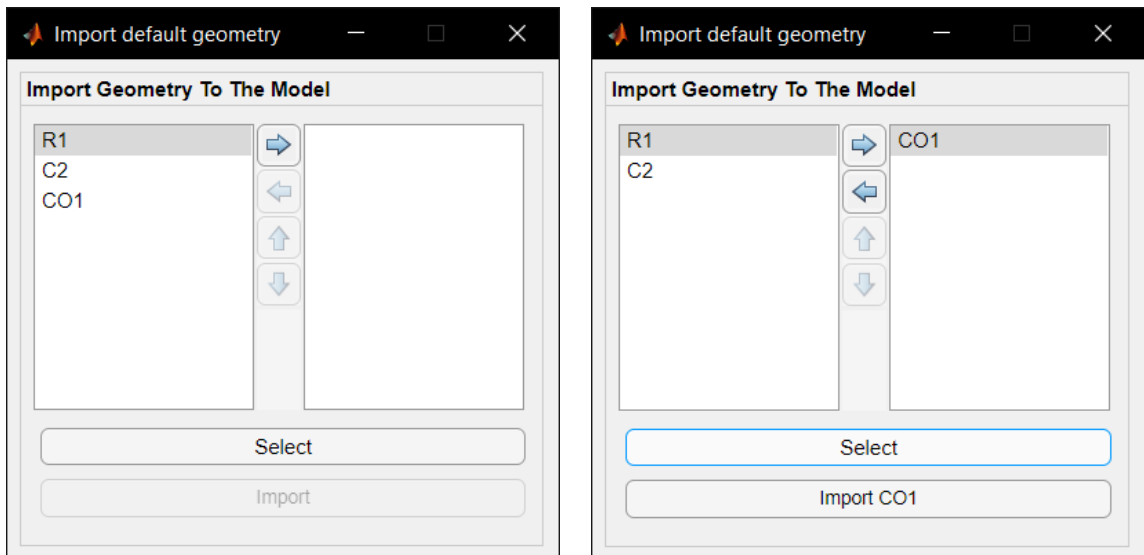
`structuralmodel.Geometry`

`ans =`

`GeometricModel2D` with properties:

`NumVertices: 4`
`NumEdges: 4`
`NumFaces: 1`
`Vertices: [4x2 double]`

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce `Geometry` → `Boolean Operation` della `Menu Bar`.



17 - Import default geometry

Assegnare le proprietà strutturali

Le proprietà degli elementi strutturali che compongono il sistema da analizzare vengono assegnate mediante la definizione di determinate proprietà strutturali. Queste differiscono in base alla tipologia di analisi che si sta svolgendo e, in particolare, consistono nello specificare le proprietà meccaniche del materiale e le proprietà geometriche delle sezioni per quanto concerne elementi di tipo *beam* ('static-planeframe' e 'static-spaceframe' *analysis*) e esclusivamente le proprietà meccaniche del materiale per quanto riguarda elementi finiti piani.

All'interno del presente capitolo:

- [Assegnare le proprietà meccaniche del materiale;](#)
- [Ricerca delle proprietà del materiale;](#)
- [Assegnare le proprietà geometriche delle sezioni;](#)
- [Ricerca delle proprietà geometriche delle sezioni.](#)

ASSEGNARE LE PROPRIETÀ MECCANICHE DEL MATERIALE

Le proprietà meccaniche del materiale possono essere definite digitando la funzione:

```
sapLab.setMaterialProperties(_)
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('setMaterialExample')
```

Sintassi e Descrizione

SINTASSI 1	<pre>sapLab.setMaterialProperties(model, RegionType, RegionID, __)</pre> <p>Assegna le proprietà meccaniche del materiale alla regione geometrica (vedasi regions) individuata (<i>RegionType</i>, <i>RegionID</i>) del modello strutturale <i>model</i>.</p>
SINTASSI 2	<pre>sapLab.assignMaterial(model, __, 'YoungsModulus', Eval)</pre> <p>Assegna il valore del modulo di elasticità alla regione geometrica specificata (Vedi SINTASSI 1).</p>
SINTASSI 3	<pre>sapLab.setMaterialProperties(model, __, 'PoissonsRatio', NuVal)</pre> <p>Assegna il valore del coefficiente di Poisson alla regione geometrica specificata (Vedi SINTASSI 1).</p>

SINTASSI 4	<pre>sapLab.setMaterialProperties(model, __ , 'MassDensity', Gval)</pre> <p>Assegna il valore di densità di massa del materiale alla regione geometrica specificata (Vedi SINTASSI 1).</p> <p>Tale parametro associato al campo gravitazionale (Vedasi Link GravitationalAcceleration) consente di simulare gli effetti dovuti al peso proprio dell'elemento strutturale.</p>
SINTASSI 5	<pre>sapLab.setMaterialProperties(model, __ , 'CTE', Aval)</pre> <p>Assegna il coefficiente di espansione termica del materiale alla regione specificata (Vedi SINTASSI 1). Tale parametro associato a TEMPERATURE (Vedi Link testutaleeeeeee) consente di simulare gli effetti dovuti alla presenza di carichi di natura termica.</p>
SINTASSI 6	<pre>mtl = sapLab.setMaterialProperties(__)</pre> <p>Fornisce in output l'oggetto <code>sap.MaterialPropertiesAssignment</code>.</p>

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Structural* → *Material Properties* della *Menu Bar*.

Select Region		Properties		
Type of region:		Coefficient	Value	Description
<input type="radio"/>	Straight Line	Region ID	1	
<input checked="" type="radio"/>	Face	E	30000	Modulus of Elasticity
<input type="radio"/>	Reinforcement	v	0.2	Poisson's Ratio (Optional)
		γ		Mass Density (Optional)
		α		Coefficient of Thermal Expansion (Optional)

Apply Done

18 - Material Properties

RICERCA DELLE PROPRIETÀ DEI MATERIALI ASSEGNATI

Per ricercare le proprietà dei materiali assegnati al modello strutturale digitare la funzione:

```
sapLab.getMaterialProperties ( _ )
```

Per maggiori informazioni digitare il comando:

```
saplab.example('findMaterialPropertiesExample')
```

Sintassi e Descrizione

SINTASSI 1	<pre>import sapLab.*; mp = getMaterialProperties(model.MaterialProperties,'RegionType',Rid)</pre> <p>Fornisce in output l'oggetto <i>sap.MaterialPropertiesAssignment</i> contenente le proprietà assegnate alla regione geometrica (<u>RegionType</u>, <u>RegionID</u>). Qualora la regione geometrica specificata dall'utente non dovesse avere nessun materiale assegnato allora la funzione ritornerà in <i>output</i>:</p> <pre>mp = [];</pre>
------------	---

ASSEGNARE LE PROPRIETÀ GEOMETRICHE DELLE SEZIONI

Creare un modello di tipo 'static-planeframe' oppure 'static-spaceframe' (Vedasi - [saplab.create\(\)](#)). Per assegnare le proprietà geometriche delle sezioni trasversali degli elementi strutturali (*Line regions*) digitare la funzione:

```
saplab.setSectionProperties(_)
```

Per maggiori informazioni digitare il comando:

```
saplab.example('assingSectionProps')
```

Sintassi e Descrizione

SINTASSI 1	<pre>saplab.setSectionProperties(model,'Line', Rid)</pre> <p>Fornisce in output l'oggetto <i>sap.SectionPropertiesAssignment</i> contenente le proprietà assegnate alla regione geometrica (<u>RegionType</u>, <u>RegionID</u>).</p>
SINTASSI 2	<pre>saplab.setSectionProperties(model,'Area', Aval)</pre> <p>Assegna l'area della sezione della regione geometrica specificata dall'utente (Vedi SINTASSI 1).</p>
SINTASSI 3	<pre>saplab.setSectionProperties(model,'Inertia', [Ixx,Iyy])</pre> <p>Assegna i momenti di inerzia alla sezione della regione geometrica specificata dall'utente (Vedi SINTASSI 1).</p>
SINTASSI 4	<pre>saplab.setSectionProperties(model,'TorsionalConstant', Jxx)</pre> <p>Assegna la costante di torsione alla sezione alla regione geometrica specificata (Vedi SINTASSI 1).</p>

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Structural* → *Section Properties* della *Menu Bar*.

Coefficient	Value	Description
Region ID	1	
A	28.48	Cross-sectional Area
Jxx	1943	Torsional Constant
Iyy	142.4	Inertia along y-y axis
Izz	6.98	Inertia along z-z axis

19 - Section Properties

RICERCA DELLE PROPRIETÀ GEOMETRICHE DELLE SEZIONI ASSEGNATE

Per ricercare le proprietà geometriche delle sezioni assegnate agli elementi strutturali del modello digitare la funzione:

```
sapLab.getSectionProperties(_)
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('findSectionPropertiesExample')
```

Sintassi e Descrizione

SINTASSI I	<pre>import sapLab.*; sp = getSectionProperties(model.SectionProperties,'RegionType', Rid)</pre> <p>Fornisce in output l'oggetto <i>sap.SectionPropertiesPropertiesAssignment</i> contenente le proprietà assegnate alla regione geometrica (<u>RegionType</u>, <u>RegionID</u>). Qualora la regione geometrica specificata dall'utente non dovesse avere nessuna sezione assegnata allora la funzione ritornerà in output</p> <pre>sp = [];</pre>
------------	--

Assegnare i carichi agli elementi strutturali

I carichi da applicare agli elementi strutturali che compongono il sistema possono differire in funzione della tipologia di analisi selezionata dall'utente.

Questi possono essere di tipo:

- **DISTRIBUTED**: carichi distribuiti lungo l'elemento strutturale (*static-planeframe, static-spaceframe*)
- **BODY**: carichi di natura termica e gravitazionale (tutte le tipologie di analisi).

All'interno del presente capitolo:

- [Assegnare i carichi di tipo *distributed*](#);
- [Ricerca dei carichi di tipo *distributed* assegnati](#);
- [Assegnare i carichi di tipo *body*](#);
- [Ricerca dei carichi di tipo *body* assegnati](#).

APPLICARE CARICHI DI TIPO *DISTRIBUTED*

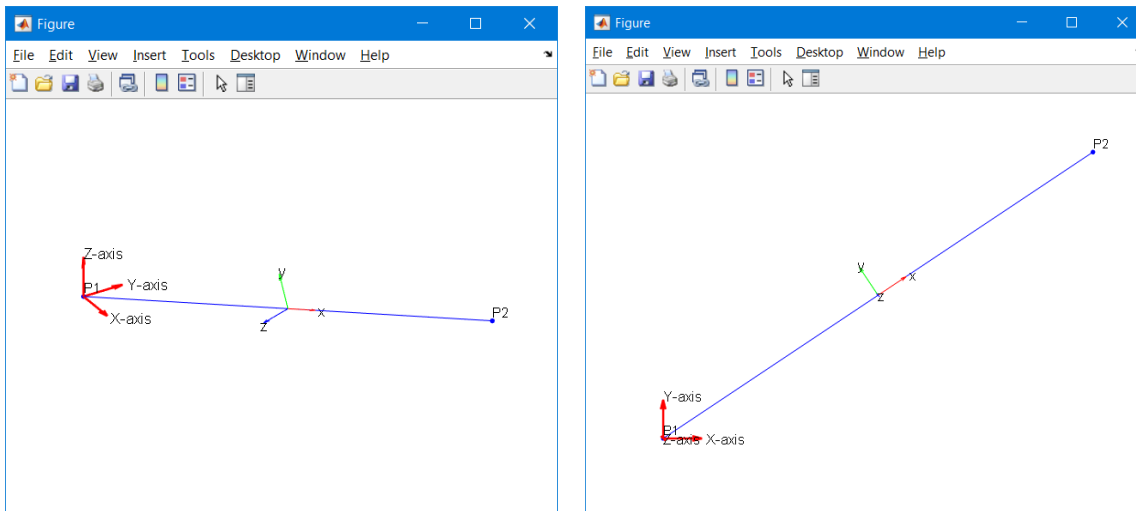
I carichi di tipo *distributed* possono essere assegnati agli elementi strutturali di tipo trave attraverso il comando:

```
saplab.applyDistributed(_);
```

I carichi di tipo *distributed* vengono assegnati alle *regions* di tipo 'Line' (Vedi [Tipologia di regioni geometriche](#)) che rappresentano gli elementi strutturali di tipo trave (beam/frame).

Nell'assegnare questa tipologia di carico è necessario considerare la disposizione dell'elemento trave nel piano 2-D (se è di tipo *static-planeframe*) o nello spazio 3-D (se è di tipo *static-spaceframe*), ossia considerare come questo è orientato nel sistema di coordinate globali (GCS - Global Coordinate System).

L'orientamento viene definito rispetto al GCS dal sistema di riferimento locale della trave (LRS - *Local Reference System*).



20 - Sistema di riferimento locale (LRS)

Tale considerazione è necessaria in quanto l'applicazione dei carichi di tipo *distributed* viene eseguita facendo riferimento al sistema LRS dell'elemento trave. Infatti, ad ogni campo specificato negli argomenti di input della funzione *saplab.applyDistributed(_)* (Vedi tabella seguente e Sintassi e Descrizione) corrisponde una direzione specifica (o asse) del sistema di riferimento LRS.

Tabella 2 - Corrispondenza tra i campi della funzione e gli assi del sistema LRS

Campo	Direzione
'XForce'	Asse X LCS
'YForce'	Asse Y LCS
'ZForce'	Asse Z LCS
'XMoment'	Asse X LCS
'YMoment'	Asse Y LCS
'ZMoment'	Asse Z LCS

Di seguito si riporta la sintassi e la descrizione della funzione in esame.

Sintassi e Descrizione

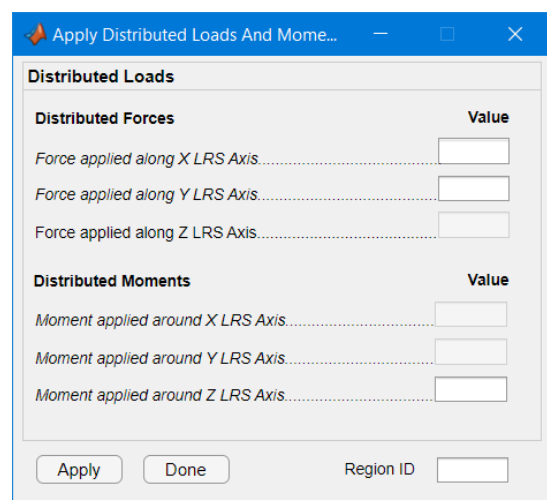
SINTASSI 1	$dl = \text{saplab.applyDistributed}(\text{model}, 'RegionType', \text{RegionID})$ Applica il carico distribuito alla regione geometrica individuata (<i>RegionType</i> , <i>RegionID</i>) del modello strutturale <i>model</i> .
SINTASSI 2	$dl = \text{saplab.applyDistributed}(\text{model}, 'Force', [qx, qy, qz])$ Applica le componenti di carico distribuito [qx,qy,qz] alla regione geometrica specificata (Vedi SINTASSI 1).

SINTASSI 3	$dl = sapLab.applyDistributed(model, 'Moment', [mx, my, mz])$ Applica le componenti di momento [mx,my,mz] alla regione geometrica specificata (Vedi SINTASSI 1).
SINTASSI 4	$dl = sapLab.applyDistributed(model, 'XForce', FxVal)$ Applica un carico distribuito di intensità pari a <u>FxVal</u> lungo l'asse X locale della trave.
SINTASSI 5	$dl = sapLab.applyDistributed(model, 'YForce', FyVal)$ Applica un carico distribuito di intensità pari a <u>FyVal</u> lungo l'asse Y locale della trave.
SINTASSI 6	$dl = sapLab.applyDistributed(model, 'ZForce', FzVal)$ Applica un carico distribuito di intensità pari a <u>FzVal</u> lungo l'asse Z locale della trave.
SINTASSI 7	$dl = sapLab.applyDistributed(model, 'XMoment', MxVal)$ Applica un momento distribuito di intensità pari a <u>MxVal</u> attorno l'asse X locale della trave.
SINTASSI 8	$dl = sapLab.applyDistributed(model, 'YMoment', MyVal)$ Applica un momento distribuito di intensità pari a <u>MyVal</u> attorno l'asse Y locale della trave.
SINTASSI 9	$dl = sapLab.applyDistributed(model, 'ZMoment', MzVal)$ Applica un momento distribuito di intensità pari a <u>MzVal</u> attorno l'asse Z locale della trave.

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Loading* → *Distributed* della *Menu Bar*.

Per maggiori informazioni digitare il comando:

```
sapLab.example('applyDistributedLoadsExample')
```



21 - Distributed Loads

RICERCA DEI CARICHI DI TIPO *DISTRIBUTED* ASSEGNATI

La ricerca dei carichi di tipo *DISTRIBUTED* presenti nel modello strutturale è possibile effettuarla digitando il comando:

```
sapLab.findDistrLoad(_);
```

Sintassi e Descrizione

SINTASSI I	<pre>dl = sapLab.findDistrLoad(model.DistributedLoads,'Line', Rid)</pre> <p>Fornisce in output l'oggetto <code>sap.DistributedLoadsAssignment</code> con la configurazione di carico assegnata alla <i>region</i> selezionata ('Line', RegionID). Qualora la regione geometrica specificata dall'utente non dovesse avere nessuna configurazione di carico assegnata allora la funzione ritornerà in <i>output</i> una variabile vuota.</p> <pre>dl = [];</pre>
------------	---

Per maggiori informazioni digitare il comando:

```
sapLab.example('findDistributedExample')
```

APPLICARE I CARICHI DI TIPO *BODY*

I carichi di tipo *BODY* possono essere assegnati digitando il comando:

```
sapLab.applyBody(_)
```

Tale operazione consente di applicare sulla struttura carichi di natura gravitazionale (specificando l'argomento '*GravitationalAcceleration*') e di natura termica (specificando l'argomento '*Temperature*'). Tuttavia, bisogna precisare che richiamare la funzione `sapLab.applyBody` non è sufficiente per applicare alla struttura le configurazioni di carico su citate, in quanto, queste, risultato dipendenti dalle proprietà strutturali (Proprietà dei materiali e/o proprietà geometriche delle sezioni) dell'elemento a cui vengono applicate.

In particolare, per applicare correttamente un carico gravitazionale ad un elemento di tipo trave (*static-planeframe* e *static-spaceframe*) è necessario specificare i valori delle seguenti proprietà:

- '*MassDensity*' per quanto concerne le proprietà del materiale;
- '*Area*' per quanto riguarda le proprietà della sezione.

Ciò equivale ad applicare all'elemento trave un carico uniformemente distribuito applicato con un verso discorde all'asse Y del sistema LRS pari a:

$$g_y = \frac{g \cdot m \cdot A}{l_e}$$

dove:

1. g è il valore assegnato all'argomento 'GravitationalAcceleration';
2. m è il valore assegnato all'argomento 'MassDensity' (Vedasi - [Proprietà meccaniche dei materiali](#)) nella funzione [saplab.setMaterialProperties\(\)](#);
3. A è il valore assegnato all'argomento 'Area' nella funzione [saplab.setSectionProperties\(\)](#) alla regione selezionata;
4. l_e è la lunghezza dell'elemento finito di tipo trave.

Per applicare, invece, lo stesso carico ad elementi finiti piani di tipo CST basta specificare i valori relativi indicati ai punti 1 e 2 dell'elenco sopra riportato.

Per applicare invece carichi di natura termica ad un elemento trave (*static-planeframe* e *static-spaceframe*) è necessario specificare i valori delle seguenti proprietà dell'elemento:

- 'YoungsModulus' e 'CTE' per quanto concerne le proprietà del materiale;
- 'Area' e 'Inertia' per quanto concerne le proprietà della sezione.

