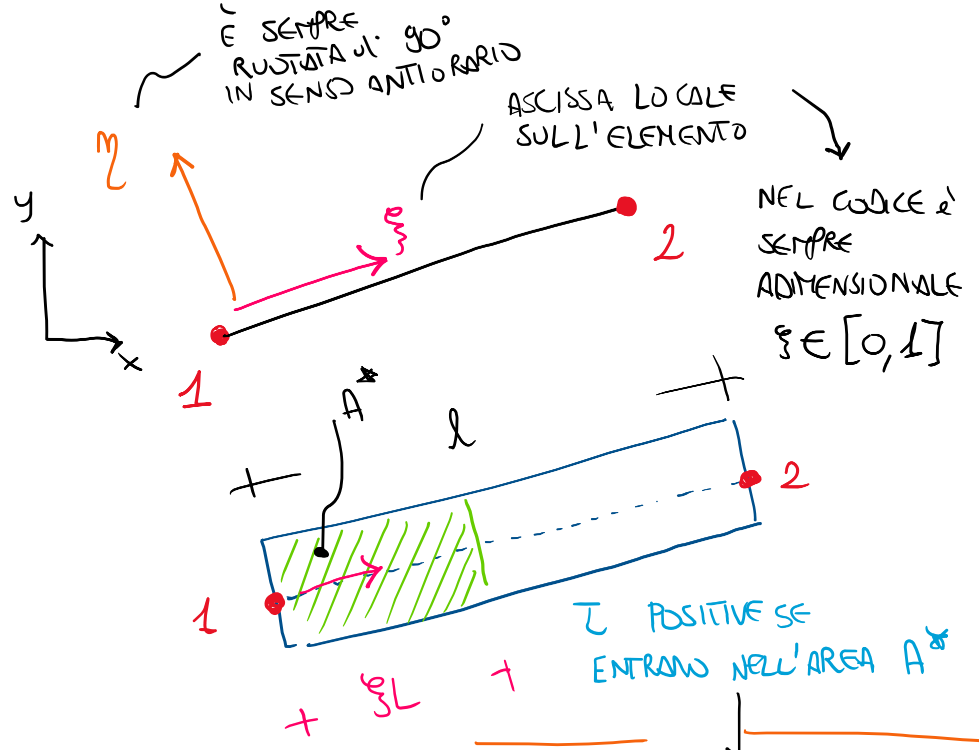
Istruzioni per il codice Saint-Venant

Marco Rossi

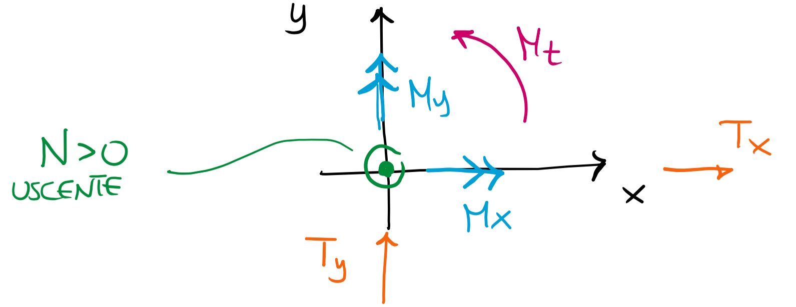
CONVENZIONE SEGNI e SIST. di RIFERIMENTO



LE σ SONO POSITIVE SE SONO SFORZI di TRAZIONE

IL SEGNO delle τ è POSITIVO QUANDO SONO DIRETTE VERSO IL PRIMO NODO dell'ELEMENTO (ANCHE LE τ di TORSIONE!)