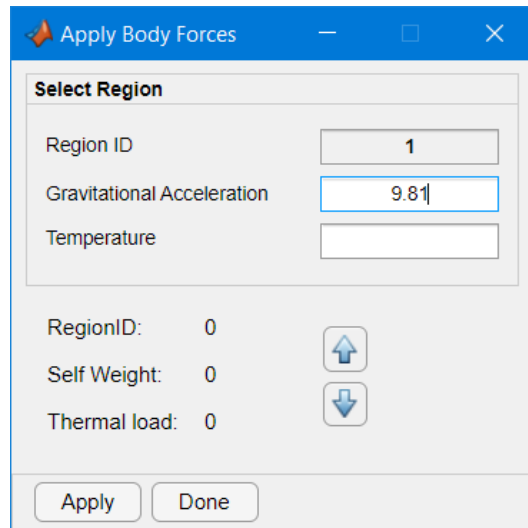
Per applicare, invece, lo stesso carico ad elementi finiti piani di tipo CST basta specificare esclusivamente i valori riguardanti la proprietà del materiale.

A differenza dei carichi di tipo *DISTRIBUTED*, i carichi di tipo *BODY* possono essere assegnati a prescindere dalla tipologia di analisi selezionata.

Sintassi e Descrizione

SINTASSI 1	$bdL = sapLab.applyBody(model, 'RegionType', RegionID)$ Applica il carico distribuito alla regione selezionata (<i>RegionType</i> , <i>RegionID</i>) della geometria del modello strutturale model.
SINTASSI 2	$bdL = sapLab.applyBody(model, 'GravitationalAcceleration', Gval)$ Applica un campo gravitazionale alla regione selezionata (Vedi SINTASSI 1). Specificare tale parametro per simulare gli effetti dovuti al peso proprio degli elementi strutturali.
SINTASSI 3	$bdL = sapLab.applyBody(model, 'Temperature', Gval)$ Applica un carico termico alla regione selezionata (Vedi SINTASSI 1). Specificare tale parametro per simulare gli effetti dovuti alla presenza di variazioni di temperatura.

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Loading* → *Body* della *Menu Bar*.



22 - Body Loads

Per maggiori informazioni digitare il comando:

```
sapLab.example('applyBodyLoadsExample')
```

RICERCA DEI CARICHI DI TIPO BODY ASSEGNATI

La ricerca dei carichi di tipo *body* presenti nel modello strutturale è possibile effettuarla digitando il comando:

```
sapLab.findBodyLoad(_);
```

Sintassi e Descrizione

```
bdL = sapLab.findBodyLoad(model.BodyLoad, 'RegionType', RegionID)
```

SINTASSI | Fornisce in *output* l'oggetto *sap.BodyLoadAssignments* con la configurazione di carico assegnata alla *region* selezionata (*'Line'*, *LineID* o *'Face'*, *FaceID*). Qualora la regione geometrica specificata dall'utente non dovesse avere nessuna configurazione di carico assegnata allora la funzione ritornerà in *output* una variabile vuota.

```
bdL = [];
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('findBodyLoadExample')
```

Condizioni al contorno

Una volta creato un modello strutturale ed assegnatogli una geometria, è possibile assegnare le condizioni al contorno.

Queste possono appartenere a due tipologie distinte che sono:

- condizioni al contorno di tipo Dirichlet;
- condizioni al contorno di tipo Neumann.

Le prime consistono nell'assegnare i vincoli alla struttura o definire degli spostamenti imposti, le seconde, invece, riguardano l'applicazione di carichi al contorno della geometria del modello.

La tabella seguente riassume quanto sopra indicato.

Tabella 3 - Tipologie di condizioni al contorno

Condizione al contorno	Tipo di assegnazione	
Dirichlet	<i>Constraint</i>	Vincoli strutturali
	<i>Displacement/Rotation</i>	Spostamenti imposti
Neumann	<i>Force/Moment</i>	Forze concentrate
	<i>Pressure</i>	Pressione

Le condizioni al contorno vengono applicate alle regioni di frontiera della geometria del modello strutturale. Queste sono individuate da:

- *Point* per analisi di tipo *static-planeframe* e *static-spaceframe*;
- *Vertex* e *Edge* per analisi che coinvolgono elementi finiti piani.

All'interno del presente capitolo:

- [Assegnare le condizioni al contorno](#);
- [Ricerca le condizioni al contorno assegnate](#);

ASSEGNARE LE CONDIZIONI AL CONTORNO

Le condizioni al contorno possono essere assegnate digitando il comando:

```
sapLab.setBoundaryCondition(_);
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('applyBoundaryCondition')
```

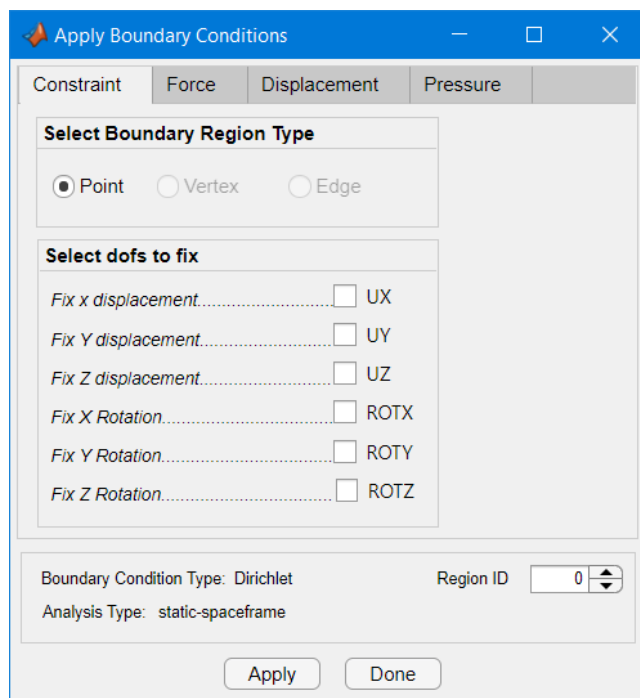
Sintassi e Descrizione

SINTASSI 1	<p><code>bc = saplab.setBoundaryCondition(model, 'RegionType', RegionID)</code></p> <p>Applica la condizione al contorno alla regione di frontiera della geometria selezionata (<u>RegionType</u>, <u>RegionID</u>) del modello strutturale <u>model</u>.</p>
SINTASSI 2	<p><code>bc = saplab.setBoundaryCondition(model, 'Constraint', Cval)</code></p> <p>Applica un vincolo strutturale di tipo <u>Cval</u> alla regione di frontiera della geometria selezionata (<u>RegionType</u>, <u>RegionID</u>) del modello strutturale <u>model</u>. Il vincolo strutturale <u>Cval</u> può assumere i seguenti valori:</p> <ul style="list-style-type: none"> ▪ 'fixed' ▪ 'free' ▪ 'pinned' ▪ 'roller-x' ▪ 'roller-y' ▪ 'roller-z' ▪ 'roller-xy' ▪ 'roller-yz' ▪ 'roller-xz' <p>N.B. La selezione delle tipologie di vincolo sopra elencate dipende dall'analisi che si sta effettuando. Se l'utente seleziona un vincolo non compatibile con la tipologia di analisi del modello allora la funzione genererà in output un errore.</p>
SINTASSI 3	<p><code>bc = saplab.setBoundaryCondition(model, 'Displacement', [ux,uy,uz])</code></p> <p>Applica le componenti di spostamento imposto [ux,uy,uz] alla regione di frontiera della geometria selezionata (<u>RegionType</u>, <u>RegionID</u>) del modello strutturale <u>model</u>.</p> <p>N.B. Il valore di <u>uz</u> non può essere applicato nel caso di modelli nel piano 2-D.</p>
SINTASSI 3	<p><code>bc = saplab.setBoundaryCondition(model, 'Rotation', [rx,ry,rz])</code></p> <p>Applica le componenti di rotazione imposta [rx,ry,rz] alla regione di frontiera della geometria selezionata (<u>RegionType</u>, <u>RegionID</u>) del modello strutturale <u>model</u>.</p> <p>N.B. L'argomento '<u>Rotation</u>' può essere selezionato solo per l'analisi di strutture a telaio nel piano 2-D o nello spazio 3-D. Qualora il modello fosse piano allora <u>rx</u> e <u>ry</u> non sono selezionabili.</p>

SINTASSI 4	<pre>bc = sapLab.setBoundaryCondition(model, 'XDisplacement', ux) bc = sapLab.setBoundaryCondition(model, 'YDisplacement', uy) bc = sapLab.setBoundaryCondition(model, 'ZDisplacement', uz)</pre> <p>Applica uno spostamento imposto alla regione di frontiera della geometria selezionata (<i>RegionType</i>, <i>RegionID</i>) del modello strutturale <i>model</i>.</p> <p>N.B. L'argomento <i>ZDisplacement</i> non può essere applicato nel caso di modelli nel piano 2-D.</p>
SINTASSI 5	<pre>bc = sapLab.setBoundaryCondition(model, 'XRotation', rx) bc = sapLab.setBoundaryCondition(model, 'YRotation', ry) bc = sapLab.setBoundaryCondition(model, 'ZRotation', rz)</pre> <p>Applica una rotazione imposta alla regione di frontiera della geometria del modello strutturale <i>model</i> selezionata (<i>RegionType</i>, <i>RegionID</i>).</p> <p>N.B. Gli argomenti <i>XRotation</i>, <i>YRotation</i> e <i>ZRotation</i> possono essere specificati solo per l'analisi di strutture a telaio nel piano 2-D e/o nello spazio 3-D.</p> <p>N.B. <i>ZRotation</i> può essere selezionato solo per l'analisi di strutture a telaio nello spazio 3-D.</p>
SINTASSI 6	<pre>bc = sapLab.setBoundaryCondition(model, 'Force', [fx, fy, fz])</pre> <p>Applica una forza concentrata avente componenti [fx,fy,fz] alla regione di frontiera della geometria selezionata (<i>RegionType</i>, <i>RegionID</i>) del modello strutturale <i>model</i>.</p> <p>N.B. Il valore di <i>fz</i> non può essere applicato nel caso di modelli nel piano 2-D.</p>
SINTASSI 7	<pre>bc = sapLab.setBoundaryCondition(model, 'XForce', fx) bc = sapLab.setBoundaryCondition(model, 'YForce', fy) bc = sapLab.setBoundaryCondition(model, 'ZForce', fz)</pre> <p>Applica una forza concentrata alla regione di frontiera della geometria selezionata (<i>RegionType</i>, <i>RegionID</i>) del modello strutturale <i>model</i>.</p> <p>N.B. L'argomento <i>ZForce</i> non può essere applicato nel caso di modelli nel piano 2-D.</p>
SINTASSI 8	<pre>bc = sapLab.setBoundaryCondition(model, 'Moment', [mx, my, mz])</pre> <p>Applica le componenti delle coppie di momento concentrate [mx,my,mz] alla regione di frontiera della geometria selezionata (<i>RegionType</i>, <i>RegionID</i>) del modello strutturale <i>model</i>.</p> <p>N.B. L'argomento <i>Moment</i> può essere selezionato solo per l'analisi di strutture a telaio nel piano 2-D o nello spazio 3-D. Qualora il modello fosse piano allora <i>mx</i> ed <i>my</i> non sono selezionabili.</p>

<p>SINTASSI 9</p>	<pre>bc = sapLab.setBoundaryCondition(model, 'XMoment', mx) bc = sapLab.setBoundaryCondition(model, 'YMoment', my) bc = sapLab.setBoundaryCondition(model, 'ZMoment', mz)</pre> <p>Applica una coppia concentrata alla regione di frontiera della geometria del modello strutturale model selezionata (RegionType, RegionID).</p> <p>N.B. Gli argomenti <i>XMoment</i>, <i>YMoment</i> e <i>ZMoment</i> possono essere specificati solo per l'analisi di strutture a telaio nel piano 2-D e/o nello spazio 3-D.</p> <p>N.B. <i>ZMoment</i> può essere selezionato solo per l'analisi di strutture a telaio nello spazio 3-D.</p>
<p>SINTASSI 10</p>	<pre>bc = sapLab.setBoundaryCondition(model, 'Pressure', [pt, pn])</pre> <p>Applica una pressione uniformemente distribuita alla regione di frontiera della geometria del modello strutturale model ('<i>Edge</i>', <i>RegionID</i>). I valori <i>pt</i> e <i>pn</i> individuano rispettivamente le componenti di pressione tangenziale e di pressione normale.</p> <p>N.B. L'argomento <i>Pressure</i> può essere specificato solo per l'analisi riguardanti gli elementi finiti piani.</p>

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Boundaries* → *<Constraint>/<Force>/<Displacement>/<Pressure>* della *Menu Bar*.



23- Apply Constraint Boundary Condition

In particolare, cliccando su *Boundaries* → *Constraint* l'applicazione SAPLAB apre la *form Apply Boundary Conditions* alla sezione *Constraint*. Dalla *form* in esame è possibile quindi individuare la regione geometrica soggetta alla condizione di vincolo e selezionare i gradi di libertà da bloccare o da lasciare liberi. È possibile osservare, inoltre, come la *form* su citata fornisca la tipologia di condizioni al contorno che si stanno applicando (Vedi voce: *Boundary Condition Type*).

Cliccando su *Boundaries* → *Force* l'applicazione SAPLAB apre la *form Apply Boundary Conditions* alla sezione *Force*. Dalla *form* in esame è possibile applicare le condizioni al contorno in termini di forze concentrate. La tipologia di regione geometrica (*Point*, *Vertex*, *Edge*) in questo caso non è selezionabile in quanto viene riconosciuta automaticamente in funzione della tipologia di analisi. In particolare, se l'analisi è del tipo static-planeframe o static-spaceframe allora la tipologia di regione geometrica viene settata automaticamente su *Point*, viceversa viene settata su *Vertex*.

Parameter	Value
Force along X-global axis.....	<input type="text"/>
Force along Y-global axis.....	<input type="text"/>
Force along Z-global axis.....	<input type="text"/>
Moment around X-global axis.....	<input type="text"/>
Moment around Y-global axis.....	<input type="text"/>
Moment around Z-global axis.....	<input type="text"/>

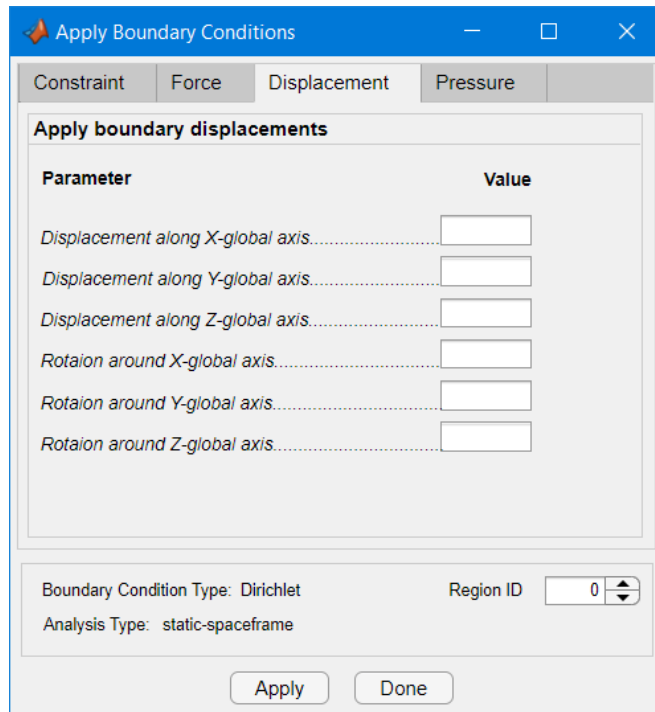
Boundary Condition Type: Neumann Region ID:

Analysis Type: static-spaceframe

Apply Done

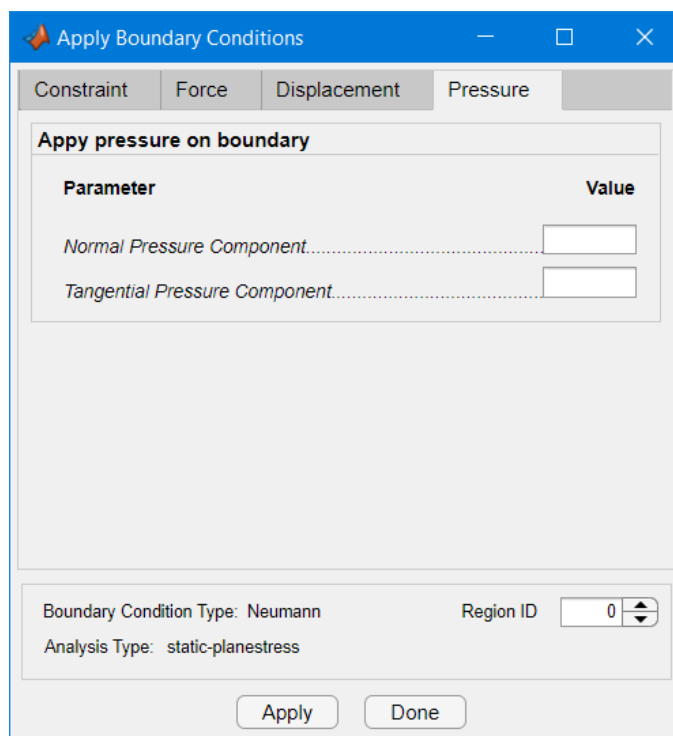
24 - Apply Force Boundary Condition

Cliccando su *Boundaries* → *Displacement* l'applicazione SAPLAB apre la *form Apply Boundary Conditions* alla sezione *Displacement*. Dalla *form* in esame è possibile applicare le condizioni al contorno in termini di spostamenti imposti. Come nel caso precedente, la tipologia di regione geometrica viene settata automaticamente.



25 - Apply Enforced Displacement Boundary Condition

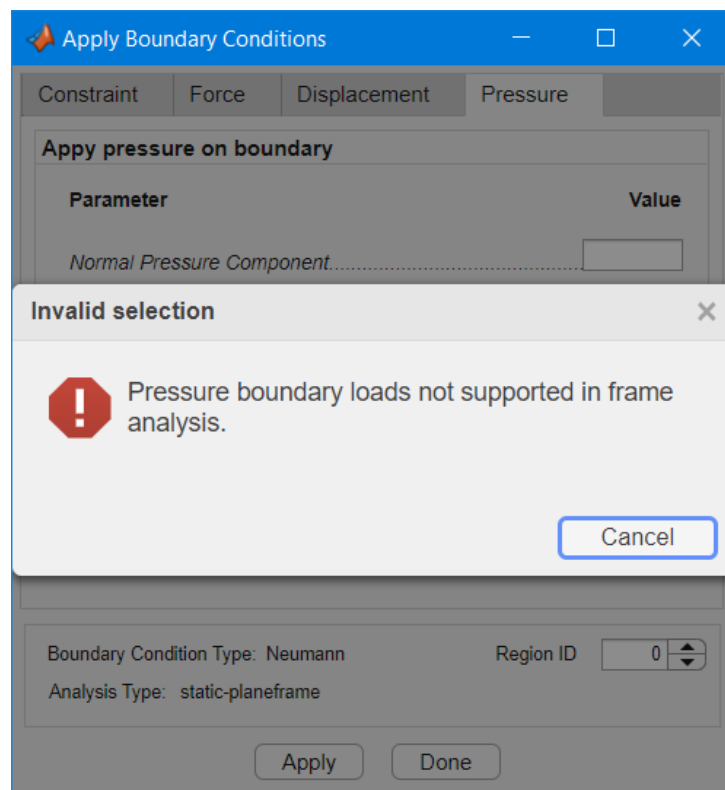
Cliccando su *Boundaries* → *Pressure* l'applicazione SAPLAB apre la *form Apply Boundary Conditions* alla sezione *Pressure*. Dalla *form* in esame è possibile applicare le condizioni al contorno in termini di pressione. Tale opzione è esclusivamente applicabile alle regions di tipo *Edge*.



26 Apply Pressure Boundary Condition

N.B. Un valore positivo di *Normal Pressure Component* corrisponde ad applicare una pressione normale alla *region* ('Edge', *EdgeID*) entrante nella superficie piana che definisce la geometria del modello strutturale. Viceversa, un valore negativo di *Normal Pressure Component* corrisponde ad applicare una pressione normale alla *region* ('Edge', *EdgeID*).

Se l'analisi del modello è del tipo *static-planeframe* o *static-spaceframe* allora condizioni al contorno di tipo *Pressure* non sono applicabili e l'applicazione genererà in *output* un errore.



27 – Condizione al contorno di tipo *pressure* non supportata

RICERCA DELLE CONDIZIONI AL CONTORNO ASSEGNATE

La ricerca delle condizioni al contorno è possibile effettuarla digitando il comando:

```
sapLab.getBoundaryCondition(_);
```

A pagina seguente è riportata la tabella 'Sintassi e Descrizione' dove sono riportate maggiore informazioni in merito.

SINTASSI I	<pre>import sapLab.*; bc = getBoundaryCondition(model.BoundaryConditions,'RegionType',Rid)</pre> <p>Fornisce <code>in</code> <code>output</code> l'oggetto <code>sap.BoundaryConditionPropertiesAssignment</code> contenente le proprietà assegnate alla regione geometrica (<u><i>RegionType</i></u>, <u><i>RegionID</i></u>). Qualora la regione geometrica specificata dall'utente non dovesse avere nessuna condizione al contorno assegnata allora la funzione ritornerà in <code>output</code></p> <pre>bc = [];</pre>
------------	--

Per maggiori dettagli digitare il comando:

```
sapLab.example('applyBodyLoadsExample')
```

Gli end-release

Gli *end-release* consentono all'utente di svincolare i gradi di libertà degli elementi finiti di tipo trave dai nodi terminali nei quali questi confluiscono.

La loro assegnazione al modello strutturale è consentita solo se l'analisi selezionata è del tipo *static-planeframe* o *static-spaceframe* e deve essere effettuata dall'utente prima di generare la mesh del modello strutturale.

La successione delle operazioni nella sequenza deve essere *assegnazione end-release* → *generazione mesh* (e non *generazione mesh* → *assegnazione end-release*) in quanto il problema in esame viene risolto al livello della mesh.

All'interno del presente capitolo:

- [Assegnare gli end-release](#);
- [Ricerca degli end release assegnati](#).

ASSEGNARE GLI END-RELEASE

L'assegnazione degli end-release agli elementi trave del modello strutturale può essere effettuata attraverso la funzione:

```
sapLab.setReleaseCondition(_);
```

Sintassi e Descrizione

```
import sapLab.*;  
bc = setReleaseCondition(model, 'Line', LineID, 'Type', Rctype)
```

Assegna un end-release alla *region* ('Line', LineID). Rctype definisce il grado di libertà da svincolare. Tale variabile può assumere valore pari a:

- *nx*: svincola lo sforzo assiale della trave;
 - *ty*: svincola il taglio agente lungo l'asse Y locale
 - *tz*: svincola il taglio agente lungo l'asse Z locale
 - *mx*: svincola il momento torcente attorno l'asse X locale
 - *my*: svincola il momento flettente attorno l'asse Y locale
 - *mz*: svincola il momento flettente attorno l'asse Z locale
-

SINTASSI 2

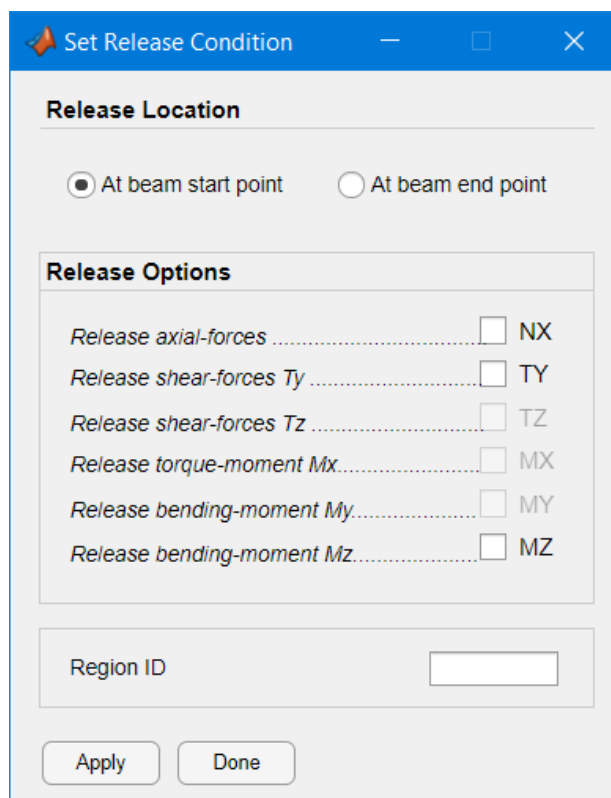
```
import sapLab.*;  
bc = setReleaseCondition(__, 'Location', Loc)
```

Specifica la posizione dell'end-release. Questa può assumere valore pari a *start* o *end*:

Per maggiori informazioni digitare il comando:

```
sapLab.example('setReleaseExample')
```

La stessa operazione è possibile eseguirla dall'interfaccia utente cliccando sulla voce *Boundaries* → *<Release>* della *Menu Bar*.



28 - Set release form

RICERCA DEGLI END-RELEASE

La ricerca degli *end-release* al modello strutturale può essere effettuata attraverso la funzione:

```
sapLab.getReleaseCondition(_);
```


SINTASSI I

```
import sapLab.*;
rc = getReleaseCondition(model.ReleaseConditions,'RegionType', Rid)
```

Fornisce in output l'oggetto *sap.StructuralRC* contenente le proprietà assegnate alla regione geometrica (*RegionType*, *RegionID*) in termini di *end-release*. Qualora la regione geometrica specificata dall'utente non dovesse avere nessun *end-release* assegnato allora la funzione ritornerà in output

```
rc = [];
```

Per maggiori informazioni digitare il comando:

```
sapLab.example('findReleaseExample')
```


Discretizzazione della geometria

Prima di procedere con la risoluzione del modello strutturale con il metodo agli elementi finiti, è necessario sottoporre la sua geometria (Vedi [Modellazione Della Geometria](#)) ad una operazione di discretizzazione in modo tale da generare una maglia (*Mesh*) di nodi ed elementi di connessione.

Questa operazione assegna la tipologia di elemento finito ad ogni elemento di connessione della *Mesh*

Per quanto riguarda la *mesh* delle *regions* di tipo *Line* SAPLAB utilizza un generatore interno che consente di discretizzare entità geometriche di tipo linea in un numero determinato di segmenti. Per quanto concerne, invece, la discretizzazione delle *regions* di tipo *Face* (Superfici piane) si utilizza il generatore di mesh interno di MATLAB® contenuto nel pacchetto [Partial Differential Equation Toolbox](#).

All'interno del presente capitolo:

- [Generazione della mesh](#);
- [Mesh refinement](#);
- [Ricerca dei parametri della mesh](#).

GENERAZIONE DELLA MESH

La generazione della *mesh* può essere effettuata attraverso il comando:

```
sapLab.mesh(_);
```

Sintassi e Descrizione

SINTASSI 1	<pre>msh = sapLab.mesh(model.Geometry)</pre> <p>Genera una <i>mesh</i> applicandola a tutte le <i>regions</i> di tipo <i>Face</i> della geometria del modello strutturale.</p>
SINTASSI 2	<pre>msh = sapLab.mesh(model.Geometry, 'Line', LineID, 'ElementSize', Value)</pre> <p>Genera una <i>mesh</i> alla region ('<i>Line</i>', <i>LineID</i>). L'argomento <i>ElementSize</i> consente di applicare la dimensione dell'elemento finito (segmento con due nodi terminali) specificato dalla variabile <i>Value</i>.</p> <p>Per poter risolvere correttamente il modello agli elementi finiti è necessario generare la <i>mesh</i> per tutte le <i>regions</i> del modello.</p>

SINTASSI 3	<code>msh = sapLab.mesh(model, 'Hmax', hmax)</code>
	<code>msh = sapLab.mesh(model, 'Hmin', hmin)</code>
	Genera una mesh applicandola a tutte le <i>regions</i> di tipo 'Face' della geometria del modello strutturale specificando la dimensione massima o minima che l'elemento finito deve avere all'interno della <i>mesh</i> .

Per maggiori informazioni digitare il comando:

```
sapLab.example('generateMeshExamples')
```

MESH REFINEMENT

L'operazione di *mesh refinement* può essere effettuata solo per versioni di MATLAB® superiori a R2021a (Vedi [MATLAB - Il linguaggio del calcolo tecnico - MATLAB & Simulink \(mathworks.com\)](https://www.mathworks.com/help/matlab/)). Versioni inferiori a quella indicata non supportano tale operazione.

L'operazione di *mesh refinement* può essere effettuata digitando il comando:

```
sapLab.mesh(_);
```

Sintassi e Descrizione

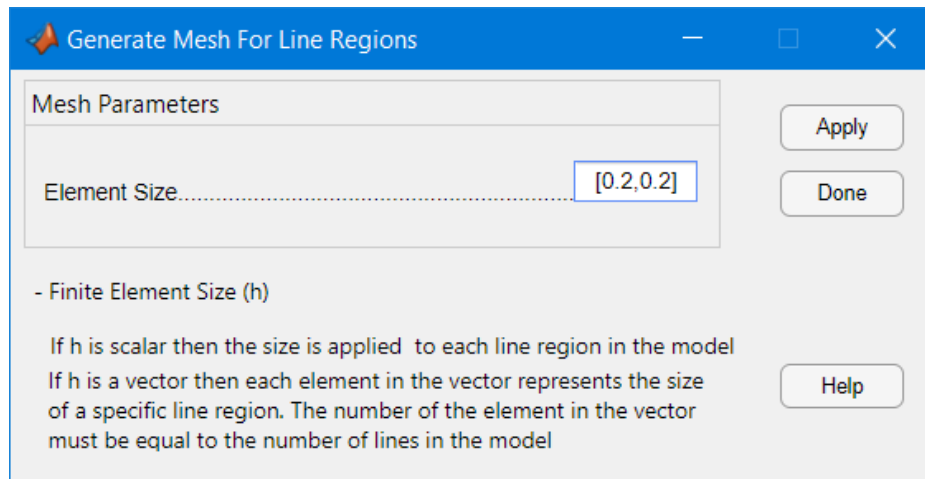
SINTASSI 1	<p><code>msh = sapLab.mesh(model.Geometry, 'Hface', [FaceID, Hvalue])</code></p> <p>Genera una <i>mesh</i> più fitta nella regione geometrica individuata di tipo 'Face' individuata dal parametro <i>FaceID</i>. Tutti gli elementi finiti ricadenti all'interno della regione geometrica in esame presenteranno una <i>mesh</i> tanto più fitta quanto più piccolo sarà il valore <i>Hvalue</i> assegnato.</p>
SINTASSI 2	<p><code>msh = sapLab.mesh(model, 'Hedge', [EdgeID, Hvalue])</code></p> <p>Genera una <i>mesh</i> più fitta nella regione geometrica individuata di tipo 'Edge' individuata dal parametro <i>EdgeID</i>. In corrispondenza del bordo in esame la <i>mesh</i> si presenterà tanto più fitta quanto più piccolo sarà il valore <i>Hvalue</i> assegnato.</p>
SINTASSI 3	<p><code>msh = sapLab.mesh(model, 'Hvertex', [VertexID, Hvalue])</code></p> <p>Genera una <i>mesh</i> più fitta nella regione geometrica individuata di tipo 'Vertex' individuata dal parametro <i>VertexID</i>. In corrispondenza del vertice in esame la <i>mesh</i> si presenterà tanto più fitta quanto più piccolo sarà il valore <i>Hvalue</i> assegnato.</p>

Per maggiori informazioni digitare il comando:

```
sapLab.example('refineMeshExamples')
```

Le operazioni effettuate per la generazione della *mesh* e per il *mesh refinement* possono essere eseguite dall'interfaccia utente cliccando sulla voce *Mesh* → *Mesh Option* della *Menu Bar*.

Se l'analisi del modello strutturale è di tipo *static-planeframe* o *static-spaceframe* il programma aprirà la seguente *form*:



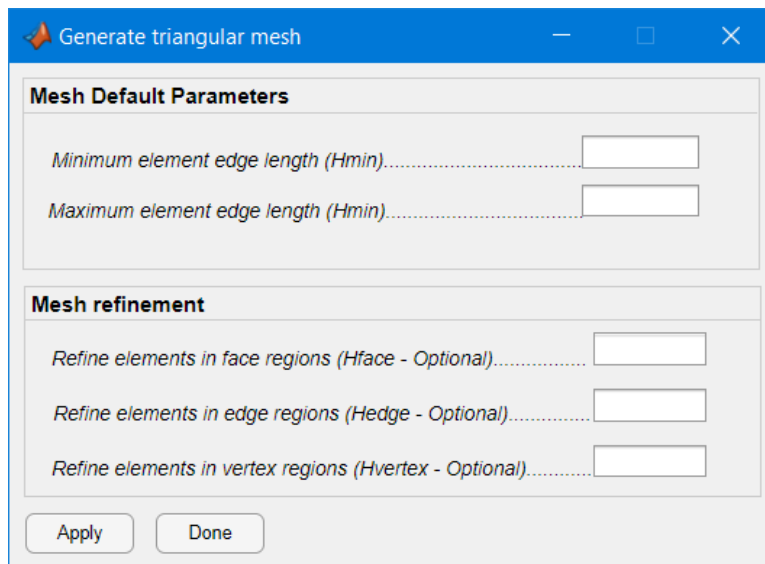
29 - Generate Mesh For Line Regions

Nel campo *ElementSize* vengono inserite le dimensioni dei valori di discretizzazione da assegnare alle regioni geometriche di tipo '*Line*'. Il valore inserito all'interno può essere di tipo scalare (ad Esempio [0.2]) oppure di tipo vettoriale (ad Esempio [0.2,0.3,0.5, ...]).

Nel primo caso [0.2] il valore di discretizzazione viene assegnato ad ogni *region(s)* della geometria del modello strutturale, pertanto, ogni linea presente viene discretizzata con segmenti di lunghezza all'incirca pari a 0.2. Nel secondo caso [0.2,0.3,0.5, ...], invece, (se in presenza di più linee), l'assegnazione del valore di discretizzazione viene fatto in modo distinto per ogni linea presente nella geometria.

L'indice di posizione dei valori di discretizzazione all'interno del vettore viene associato ai valori di identificazione delle linee in questione.

Se l'analisi selezionata dall'utente non coincide con quelle indicate in precedenza, allora il programma aprirà la *form*:



30 - Generate Triangular Mesh

Dalla *form* sopra raffigurata è possibile eseguire una mesh uniforme settando i parametri **Mesh Default Parameters** oppure procedere con il **Mesh refinement** di elementi della mesh.

RICERCA DELLE PROPRIETÀ DELLA MESH

La ricerca delle proprietà della *mesh* può essere eseguita mediante le tre funzioni indicate di seguito.

1. *saplab.findElementsInMesh(_)*: Consente all'utente di trovare la corrispondenza tra gli elementi della *mesh* e le regioni della geometria della struttura. (Ad esempio, trovare quali elementi giacciono della *mesh* sulla *region* 'Face' 1) ;
2. *saplab.findNodesInMesh(_)*: Consente all'utente di trovare la corrispondenza tra i nodi della *mesh* e le regioni della geometria della struttura. (Ad esempio, trovare quali nodi della *mesh* giacciono sulla *region* 'Face' 1) ;
3. *saplab.findFRPInMesh(_)*: Consente all'utente di trovare la corrispondenza tra i nodi/elementi della *mesh* e gli elementi di rinforzo in FRP assegnati alla struttura;

FUNZIONE *saplab.findElementsInMesh()*

La funzione *saplab.findElementsInMesh(_)* consente all'utente di trovare gli elementi finiti della mesh appartenenti alla regione geometrica specificata negli argomenti di input della funzione.

Sintassi e Descrizione

SINTASSI 1	<pre>Import sapLab.* msh = findElementsInMesh(model.Mesh, 'Point', PointID)</pre> <p>Ricerca tutti gli elementi confluenti nel punto avente identificativo <i>PointID</i>.</p>
SINTASSI 2	<pre>Import sapLab.* msh = findElementsInMesh(model.Mesh, 'Line', LineID)</pre> <p>Ricerca tutti gli elementi appartenenti alla linea avente identificativo <i>LineID</i>.</p>
SINTASSI 3	<pre>Import sapLab.* msh = findElementsInMesh(model.Mesh, 'Vertex', VertexID)</pre> <p>Ricerca tutti gli elementi confluenti nel vertice avente identificativo <i>VertexID</i>.</p>
SINTASSI 4	<pre>Import sapLab.* msh = findElementsInMesh(model.Mesh, 'Edge', EdgeID)</pre> <p>Ricerca tutti gli elementi appartenenti al bordo avente identificativo <i>EdgeID</i>.</p>
SINTASSI 5	<pre>Import sapLab.* msh = findElementsInMesh(model.Mesh, 'Face', FaceID)</pre> <p>Ricerca tutti gli elementi appartenenti alla superficie piana avente identificativo <i>FaceID</i>.</p>

Per maggiori informazioni digitare il comando:

```
sapLab.example('findElemsInMeshExample')
```

FUNZIONE saplab.findNodesInMesh()

La funzione *saplab.findNodesInMesh()* consente all'utente di trovare i nodi della mesh appartenenti alla regione geometrica specificata negli argomenti di input della funzione.

Sintassi e Descrizione

SINTASSI 1	<pre>msh = sapLab.findNodesInMesh(model.Mesh, 'Point', PointID)</pre> <p>Ricerca il nodo appartenente al punto avente identificativo <i>PointID</i>.</p>
SINTASSI 2	<pre>msh = sapLab.findNodesInMesh(model.Mesh, 'Line', LineID)</pre> <p>Ricerca tutti i nodi appartenenti alla linea avente identificativo <i>LineID</i>.</p>

SINTASSI 3	<i>msh = sapLab.findNodesInMesh(model.Mesh, 'Vertex', VertexID)</i> Ricerca il nodo appartenente al vertice avente identificativo <i>VertexID</i> .
SINTASSI 4	<i>msh = sapLab.findNodesInMesh(model.Mesh, 'Edge', EdgeID)</i> Ricerca tutti nodi appartenenti al bordo avente identificativo <i>EdgeID</i> .
SINTASSI 5	<i>msh = sapLab.findNodesInMesh(model.Mesh, 'Face', FaceID)</i> Ricerca tutti i nodi appartenenti alla superficie piana avente identificativo <i>FaceID</i> .

Per maggiori informazioni digitare il comando:

```
sapLab.example('findNodesInMeshExample')
```

FUNZIONE sapLab.findFRPInMesh()

La funzione *sapLab.findNodesInMesh()* consente all'utente di trovare i nodi e/o elementi della mesh appartenenti ai fibrorinforzi FRP assegnati alla struttura.

Sintassi e Descrizione

SINTASSI 1	<i>msh = sapLab.findNodesInMesh(model.Mesh, 'Parameter', 'Node')</i> Effettua la ricerca per nodi nell'oggetto mesh <i>model.Mesh</i> .
SINTASSI 2	<i>msh = sapLab.findNodesInMesh(model.Mesh, 'Parameter', 'Elements')</i> Effettua la ricerca per elementi nell'oggetto mesh <i>model.Mesh</i> .
SINTASSI 3	<i>msh = sapLab.findNodesInMesh(__, 'Reinforcement', ReiID)</i> Ricerca i nodi/elementi appartenenti al rinforzo avente identificativo <i>ReiID</i> .
SINTASSI 4	<i>msh = sapLab.findNodesInMesh(__, 'Spring', SprID)</i> Ricerca i nodi/elementi appartenenti agli elementi molla che connettono il rinforzo al bordo della trave.

Per maggiori informazioni digitare il comando:

```
sapLab.example('findFRPInMeshExample')
```


Analisi FEM con elementi FRP

L'analisi agli elementi finiti che consente di simulare travi in cemento armato dotate di fibro-rinforzo FRP nel piano 2-D viene generata digitando il comando:

```
model = saplab.create('attributes','static-coupled-planestress')
```

(Vedi: [Avvio dell'applicazione - Creare un modello SAPLAB](#))

Tale funzione crea un modello strutturale `model` che consente all'utente di accoppiare ('*coupled*') elementi finiti triangolari di tipo *PLANE-STRESS* (trave in c.a.) con elementi finiti monodimensionali di tipo *TRUSS* (rinforzo in FRP).

L'accoppiamento degli elementi finiti sopra citati viene garantito da una serie di molle orizzontali e verticali (interfaccia di collegamento) che connettono i gradi di libertà associati ai nodi della mesh del bordo della trave in c.a. con quelli degli elementi *truss* del rinforzo in FRP.

All'interno del presente capitolo vengono descritte le funzioni necessarie per:

- [creare un rinforzo strutturale](#);
- [assegnare le proprietà meccaniche del fibro-rinforzo](#);
- [creare una interfaccia di collegamento](#);
- [modellare il comportamento meccanico dell'interfaccia di collegamento](#).

CREARE UN RINFORZO STRUTTURALE

Per creare un rinforzo strutturale digitare il comando:

```
saplab.structuralREI(_);
```

Sintassi e Descrizione

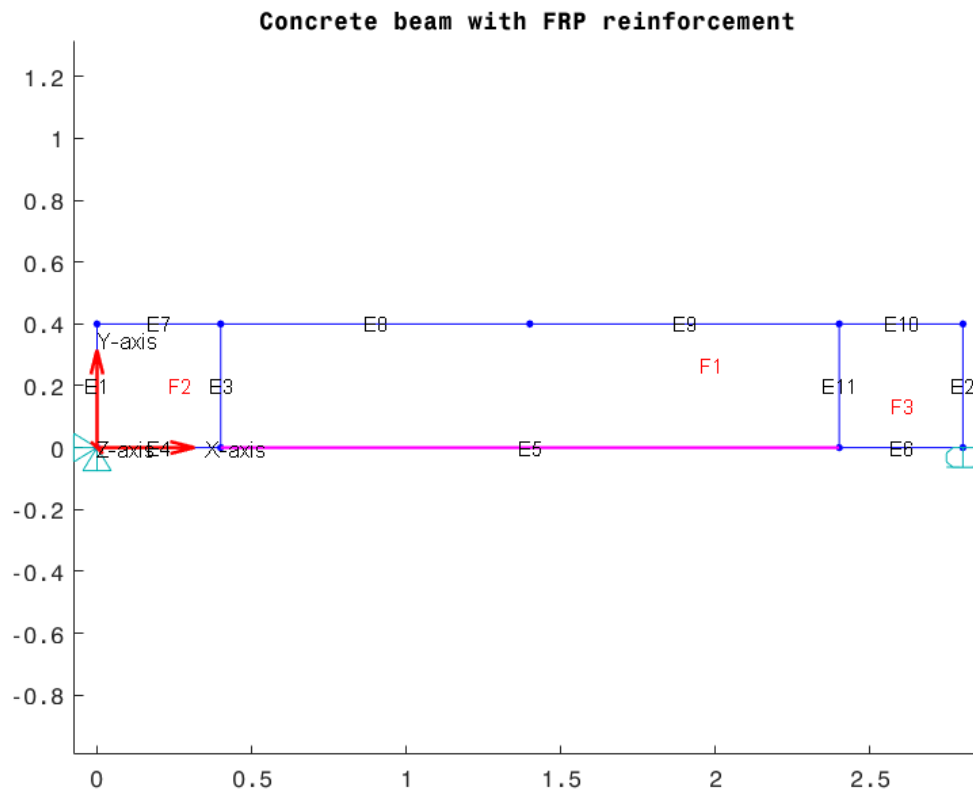
SINTASSI 1	<pre><code>rei = saplab.structuralREI(model,'Edge',EdgeID)</code></pre> <p>Assegna il rinforzo strutturale al bordo <i>EdgeID</i> della geometria del modello strutturale <i>model</i></p>
SINTASSI 2	<pre><code>rei = saplab.structuralREI(__, 'thickness', tval)</code></pre> <p>Assegna il valore dello spessore del rinforzo strutturale.</p>

Per maggiori informazioni digitare il comando:

```
saplab.example('createFRPEXample')
```

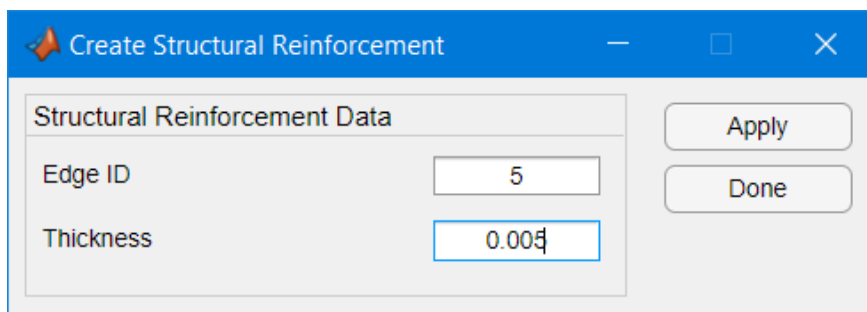
Nel creare un rinforzo strutturale è necessario specificare il bordo della trave a cui è collegato (ad esempio Edge n.1) e lo spessore di cui è dotato ($t = 5 \text{ mm}$).

Plottando a schermo la geometria della struttura analizzata nell'esempio su citato, è possibile osservare il rinforzo strutturale (color magenta) applicato al bordo n.5.



31 - Concrete beam with FRP reinforcement

Per creare un rinforzo strutturale dall'interfaccia utente cliccare sulla voce *Structural* → *Reinforcement Option* della *Menu Bar*.



32 - Create structural reinforcement

ASSEGNARE LE PROPRIETÀ DEL MATERIALE DEL FIBRORINFORZO

Per assegnare le proprietà meccaniche del materiale che costituisce il rinforzo strutturale digitare il comando:

`saplab.setMaterialProperties(_)`

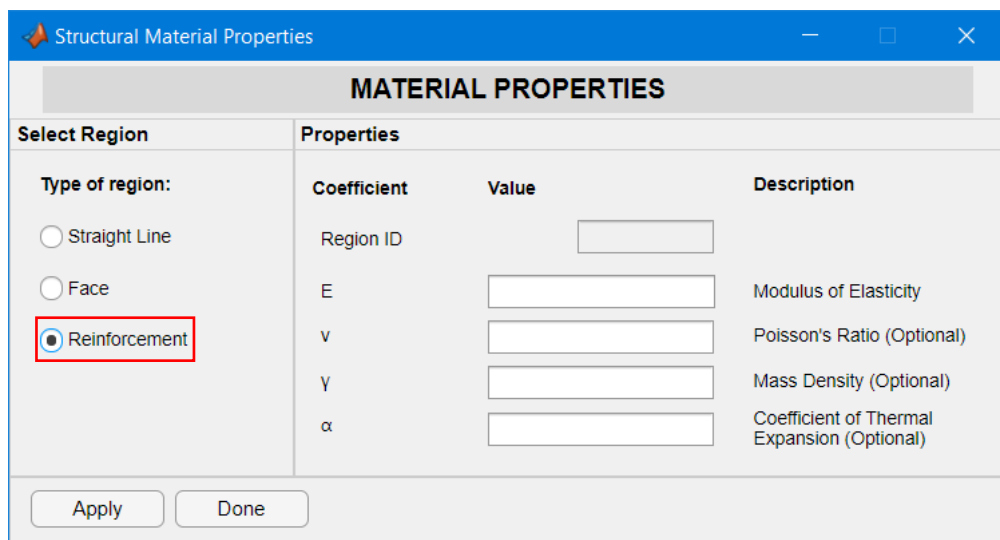
(Vedi: [Assegnare le proprietà strutturali – Assegnare le proprietà meccaniche del materiale](#))

In questo caso è necessario specificare nella funzione su citata la regione a cui assegnare le proprietà meccaniche dei materiali come specificato di seguito.

`mtl = saplab.setMaterialProperties(model, 'Reinforcement', ReiID, Property, Value)`

Per assegnare le proprietà del materiale al rinforzo strutturale dall'interfaccia utente cliccare sulla voce *Structural* → *Material Properties* della *Menu Bar*.

Selezionare dalla form ***Structural Material Properties*** il *radio-button* *Reinforcement*.



MATERIAL PROPERTIES			
Select Region	Properties		
Type of region:	Coefficient	Value	Description
<input type="radio"/> Straight Line	Region ID	<input type="text"/>	
<input type="radio"/> Face	E	<input type="text"/>	Modulus of Elasticity
<input checked="" type="radio"/> Reinforcement	v	<input type="text"/>	Poisson's Ratio (Optional)
	Y	<input type="text"/>	Mass Density (Optional)
	α	<input type="text"/>	Coefficient of Thermal Expansion (Optional)

Apply Done

33 - *Structural Material Properties (Reinforcement)*

CREARE UNA INTERFACCIA DI COLLEGAMENTO

Per creare una interfaccia di collegamento tra il bordo della trave e il rinforzo strutturale digitare il comando:

`saplab.structuralFEI(_);`

Per maggiori informazioni digitare il comando:

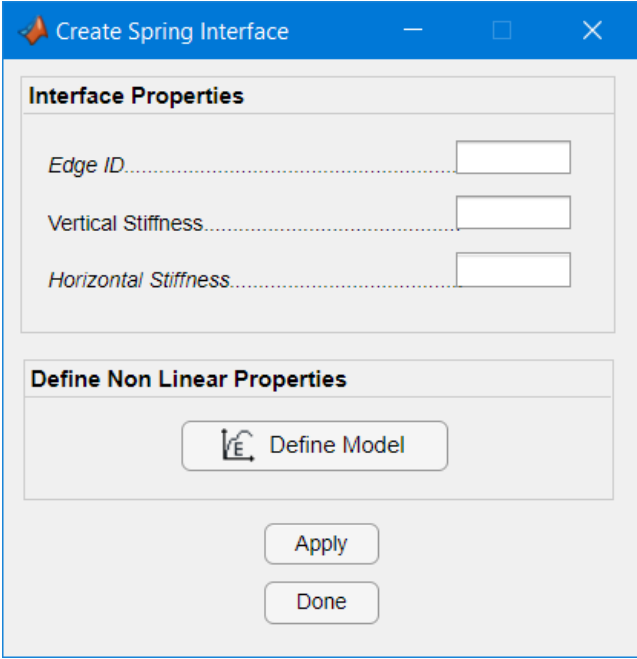
`saplab.example('createFEIExample')`

Sintassi e Descrizione

SINTASSI 1	<pre>fei = saplab.structuralFEI(model, 'Edge', EdgeID)</pre> <p>Assegna l'interfaccia di collegamento al bordo <i>EdgeID</i> della geometria del modello strutturale <i>model</i>. Il bordo selezionato deve possedere un rinforzo strutturale.</p>
SINTASSI 2	<pre>fei = saplab.structuralFEI(__, 'HorizontalStiffness', Kh)</pre> <p>Assegna il valore della rigidezza alle molle orizzontali che costituiscono l'interfaccia.</p>
SINTASSI 3	<pre>fei = saplab.structuralFEI(__, 'VerticalStiffness', Kv)</pre> <p>Assegna il valore della rigidezza alle molle verticali che costituiscono l'interfaccia.</p>

Per creare una interfaccia di collegamento tra il rinforzo strutturale e il bordo della trave dall'interfaccia utente cliccare sulla voce *Structural* → *Interface* della *Menu Bar*. Dalla *form Create Spring Interface* è possibile assegnare le proprietà su citate.

N.B. Bisogna assegnare entrambi i valori di rigidezza *Kh* e *Kv* per non generare un sistema labile.



34 - Create Spring Interface

MODELLARE LE PROPRIETÀ MECCANICHE DELL'INTERFACCIA DI COLLEGAMENTO

SAPLAB consente all'utente di definire due tipologie di comportamento meccanico dell'interfaccia di collegamento elencate di seguito.

1. La prima tipologia consiste nel considerare il comportamento elastico lineare. Per definire tale comportamento basta impostare un valore scalare nei campi 'HorizontalStiffness' e 'VerticalStiffness' come mostrato di seguito.

```
feiH = saplab.structuralFEI(__, 'HorizontalStiffness', 1000)
feiV = saplab.structuralFEI(__, 'VerticalStiffness', 0.1)
```

2. La seconda, invece, consiste nell'assegnare alla suddetta interfaccia un legame costitutivo. Questo può essere generato digitando il comando:

```
saplab.defineMatModel(_)
```

Sintassi e Descrizione

Sintassi e Descrizione	
SINTASSI I	<pre>mtl = saplab.defineMatModel(model,'Force', f, 'u', u)</pre> <p>Genera un oggetto <code>sap.NonLinearMaterialProperties</code> contenente un legame costitutivo di tipo <i>multipoint</i> forza-spostamento.</p>

Per maggiori informazioni vedere l'esempio riportato di seguito.

Si consideri una molla la cui relazione costitutiva $F = Ku$ viene sostituita dalla legge non lineare:

$$F = 10 \cdot u - 2 \cdot u^2$$

Per generare un oggetto `sap.NonLinearMaterialProperties` e quindi il relativo legame costitutivo, digitare i comandi indicati di seguito.

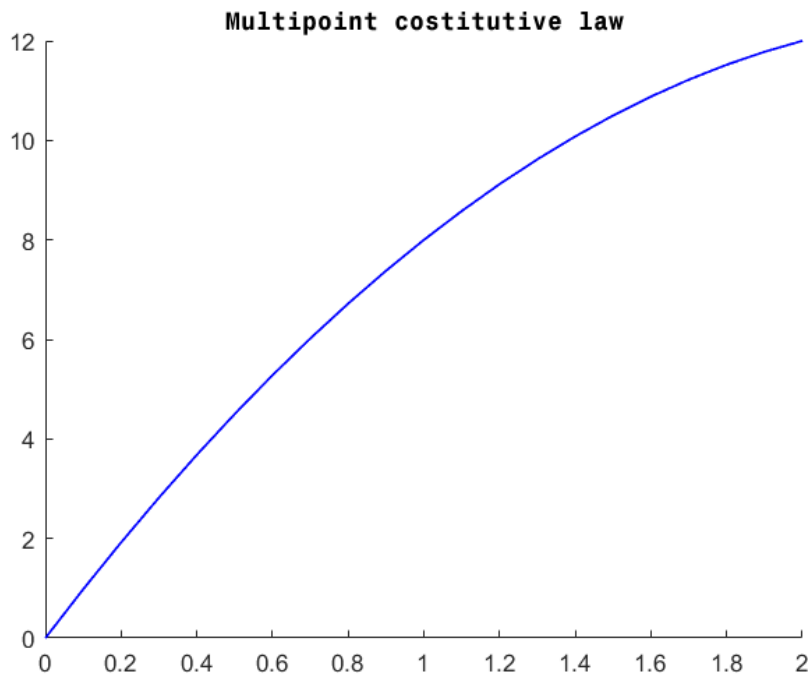
```
u = 0:0.01:2;
f = 10*x-2*x.^2
mtl = saplab.defineMatModel('Force', f, 'u', u);
```

```
mtl =
  NonLinearMaterialProperties with properties:
```

```
    force: [21x1 double]
 displacement: [21x1 double]
```

Per plottare la curva creata digitare:

```
mtl.plot();
```



35 - Multipoint constitutive law

Per assegnare, a questo punto, il comportamento non lineare all'interfaccia di collegamento richiamare la funzione `saplab.structuralFEI(_)` descritta in precedenza e digitare il comando indicato di seguito.

```
mtl = saplab.defineMatModel('Force', f, 'u', u);
feiH = saplab.structuralFEI(__, 'HorizontalStiffness', mtl);
```

È interessante notare come, assegnando il comportamento non lineare all'interfaccia utente, la funzione `saplab.structuralFEI` imposti il valore di rigidezza `'HorizontalStiffness'` pari a quello elastico. Infatti, mandando in output la variabile `feiH` si ottiene un valore di rigidezza della molla orizzontale pari a 9.98.

`feiH =`

StructuralFEInterfaceAssignment with properties:

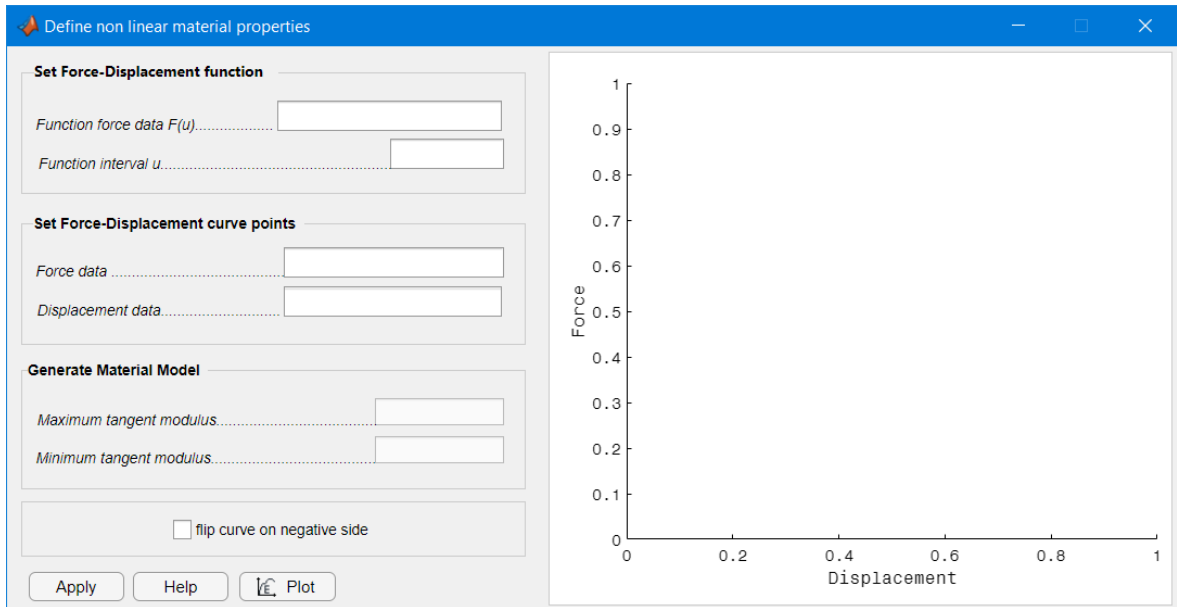
```
RegionType: 'Edge'
RegionID: 5
HorizontalStiffness: 9.9800
VerticalStiffness: []
```

36 Matlab output

Non è difficile dimostrare che la tangente alla curva $F = 10 \cdot u - 2 \cdot u^2$ per $u = 0$ si ottiene:

$$\left(\frac{\partial f}{\partial u}\right)_{u=0} = K_0 = 10 \cong 9.98$$

Per assegnare un comportamento non lineare al materiale dell'interfaccia di collegamento dall'interfaccia utente di SAPLAB cliccare sul pulsante 'Define Model' (Vedi figura 34).

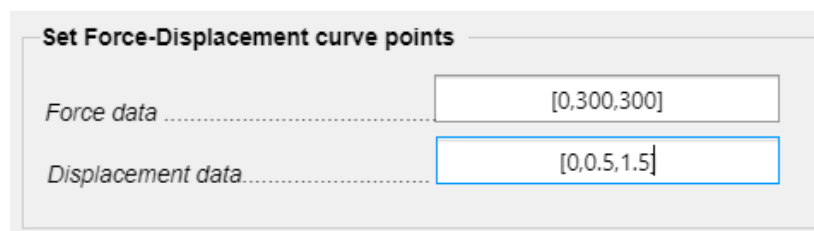


37 - Define non linear material properties


Il legame costitutivo può essere generato dall'utente seguendo due modalità differenti. La prima consiste nell'inserire la funzione non lineare e il relativo intervallo di esistenza, ovvero gli spostamenti (figura 38); la seconda, invece, consiste nell'inserire manualmente i punti della funzione non lineare (figura 39).

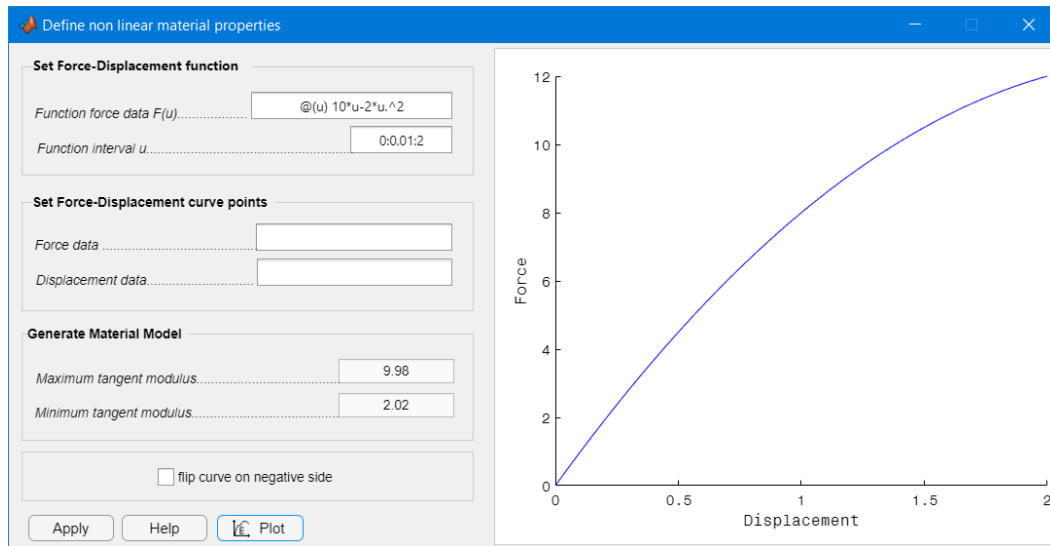


38 - Modalità di inserimento 1

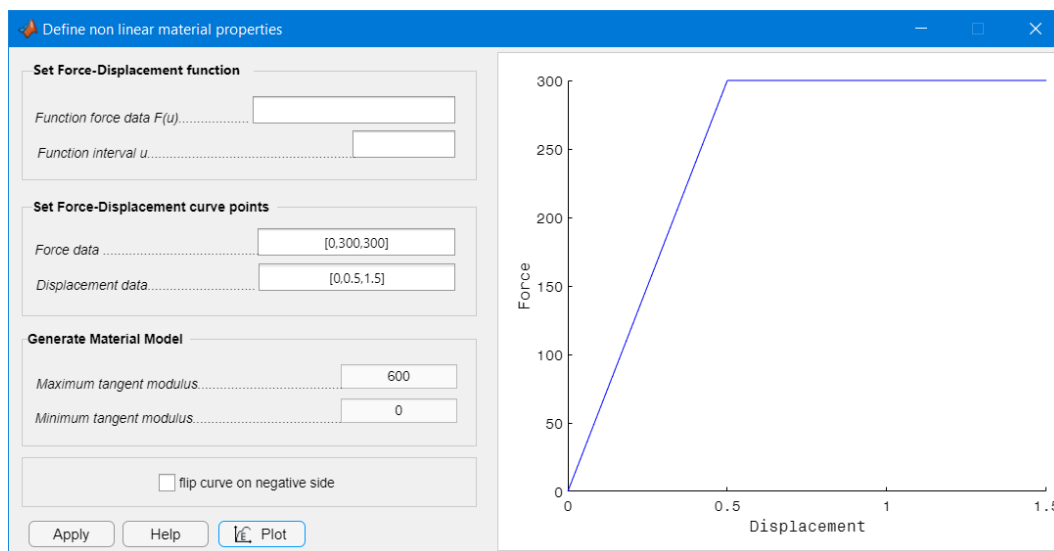


39 - Modalità di inserimento 2

Cliccando sul pulsante  **Plot** è possibile visualizzare la curva corrispondente ai dati inseriti (modalità 1, figura 40; modalità 2, figura 41).

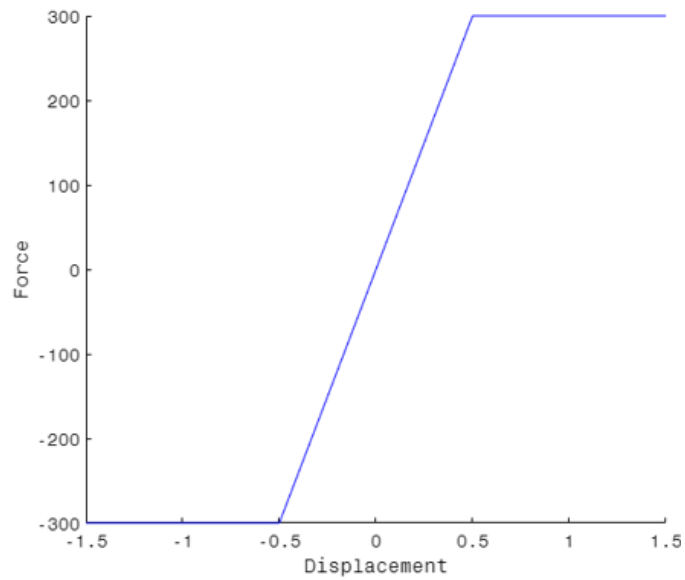


40 - Curva n.1



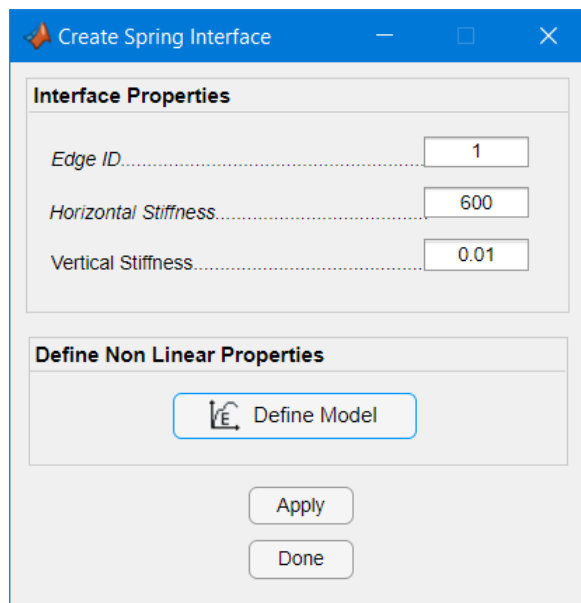
41 - Curva n.2

Qualora si volesse ribaltare il grafico (ovvero i punti della curva) rispetto all'origine degli assi, e così ottenere i medesimi valori della curva anche nella parte negativa degli assi *displacement* e *force*, spuntare il *checkbox* 'flip curve on negative side'.



42 - flip curve

Cliccando, infine, su *Apply* si assegna il legame costitutivo all'interfaccia di collegamento selezionata. È possibile notare che, una volta cliccato il pulsante in esame, nella *form Create Spring Interface* comparirà il modulo tangente relativo alla fase elastica.



43 - Create non linear spring interface

Visualizzazione della geometria

SAPLAB consente all'utente di visualizzare la geometria del modello strutturale grazie ad un insieme di funzioni appositamente create per convertire i dati numerici che l'utente inserisce nella fase di *input* (o *pre-processing*) in dati grafici.

Le funzioni sopra citate possono essere richiamate dall'utente digitando il comando:

```
saplab.gplot(_)
```

All'interno del presente capitolo:

- [Esempio n.1 : plottare la geometria di un modello 'static-planeframe' o 'static-spaceframe';](#)
- [Esempio n.2 : plottare le *labels* degli elementi presenti nella geometria del modello dell'esempio n.1;](#)
- [Esempio n.3 : plottare la geometria di un modello 'static-planestress' o 'static-planestrain';](#)
- [Esempio n.4 : plottare le *labels* degli elementi presenti nella geometria del modello dell'esempio n.3;](#)
- [Esempio n.5 : plottare una geometria CSG;](#)
- [Esempio n.6 : plottare la geometria di un modello 'coupled-static-planestress';](#)
- [Esempio n.7 : plottare le *labels* degli elementi presenti nella geometria del modello dell'esempio n.6;](#)

ESEMPIO I – PLOT GEOMETRIA (STATIC-PLANEFRAME/SPACEFRAME)

Creare un modello 'static-planeframe' :

```
import saplab.*;
```

```
model = create();
```

Aggiungere tre linee consecutive avente lunghezza pari a 6 m come indicato di seguito.

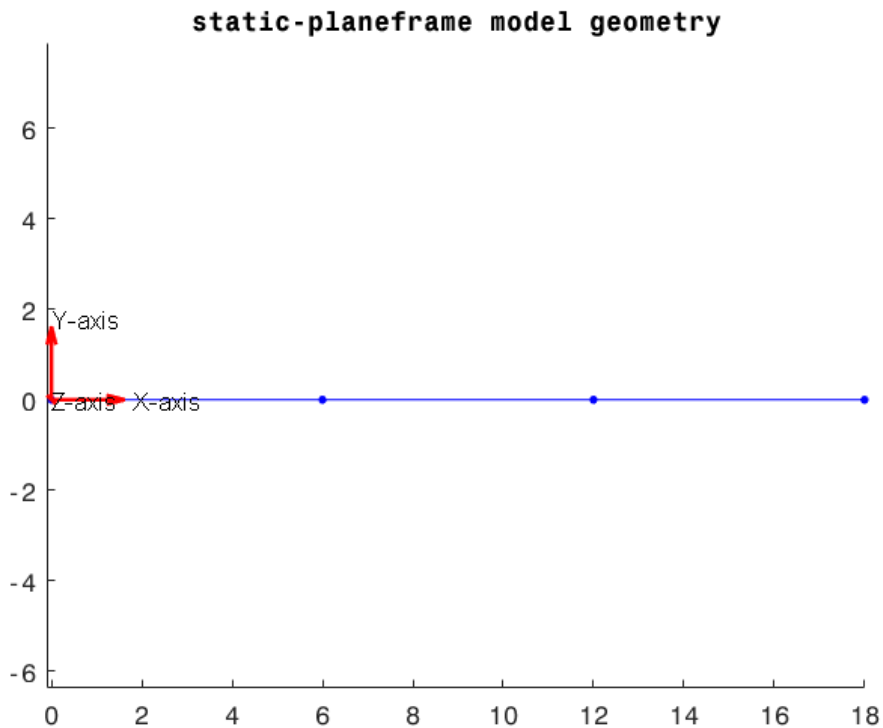
```
addline(model, 'Coordinates', [0,0],[6,0]);
```

```
addline(model, 'Coordinates', [6,0],[12,0]);
```

```
addline(model, 'Coordinates', [12,0],[6,0]);
```

Per plottare a schermo la geometria del modello appena creato digitare il comando:

```
gplot(model.Geometry);
```



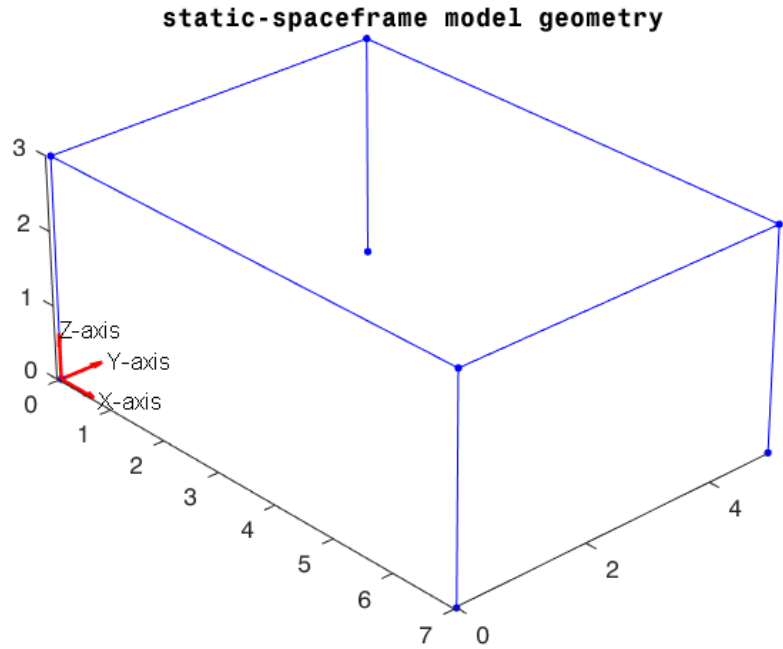
44 - static-planeframe geometry

La stessa operazione può essere effettuata per plottare la geometria relativa ad un modello *static-spaceframe*.

```
import saplab.*;  
model = create('attributes','static-spaceframe');  
addline(model, 'Coordinates', [0,0,0],[0,0,3]);  
addline(model, 'Coordinates', [0,0,3],[7,0,3]);  
addline(model, 'Coordinates', [7,0,3],[7,0,0]);  
addline(model, 'Coordinates', [0,5,0],[0,5,3]);  
addline(model, 'Coordinates', [0,5,3],[7,5,3]);  
addline(model, 'Coordinates', [7,5,3],[7,5,0]);  
addline(model, 'Coordinates', [0,0,3],[0,5,3]);  
addline(model, 'Coordinates', [7,0,3],[7,5,3]);
```

Per plottare a schermo la geometria del modello appena creato digitare il comando:

```
gplot(model.Geometry);
```



45 - static-spaceframe geometry

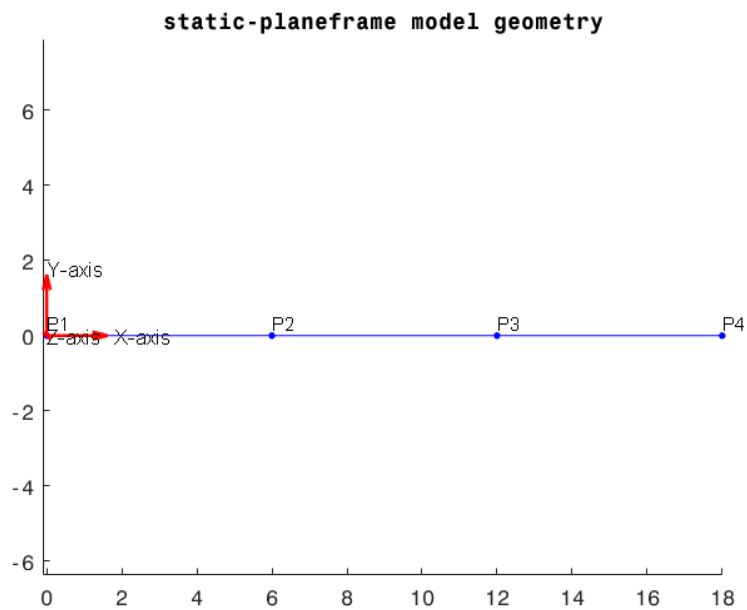
ESEMPIO 2 – PLOT LABELS ESEMPIO I

Per plottare le labels della geometria è necessario specificare l'argomento di output della funzione `saplab.gplot()`.

- Per plottare le *labels* relative ai punti della geometria digitare i comandi:

```
h = gplot(model.Geometry);
```

```
h.PointLabels('on')
```

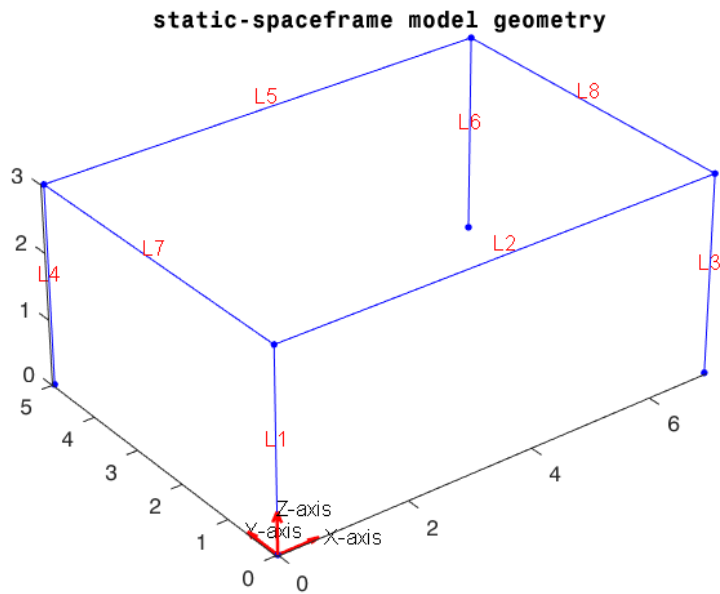


46 - Plot point labels

- Per plottare le *labels* relative alle **linee** della geometria digitare i comandi:

```
h = gplot(model.Geometry);
```

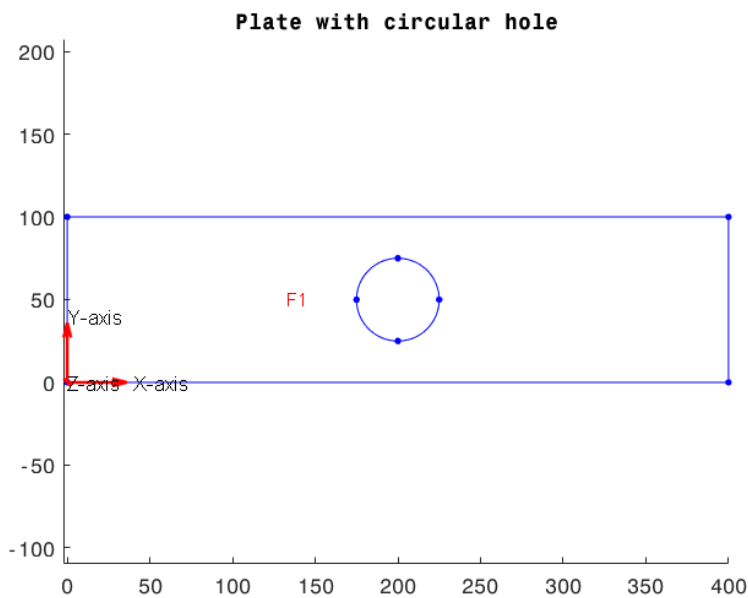
```
h.LineLabels('on');
```



47 - Plot line labels

ESEMPIO 3 – PLOT GEOMETRIA (STATIC-PLANESTRESS/PLANESTRAIN)

Creare un modello 'static-planestress' e modellare la geometria relativa ad una lastra con foro circolare.



48 - Plate with circular hole

```
import saplab.*;
```

```
model = create('attributes', 'static-planestress');
```

Aggiungere un rettangolo di dimensioni 400x100 e un cerchio di raggio pari a 25 posizionato al centro del suddetto rettangolo.

```
rectID = addrect(model, [0,400,0,100]);
```

```
circID = addcirc(model, [200,50], 25);
```

```
formula = strcat(rectID,'-',circID);
```

```
mGeom = booloper(model, formula);
```

```
model.Geometry = mGeom;
```

Per plottare a schermo la geometria del modello appena creato digitare il comando:

```
gplot(model.Geometry);
```

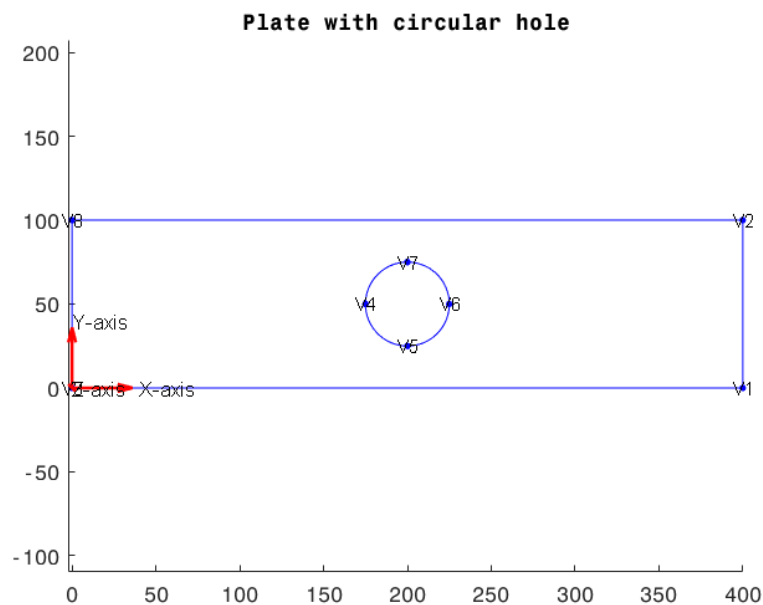
ESEMPIO 4 – PLOT LABELS ESEMPIO 3

Per plottare le labels della geometria è necessario specificare l'argomento di output della *funzione saplab.gplot()*.

- Per plottare, quindi, le *labels* dei vertici della geometria digitare i comandi:

```
h = gplot(model.Geometry);
```

```
h.VertexLabels('on');
```

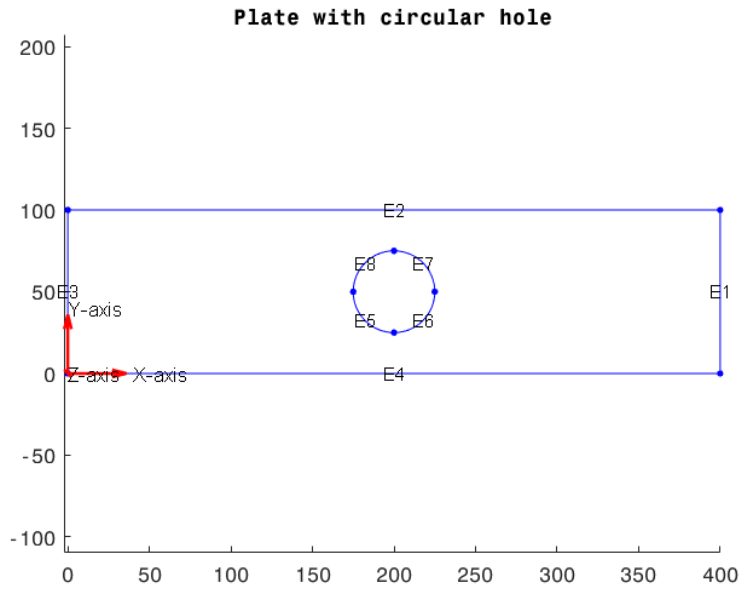


49 - Plate with circular hole with vertex labels on

- Per plottare, quindi, le *labels* dei bordi della geometria digitare i comandi:

```
h = gplot(model.Geometry);
```

```
h.EdgeLabels('on');
```

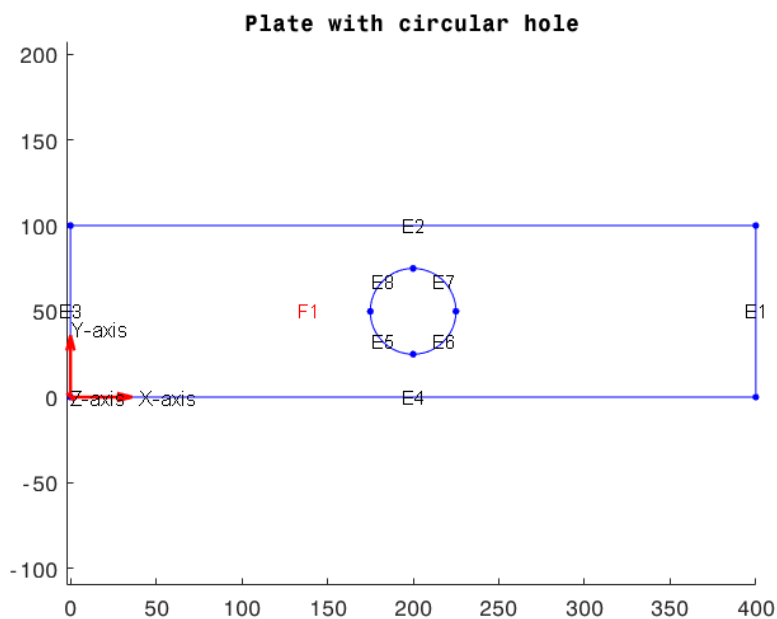


50 - Plate with circular hole with edge labels on

- Per plottare, quindi, la *label* della superficie piana della geometria digitare i comandi:

```
h = gplot(model.Geometry);
```

```
h.FaceLabels('on');
```



51 - Plate with circular hole with face labels on

ESEMPIO 5 – PLOTTARE UNA GEOMETRIA CSG

SAPLAB consente all'utente di lavorare con le geometrie di tipo CSG (*Constructive Solid Geometry*). Per maggiori informazioni sull'argomento vedere il link [2-D Geometry Creation at Command Line - MATLAB & Simulink - MathWorks Italia](#).

Si consideri la geometria dell'esempio n.3 descritta secondo il formato CSG:

```
PlateWithCircularHole =
```

```
    2.0000    2.0000    2.0000    2.0000    1.0000    1.0000    1.0000    1.0000
  400.0000  400.0000         0         0  175.0000  200.0000  225.0000  200.0000
  400.0000         0         0  400.0000  200.0000  225.0000  200.0000  175.0000
         0  100.0000         0         0   50.0000   25.0000   50.0000   75.0000
  100.0000  100.0000  100.0000         0   25.0000   50.0000   75.0000   50.0000
    1.0000    1.0000         0    1.0000         0         0         0         0
         0         0    1.0000         0    1.0000    1.0000    1.0000    1.0000
         0         0         0         0  200.0000  200.0000  200.0000  200.0000
         0         0         0         0   50.0000   50.0000   50.0000   50.0000
         0         0         0         0   25.0000   25.0000   25.0000   25.0000
```

Per applicare la geometria al modello strutturale digitare il comando:

```
gm = saplab.importCSG(PlateWithCircularHole);
```

La funzione ritorna in *output* l'oggetto *sap.GeometricModel2D*:

```
gm =
```

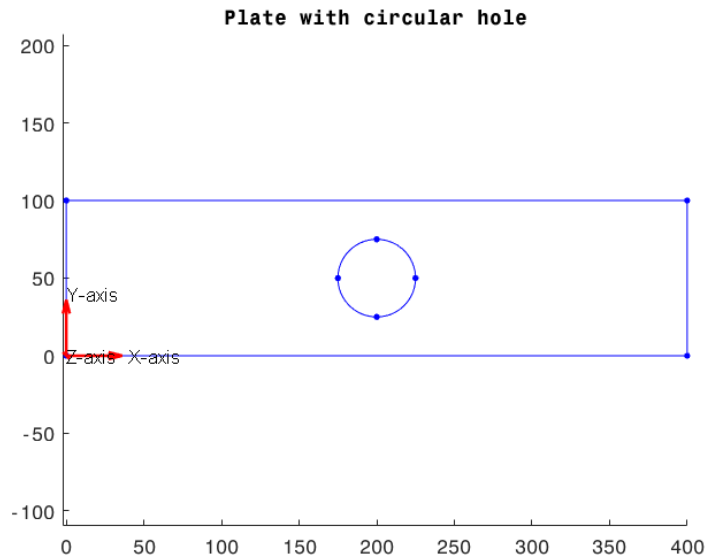
GeometricModel2D with properties:

```
    NumVertices: 8
    NumEdges: 8
    NumFaces: 1
    Vertices: [8x2 double]
```

A questo punto, dopo aver assegnato la geometria al modello strutturale, è possibile eseguire il *plot*.

```
model.Geometry = gm;
```

```
saplab.gplot(model);
```



52 - Plate with circular hole (CSG)

ESEMPIO 6 – PLOT GEOMETRIA (COUPLED-STATIC-PLANESTRESS)

Importare la seguente geometria CSG:

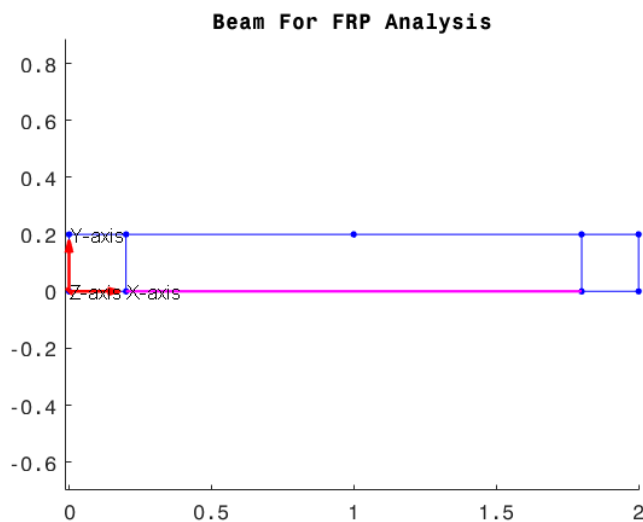
BeamForFRPAnalysis =

2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
0	2.0000	0.2000	0	0.2000	1.8000	0	0.2000	0.9995	1.8000	1.8000
0	2.0000	0.2000	0.2000	1.8000	2.0000	0.2000	0.9995	1.8000	2.0000	1.8000
0.2000	0	0	0	0	0	0.2000	0.2000	0.2000	0.2000	0
0	0.2000	0.2000	0	0	0	0.2000	0.2000	0.2000	0.2000	0.2000
2.0000	3.0000	2.0000	2.0000	1.0000	3.0000	0	0	0	0	1.0000
0	0	1.0000	0	0	0	2.0000	1.0000	1.0000	3.0000	3.0000

```
model.Geometry = sapLab.importCSG(BeamForFRPAnalysis);
```

Per plottare la geometria 'BeamForFRPAnalysis.mat' digitare il comando:

```
h = sapLab.gplot(model)
```



53 - Beam For FRP Analysis

N.B. La geometria espressa in formato CSG non contiene il rinforzo strutturale. Per assegnarne uno fare riferimento alle procedure indicate al para. ([Analisi FEM con elementi FRP - Creare un rinforzo strutturale](#)).

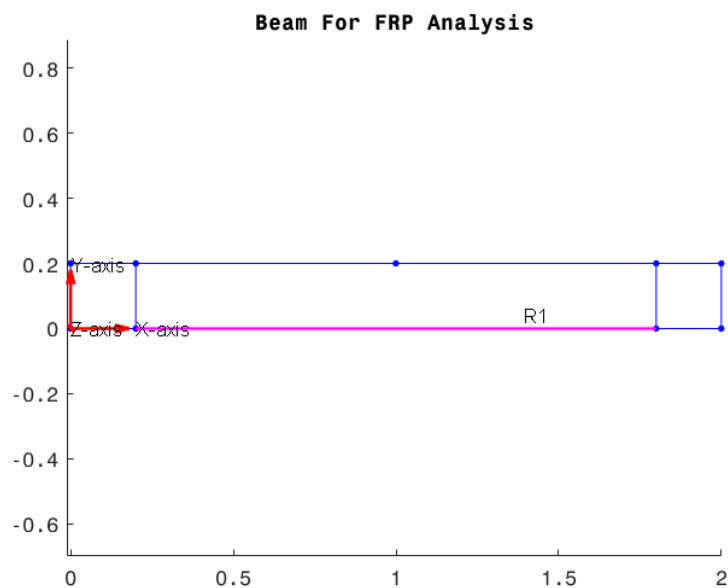
ESEMPIO 7 – PLOT LABELS ESEMPIO 6

La procedura per plottare a schermo i labels è analoga a quella vista per l'esempio n.3.

In questo caso, tuttavia, è possibile osservare la *label* relativa al rinforzo strutturale inserito nel modello. Per effettuare tale operazione digitare il comando:

```
h = sapLab.gplot(model)
```

```
h.ReinforcementLabels('on');
```



54 - Beam For FRP Analysis with Labels on

Visualizzazione della mesh

Nel paragrafo precedente sono state descritte le procedure necessarie per visualizzare graficamente i dati e la geometria del modello strutturale. Ora si procederà a mostrare le operazioni per plottare e rappresentare la mesh della geometria del su citato modello e a ispezionare i dati contenuti all'interno dell'oggetto `sap.DiscretizedFEModel`. Quest'ultimo contiene al suo interno tutti i dati relativi alla mesh e generati dall'utente mediante i comandi indicati al para. [Discretizzazione della geometria](#).

All'interno del presente capitolo:

- [Plottare la mesh di un modello SAPLAB](#);

PLOTTARE LA MESH DI UN MODELLO SAPLAB

Creare una mesh digitando il comando `saplab.mesh(_)`.

```
msh = sapLab.mesh(model, args{:})
```

```
msh =
```

```
DiscretizedFEModel with properties:
```

```
    Elements: [3×420 double]  
      Nodes: [2×251 double]  
MinElementSize: 0.2000  
MaxElementSize: 0.4000  
GeometricOrder: 'linear'
```

L'oggetto *mesh* generato da SAPLAB contiene 420 elementi e 251 nodi. Per plottare la mesh associata alla geometria del modello strutturale digitare il comando di seguito mostrato.

```
sapLab.mplot(__)
```

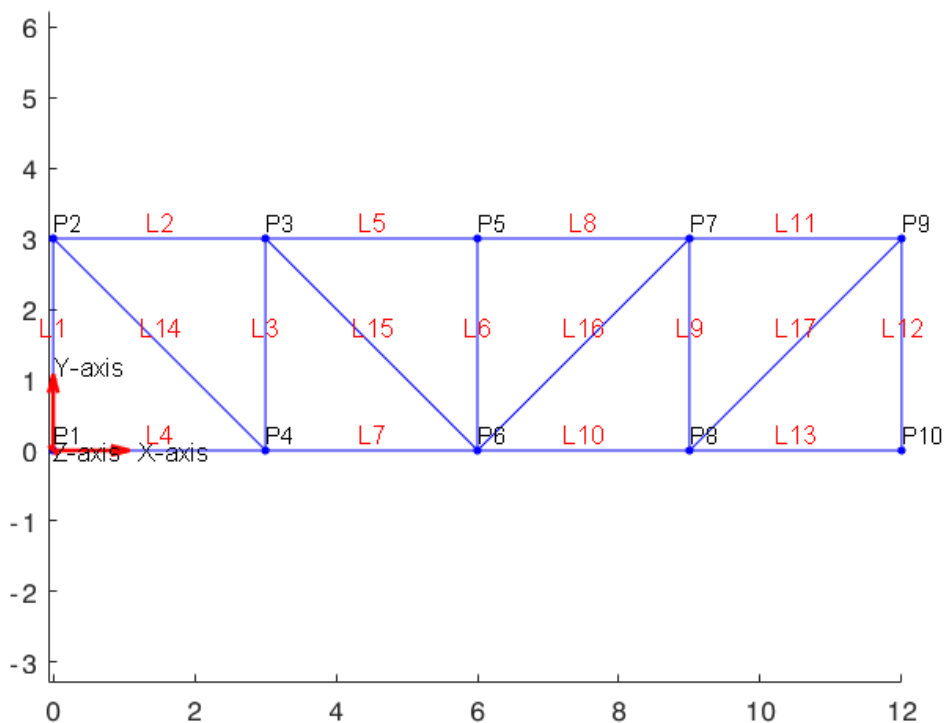
Sintassi e Descrizione

SINTASSI 1	<pre>msh = sapLab.mplot(model)</pre> <p>Plotta la <i>mesh</i> della geometria associata al modello strutturale <i>model</i>.</p>
SINTASSI 2	<pre>msh = sapLab.mplot(model, 'ElementLabels', OnOff)</pre> <p>Plotta la <i>mesh</i> della geometria associata al modello strutturale <i>model</i> attivando le etichette associate agli elementi finiti della <i>mesh</i>.</p>

	<code>msh = sapLab.mplot(model, 'NodeLabels', OnOff)</code>
SINTASSI 3	Plotta la <i>mesh</i> della geometria associata al modello strutturale <i>model</i> attivando le etichette associate ai nodi della <i>mesh</i> .
	<code>msh = sapLab.mplot(model, 'FRPElementLabels', OnOff)</code>
SINTASSI 4	Plotta la <i>mesh</i> della geometria associata al modello strutturale <i>model</i> attivando le etichette associate agli elementi dei rinforzi strutturali
	<code>msh = sapLab.mplot(model, 'FRPElementLabels', OnOff)</code>
SINTASSI 5	Plotta la <i>mesh</i> della geometria associata al modello strutturale <i>model</i> attivando le etichette associate ai nodi dei rinforzi strutturali

ESEMPIO DI VISUALIZZAZIONE DELLA MESH (GEOMETRIA ID)

Si consideri, a titolo di esempio, una trave reticolare di tipo Pratt (*modello static-planeframe*) di seguito raffigurata.



55 - Trave reticolare di tipo Pratt

Si generi una mesh attraverso il comando:

```
msh = sapLab.mesh(model, 'Line', 1:17, 'ElementSize', 0.7);
```

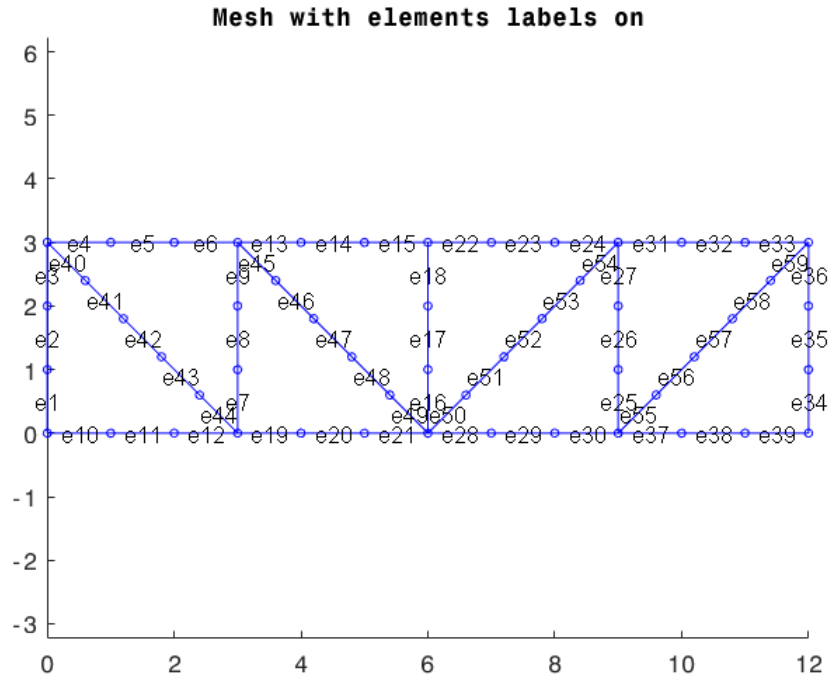
Per plottare la mesh digitare i comandi di seguito riportati.

```
sapLab.mplot(model, 'ElementLabels', 'on');
```

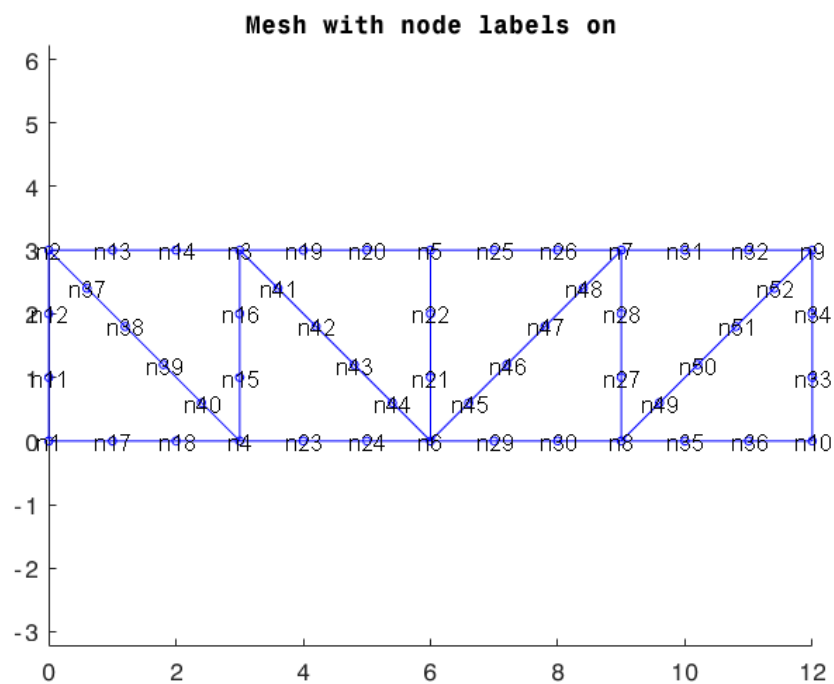
```
sapLab.mplot(model, 'NodeLabels', 'on');
```

Nel primo caso SAPLAB plotta la mesh della geometria evidenziando le etichette degli elementi mentre nel secondo caso il programma visualizza la mesh della geometria del modello mettendo in evidenza le etichette dei nodi.

I risultati vengono raffigurati a pagina seguente.



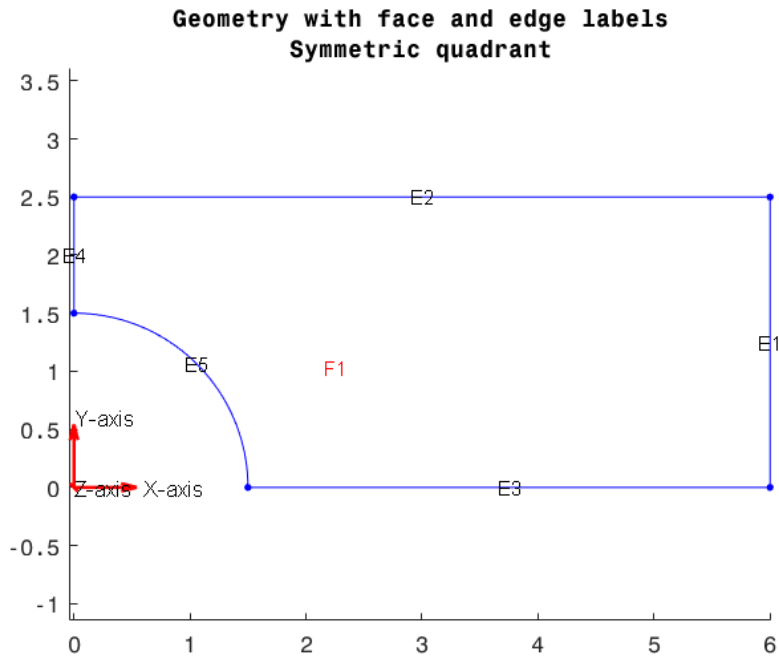
56 - Mesh with element labels on



57 - Mesh with node labels on

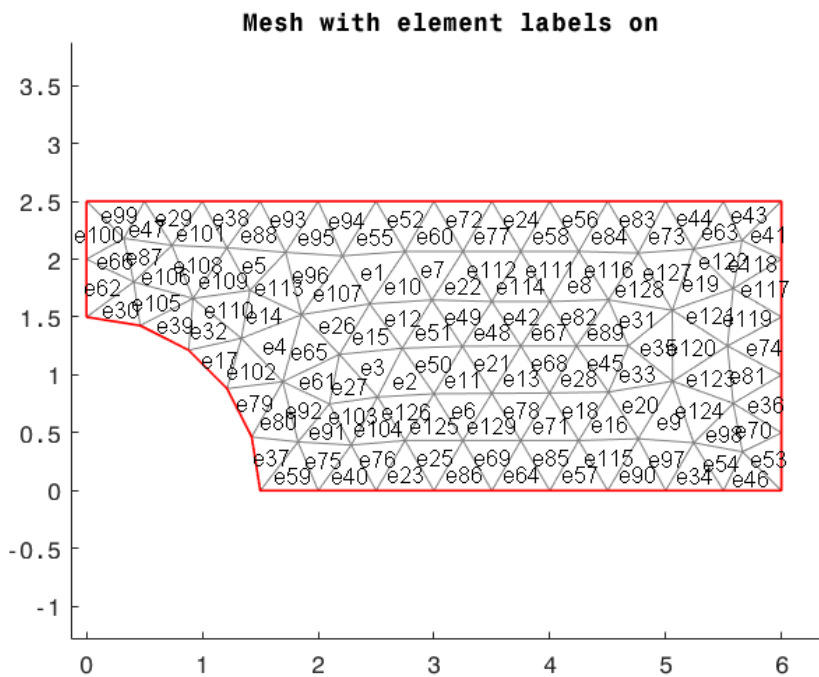
ESEMPIO DI VISUALIZZAZIONE DELLA MESH (GEOMETRIA 2D)

Si consideri una lastra con foro circolare.

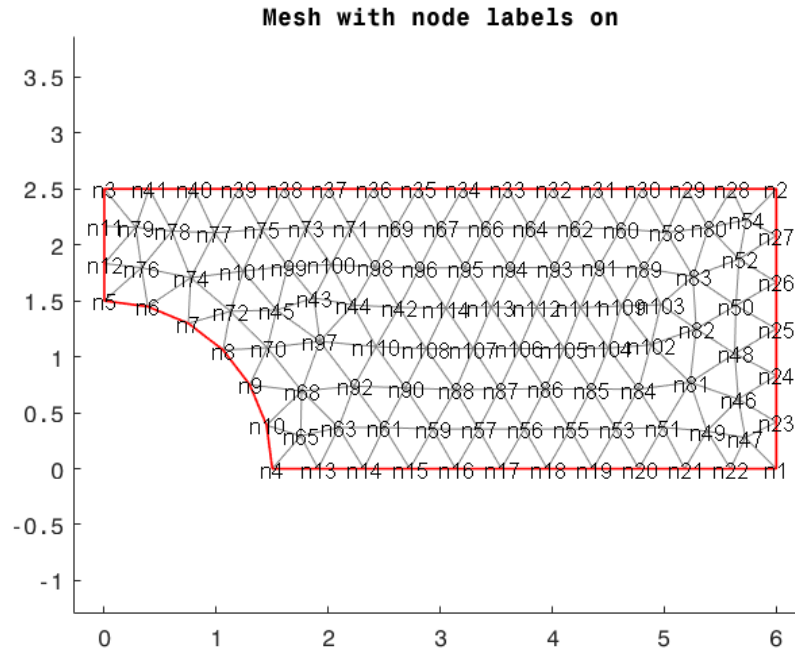


58 - Plate with circular hole

Generando una mesh della struttura sopra raffigurata e plottandone il risultato fornito dalla discretizzazione della superficie si ottiene:



59 - Mesh with element labels on



60 - Mesh with nodes labels on

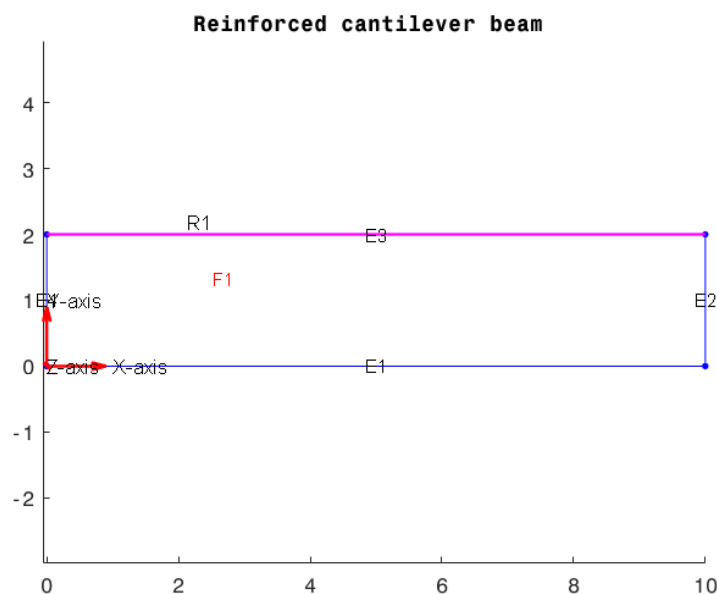
```

msh = sapLab.mesh(model);
saplab.mplot(model, 'ElementLabels', 'on');
saplab.mplot(model, 'NodeLabels', 'on');

```

ESEMPIO DI VISUALIZZAZIONE DELLA MESH (GEOMETRIA 2D E RINFORZO STRUTTURALE)

Si suppone di voler rappresentare la mesh di un modello di tipo *coupled-static-planestress* come quello raffigurato sotto.



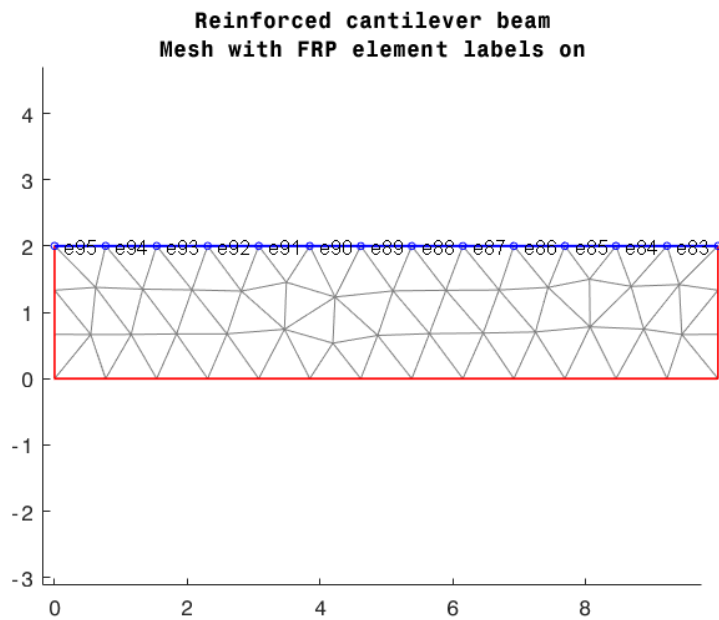
61 - Reinforced cantilever beam

Digitando i comandi già trattati negli esempi precedenti è possibile ispezionare le proprietà della mesh relative alla superficie piana F1.

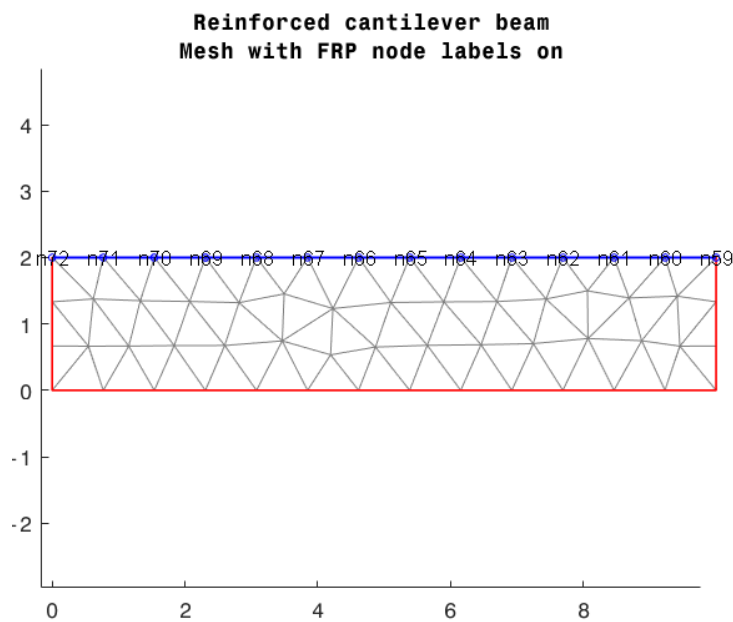
Analogamente è possibile ispezione le proprietà della mesh relative al rinforzo strutturale R1. Per effettuare tale operazione è necessario digitare i comandi indicati di seguito:

```
sapLab.mplot(model, 'FRPElementLabels', 'on');
```

```
sapLab.mplot(model, 'FRPNodeLabels', 'on');
```



62 - Mesh with element labels on



63 - Mesh with node labels on

Nel primo caso SAPLAB plotta la *mesh* della geometria evidenziando le etichette degli elementi relativi al rinforzo strutturale mentre nel secondo caso il programma visualizza la *mesh* della geometria del modello mettendo in evidenza le etichette dei nodi appartenenti al rinforzo.

Soluzione del modello FEM

Per ricavare la soluzione del modello agli elementi finiti scritto in SAPLAB è possibile usufruire di due tipologie di risolutori *saplab.elinSolver()* e *saplab.nlinSolver()*.

Il primo può essere utilizzato per risolvere problemi elastici lineari ed è applicabile per tutte le tipologie di analisi. Il secondo, invece, può essere impiegato nella risoluzione di problemi non lineari (linearità dei materiali) e per il momento può essere applicato esclusivamente a modelli di tipo *'coupled-static-planestress'*.

Analisi	elinSolver()	nlinSolver()
static-planeframe	✓	✗
static-spaceframe	✓	✗
static-planestress	✓	✗
static-planestrain	✓	✗
coupled-static-planestress	✓	✓

Tabella 4 - Risolutori SAPLAB

RISOLUTORE FEM: ELINSOLVER()

Per risolvere un problema elastico lineare digitare il comando:

saplab.elinSolver().

Sintassi e Descrizione

SINTASSI I	$R = \text{saplab.elinSolver}(\text{model})$ Risolve il modello strutturale <i>model</i> e fornisce in output l'oggetto <i>sap.StructuralResults</i> contenente i risultati dell'analisi FEM.
------------	--

RISOLUTORE FEM: NLINSOLVER()

Per risolvere un problema elastico lineare digitare il comando:

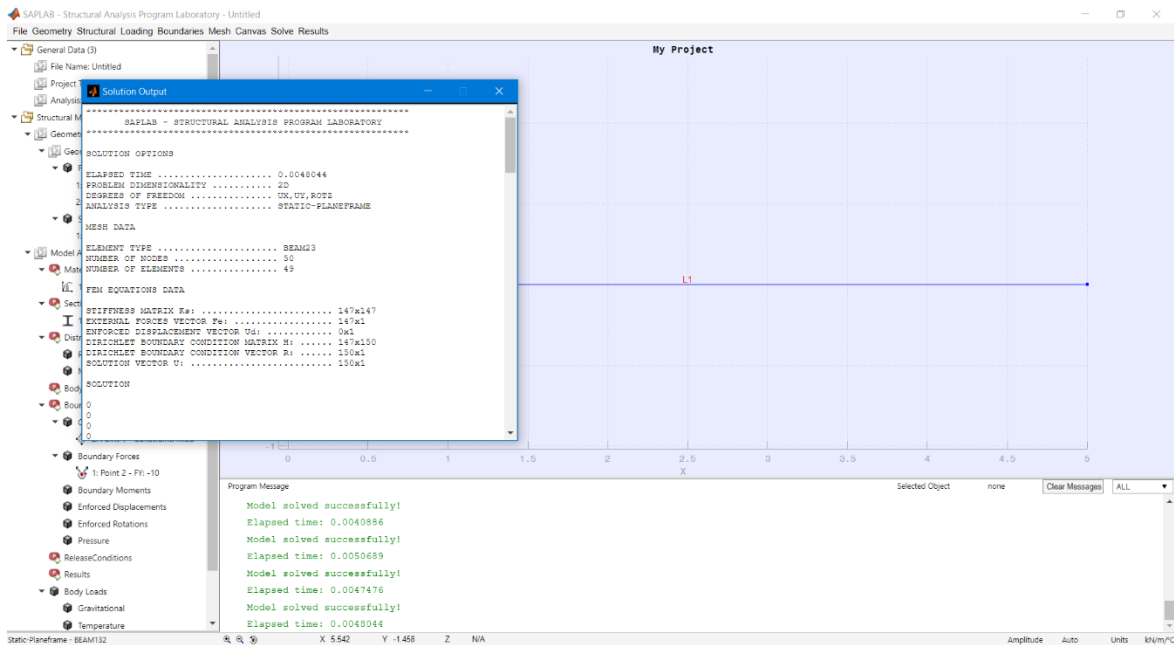
saplab.nlinSolver().

$$R = \text{sapLab.nlinSolver}(\text{model})$$

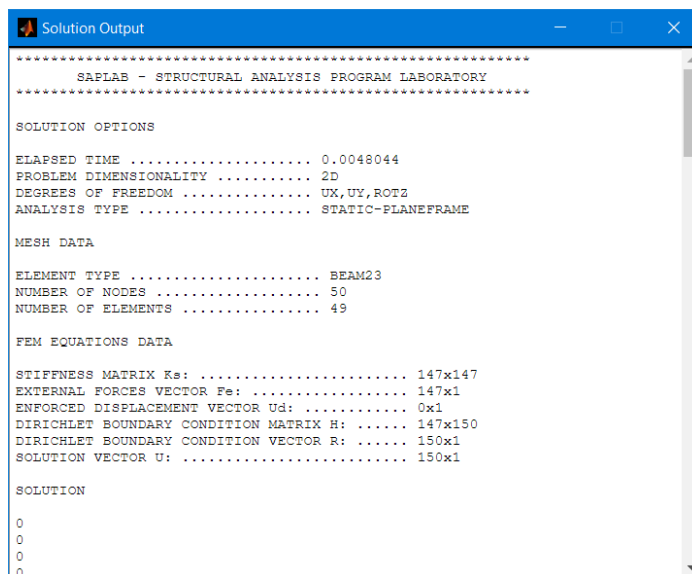
SINTASSI I

Risolve il modello strutturale *model* considerando le eventuali non linearità dei materiali definiti nel modello e fornisce in output l'oggetto [*sap.StructuralResults*](#) contenente i risultati dell'analisi FEM.

Per risolvere il modello strutturale dall'interfaccia utente e il bordo cliccare sulla voce *Solve* → *<Elastic Linear Solver (ELINSOLVER)>/<Non Linear Solver (NLINSOLVER)>* della *Menu Bar*.



64 - Solve output



65 - Solution Output

Una volta calcolata la soluzione, il programma fornisce in *output* un riassunto dei dati relativi al modello strutturale nella *form Solution Output*.

Post-Processing dei risultati

I risultati del modello FEM vengono salvati all'interno della variabile di *output* del risolutore utilizzato (Vedi [Soluzione del modello FEM](#)). Questa contiene l'oggetto della classe *sap.StructuralResults*.

```
R = saplab.elinSolver(model)
```

Per osservare i risultati ottenuti ispezionare il contenuto della variabile *R*.

All'interno del presente capitolo:

- [La soluzione del modello FEM in termini di spostamenti nodali](#);
- [I diagrammi di sollecitazione](#);
- [Stress/Strain plot: Diagrammare le tensioni e le deformazioni](#);
- [Le tensioni di contatto tra un rinforzo strutturale in FRP e trave](#).

LA FUNZIONE RPLOT()

In SAPLAB la rappresentazione dei risultati viene effettuata attraverso la funzione `saplab.rplot()`.

		Sintassi e Descrizione
	<code>saplab.rplot(model, __)</code>	
SINTASSI 1	Visualizza i risultati del modello strutturale <i>model</i> passato come primo argomento di input alla funzione.	
	<code>saplab.rplot(__, 'Deformation')</code>	
SINTASSI 2	Visualizza la deformata della struttura.	
	<code>saplab.rplot(__, 'XYData', Field)</code>	
SINTASSI 3	Specificando la voce <i>XYData</i> viene generata una mappa a colori del risultato passato in input alla funzione nell'argomento <i>Field</i> . Ogni valore contenuto in quest'ultimo deve essere associato ad un nodo della struttura.	
	<code>saplab.rplot(__, 'Diagram', DiagramValue)</code>	
SINTASSI 4	Specificando la voce <i>Diagram</i> viene visualizzato il diagramma di sollecitazione corrispondente definito dall'argomento <i>DiagramValue</i> . Tale opzione vale solo per i modelli <i>static-planeframe</i> e <i>static-spaceframe</i> .	

La funzione in oggetto, fatto eccezione per la sintassi n. 2, deve ricevere in *input* i risultati contenuti nell'oggetto *sap.StructuralResults* generato in *output* dal risolutore FEM. L'oggetto dei risultati contiene le proprietà:

- *NodalSolution* che contiene i valori nodali delle componenti di spostamento;
- *ForcesAndMoments* che contiene i valori di sollecitazione degli elementi di tipo trave;
- *NodalStress* che contiene i valori di tensione nodali della *mesh* riferite agli assi X e Y globali;
- *NodalStrain* che contiene i valori di deformazione nodali della *mesh* riferite agli assi X e Y
- *VonMisesStress* che contiene i valori di tensione nodali della *mesh* riferite ai piani principali;
- *TangentialStress* che contiene i valori di tensione tangenziali che si generano all'interfaccia di collegamento delle travi con FRP.

LA SOLUZIONE DEL MODELLO FEM – GLI SPOSTAMENTI NODALI

Per ricavare la soluzione relativa agli spostamenti nodali del sistema analizzato digitare il comando:

```
nodalSolution = R.NodalSolution;
```

```
nodalSolution =  
  
    struct with fields:  
  
    ux: [50×1 double]  
    uy: [50×1 double]  
    rz: [50×1 double]
```

Figura 66 - Struttura Dati *NodalSolution*

Tale operazione salva all'interno della variabile *nodalSolution* una struttura di dati (Vedi link [Structure array - MATLAB - MathWorks Italia](#)) contenente gli spostamenti ricavati dalla soluzione del modello FEM.

I campi della struttura di dati contenuta all'interno della proprietà *NodalSolution* dell'oggetto *R* (*sap.StructuralResults*) possono variare a seconda della tipologia di analisi effettuata. Maggiori informazioni in merito sono contenute nella tabella riportata di seguito.

Analisi del modello	Campo					
	ux	uy	uz	rx	ry	rz
'static-planeframe'	✓	✓	✗	✗	✗	✓
'static-spaceframe'	✓	✓	✓	✓	✓	✓
'static-planestress'	✓	✓	✗	✗	✗	✗
'static-planestrain'	✓	✓	✗	✗	✗	✗
'coupled-static-planestress'	✓	✓	✗	✗	✗	✗

Tabella 5 - Campi di spostamento

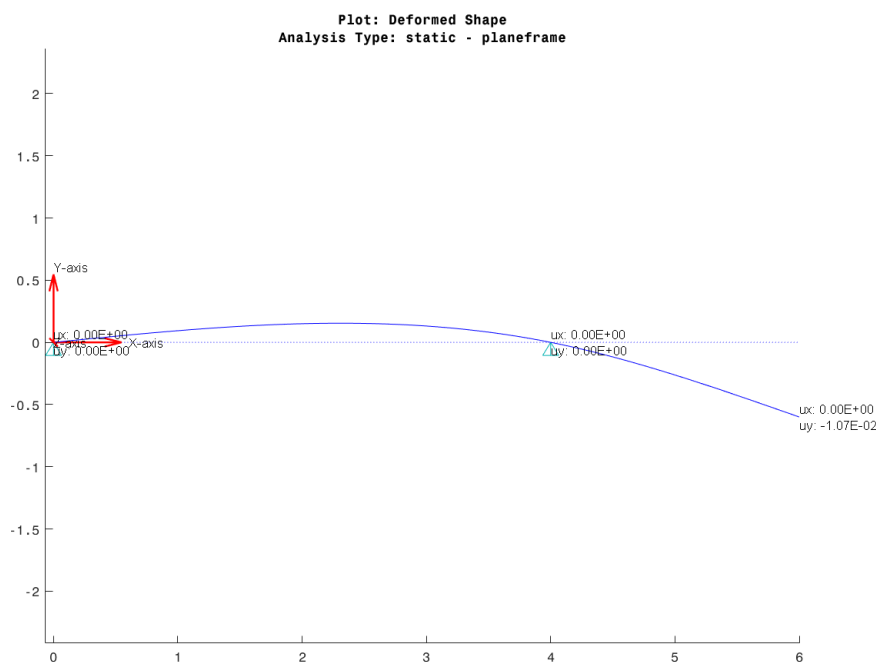
PLOTTARE GLI SPOSTAMENTI NODALI

La variabile *nodalSolution* contiene gli spostamenti calcolati rispetto agli assi X e Y globali e le rotazioni rispetto all'asse Z globale. Per diagrammare i risultati contenuti nella variabile *nodalSolution* digitare il comando:

```
sapLab.rplot(model, 'Deformation');
```

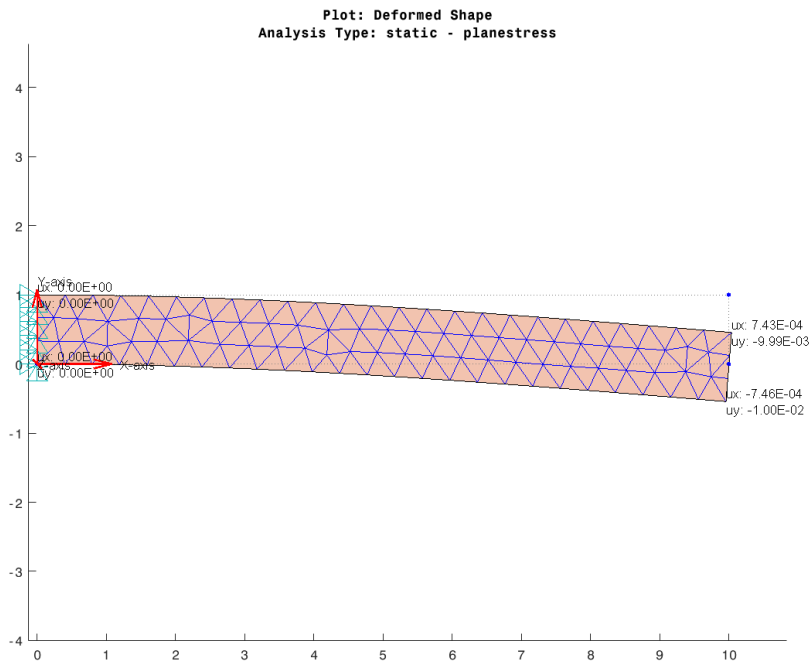
Di seguito si riporta:

- il risultato ottenuto analizzando una struttura composta da una campata interna e una trave in semplice appoggio dotata di uno sbalzo terminale e caricata all'estremità di quest'ultimo da una forza concentrata verticale ('static-planeframe').



67 – Deformation results (a)

b) Il risultato ottenuto analizzando una trave a sbalzo (*Cantilever Beam*) con elementi triangolari di tipo *PLANESTRESS* soggetta ad uno spostamento imposto pari a -0.01 m applicato alla sua estremità.



68 - Deformation results (b)

LE SOLUZIONI PER ELEMENTO FINITO

La soluzione del modello FEM può essere ottenuta anche raggruppando gli spostamenti nodali per ogni elemento finito presente nel sistema. Per effettuare tale operazione digitare il comando:

```
elemSolution = R.ElementSolution;
```

```
elemSolution =
```

```
struct with fields:
```

```
ux: [2×118 double]
uy: [2×118 double]
rz: [2×118 double]
```

Figura 69 - Struttura di dati ElementSolution

È possibile notare come ogni campo della struttura di dati contenga una matrice di dimensione pari a $n \times N$ (con N pari al numero di elementi finiti presenti nel sistema e n pari al numero di nodi connessi da ogni elemento finito).

Per quanto concerne l'analisi 'coupled-static-planestress', la struttura di dati contenuta nella proprietà *ElementSolution* presenta una forma differente rispetto a quella sopra riportata. Ciò deriva dalla natura ibrida del modello che prevede l'accoppiamento di elementi finiti di diversa tipologia che connettono una quantità di nodi differente (3 per gli elementi triangolari di tipo *PLANE-STRESS* a 2 per gli elementi di tipo *TRUSS*).

```
elemSolution = R.ElementSolution;

elemSolution =

    struct with fields:

        Triangle: [1x1 struct]
        Truss: [1x1 struct]
        Spring: [1x1 struct]
```

Figura 70 - Struttura Dati Element Solution

Per ottenere gli spostamenti raggruppati per ogni elemento finito triangolare di tipo *PLANE-STRESS* presente nel modello digitare il comando:

```
tri = elemSolution.Triangle;

tri =

    struct with fields:

        ux: [3x9158 double]
        uy: [3x9158 double]
```

Figura 71 - Campo *Triangle*

Per ottenere gli spostamenti raggruppati per ogni elemento finito di tipo *TRUSS* presente nel modello digitare il comando:

```
bars = elemSolution.Truss;

bars =

    struct with fields:

        ux: [2x160 double]
        uy: [2x160 double]
```

Figura 72 - Campo *Truss*

Per ottenere gli spostamenti raggruppati per ogni elemento finito di tipo *SPRING* presente nel modello digitare il comando:

```

spring = elemSolution.Spring;

spring =
    struct with fields:
        ux: [2×161 double]
        uy: [2×161 double]

```

Figura 73 - Campo Spring

I DIAGRAMMI DI SOLLECITAZIONE

Per ottenere in *output* i diagrammi di sollecitazione (opzione valida esclusivamente per i modelli del tipo *static-planeframe* o *static-spaceframe*) digitare il comando:

```
stressValues = R.ForcesAndMoments;
```

Ciò che si ottiene è una struttura di dati del tipo indicato di seguito.

```

stressValue =
    struct with fields:
        Nx: {2×1 cell}
        Ty: {2×1 cell}
        Mz: {2×1 cell}

```

Figura 74 - Struttura Dati Force And Moments

I campi della struttura di dati contenuta all'interno della proprietà *ForcesAndMoments* dell'oggetto R (*sap.StructuralResults*) possono variare a seconda della tipologia di analisi effettuata. Maggiori informazioni in merito sono contenute nella tabella riportata di seguito.

Campo	Nx	Ty	Tz	Mx	My	Mz
Analisi del modello						
'static-planeframe'	✓	✓	✗	✗	✗	✓
'static-spaceframe'	✓	✓	✓	✓	✓	✓

Tabella 6 - Campi della struttura di dati ForcesAndMoments

PLOTTARE IL DIAGRAMMA DI SFORZO NORMALE

Per plottare il diagramma dello sforzo normale digitare il comando indicato di seguito.

```
sapLab.rplot(model, 'diagrams', R.ForcesAndMoments.Nx);
```

PLOTTARE IL DIAGRAMMA DELLO SFORZO DI TAGLIO

Digitare il comando n.1 per mostrare a schermo il diagramma dello sforzo di taglio agente lungo l'asse Y locale della trave mentre digitare il comando n.2 per plottare il diagramma relativo allo sforzo di taglio agente lungo l'asse Z locale della trave.

1. `sapLab.rplot(model, 'diagrams', R.ForcesAndMoments.Ty);`
2. `sapLab.rplot(model, 'diagrams', R.ForcesAndMoments.Tz);`

PLOTTARE IL DIAGRAMMA DEL MOMENTO FLETTENTE

Digitare il comando n.1 per mostrare a schermo il diagramma del momento flettente agente attorno l'asse Y locale della trave mentre digitare il comando n.2 per plottare il diagramma relativo al momento flettente agente attorno l'asse Z locale della trave.

1. `sapLab.rplot(model, 'diagrams', R.ForcesAndMoments.My);`
2. `sapLab.rplot(model, 'diagrams', R.ForcesAndMoments.Mz);`

PLOTTARE IL DIAGRAMMA DEL MOMENTO TORCENTE

Per plottare a schermo il diagramma del momento torcente digitare il comando;

```
sapLab.rplot(model, 'diagrams', R.ForcesAndMoments.Mx);
```

STRESS/STRAIN PLOT: DIAGRAMMARE TENSIONI E DEFORMAZIONI

Per ottenere i valori di tensione e/o deformazione (opzione valida esclusivamente per i modelli del tipo *'static-planestress'*, *'static-planestrain'* e *'coupled-static-planestress'*) digitare i comandi indicati di seguito:

```
nodalStress = R.NodalStress;  
vmStress = R.VonMisesStress;  
nodalStrain = R.NodalStrain;
```

Le variabili *nodalStress/nodalStrain/vmStress* conterranno una struttura di dati con all'interno i valori delle componenti di tensione/deformazione come visto in precedenza.

In tabella seguente vengono elencati i campi delle strutture di dati relative alle proprietà dell'oggetto `sap.StructuralResults`, *NodalStress*, *NodalStrain* e *VonMisesStress*.

Campo	sxx	syx	syy	exx	eyx	exy	s11	s22	s12
Proprietà									
<i>Nodal Stress</i>	✓	✓	✓	✗	✗	✗	✗	✗	✗
<i>Von Mises Stress</i>	✗	✗	✗	✗	✗	✗	✓	✓	✓
<i>Nodal Strain</i>	✗	✗	✗	✓	✓	✓	✗	✗	✗

Tabella 7 - Campi delle proprietà di tensione/deformazione

Per rappresentare graficamente i valori desiderati passare in input alla funzione [saplab.rplot\(\)](#) la struttura di dati da plottare come mostrato di seguito.

```
saplab.rplot(model, 'XYData', R.NodalStress.sxx);
```

LE TENSIONI DI CONTATTO TRA RINFORZO STRUTTURALE IN FRP E TRAVE

Per ottenere i valori delle tensioni di contatto che si sviluppano tra il rinforzo strutturale in FRP e la trave (in C.A.) digitare il comando:

```
tauStress = R.TangentialStress;
```

struct with fields:

```
Stress: [-0.0069 -0.0047 -0.0036 -0.0029 -0.0025 -0.0022 -0.0020 -0.0018 -0.0017 -0.0016 -0.0015 ... ]
PeakStress: 0.0069
MinStress: 1.1926e-04
Stiffness: [108x1 double]
Elongation: [108x1 double]
AbsoluteLocation: [0 14.9533 29.9065 44.8598 59.8131 74.7664 89.7196 104.6729 119.6262 134.5794 149.5327 ... ]
PeakElongation: 0.0258
Data: [108x3 table]
```

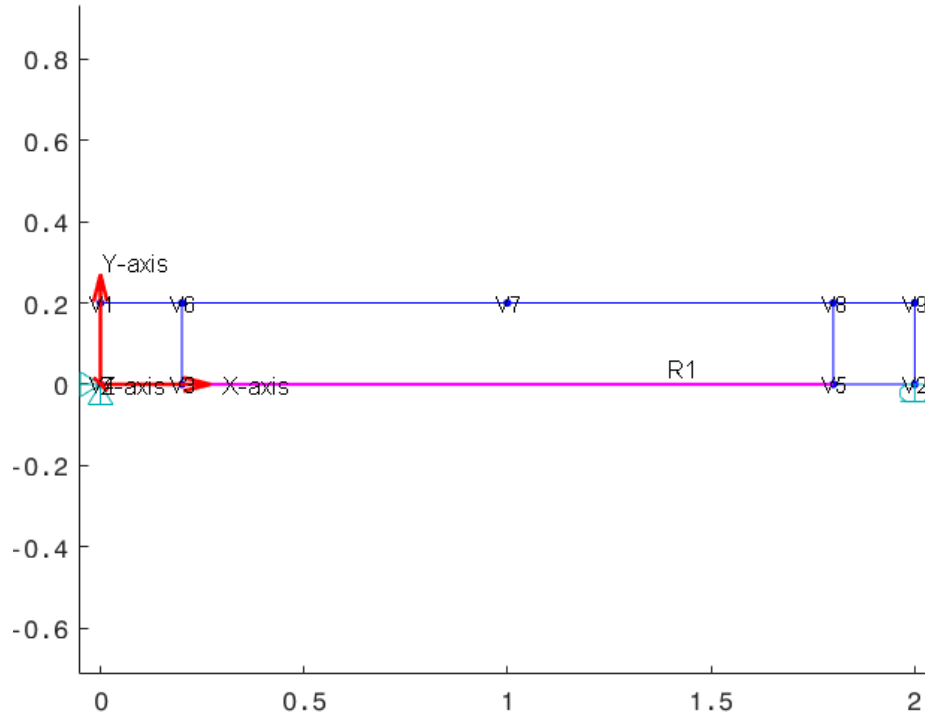
Figura 75 - Struttura di dati TangentialStress

tauStress è una struttura di dati contenente i seguenti campi:

- *Stress*: contiene i valori di tensione lungo la regione di contatto tra trave e rinforzo;
- *PeakStress*: Contiene il valore di picco della tensione tangenziale;
- *MinStress*: contiene il valore minimo di tensione;
- *Stiffness*: contiene i valori di rigidità di ogni molla che simula lo strato adesivo (o interfaccia di collegamento);
- *Elongation*: contiene i valori di elongazione di ogni molla che simula lo strato adesivo (o interfaccia di collegamento);

A titolo di esempio, si diagrammano i risultati ottenuti dall'analisi della struttura rappresentata in figura soggetta ad uno spostamento imposto applicato al vertice n. 7.

Simply Supported Beam With FRP Structural Reinforcement



76 - Beam with structural reinforcement

Digitando i comandi riportati di seguito è possibile diagrammare le tensioni che si sviluppano all'interfaccia tra il rinforzo strutturale e la trave.

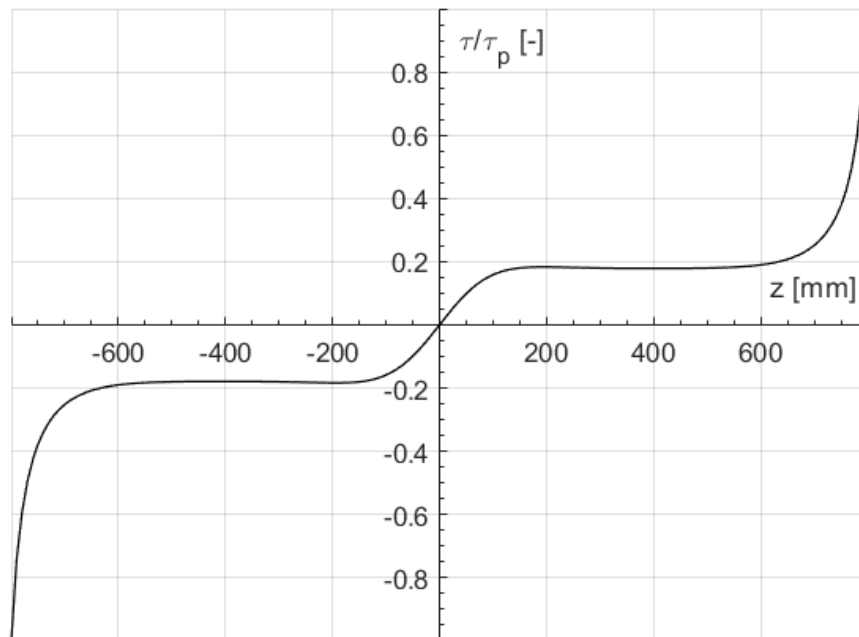


Figura 77 - Tensioni tangenziali

Si riportano di seguito i comandi MATLAB® da effettuare per graficare le tensioni tangenziali contenute all'interno della struttura di dati *TangentialStress*.

```
tStress = R.TangentialStress.Stress;  
maxStress = R.TangentialStress.PeakStress;  
tAdim = tStress/PeakStress;  
sLoc = R.TangentialStress.AbsoluteLocation;  
plot(sLoc-mean(sLoc), tAdim);
```

Il risultato ottenuto è riportato in figura 77.