CONVENZIONI sulle AZIONI INTERNE



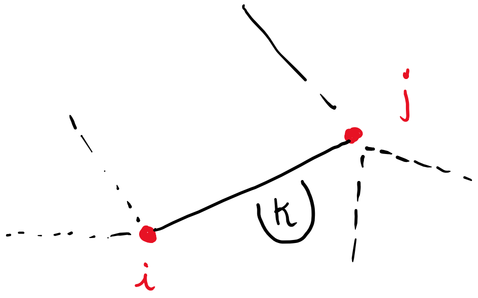
DEFINIZIONE della GEOMETRIA

SECTION.nodes = $\begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_i & y_i \\ \vdots & \vdots \\ x_j & y_j \\ \vdots & \vdots \end{bmatrix}$

MATRICE $N \times 2$ ← NUMERO NODI

← RIGA i ⟹ COORD. NODO i

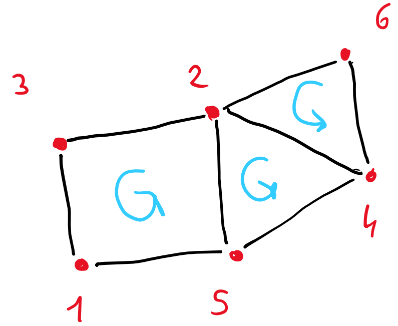
← RIGA j ⟹ COORD. NODO j



SECTION.elem = $\begin{bmatrix} 1 & 2 \\ 10 & 15 \\ \vdots & \vdots \\ i & j \\ \vdots & \vdots \end{bmatrix}$

MATRICE $M \times 2$ ← NUMERO ELEMENTI

← RIGA K ⇓ NODI dell'ELEMENTO K



$C_1 = [1, 5, 2, 3]$ $\quad C_2 = [2, 5, 4]$

$C_3 = [6, 2, 4]$

SECTION.thickness = $\begin{bmatrix} t \\ 2t \\ \vdots \end{bmatrix}$ ← RIGA K ⇓ SPESSORE ELEMENTO K

SECTION.closedPath = $= \{C_1, C_2, C_3\}$

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR | PUBLISH | VIEW

New | Open | Save | Compare | Print — FILE
Go To | Find | Bookmark — NAVIGATE
Refactor — CODE
Profiler | Analyze — ANALYZE
Run Section | Section Break | Run and Advance | Run to End — SECTION
Run | Step | Stop — RUN

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```matlab

%% *************************************************************************************************************
% ************                                  INITIAL DEFINITIONS                                  ************
% *************************************************************************************************************

clear all
close all
clc

%define the symbolic variables
global x y
syms x y
syms xi eta

%define the symbolic parameters
syms B positive real
syms H positive real
syms tf positive real
syms tw positive real




%%
%% *************************************************************************************************************
% ************                               GEOMETRY OF THE CROSS SECTION                               ************
% *************************************************************************************************************
fprintf('- define the cross section...\n');

%define the nodes of the section
SECTION.nodes = [
            0,      0;
            B/2,    0;
            B,      0;
            0,      H;
            B/2,    H;
            B,      H;
```

VARIABILI SIMBOLICHE
(DA NON UTILIZZARE PER I PARAMETRI)

QUI VANNO DEFINITI I PARAMETRI
SIMBOLICI del PROBLEMA

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 | Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR | PUBLISH | VIEW

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

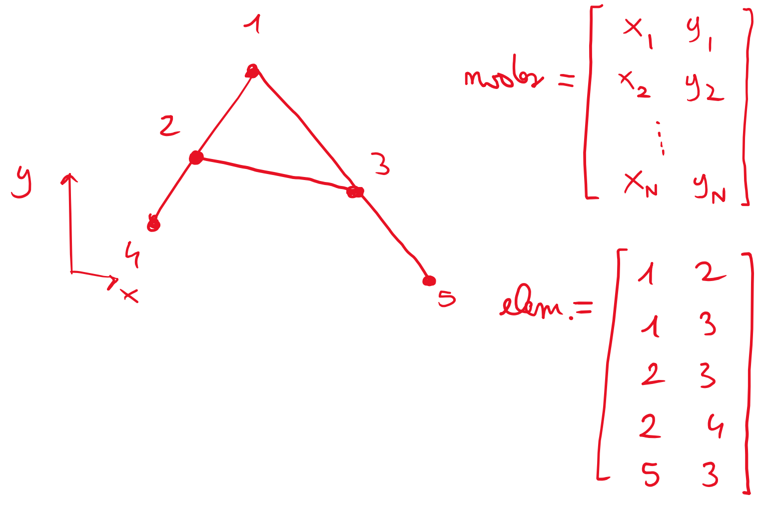
```matlab

%%
%% ******************************************************************************************************
%  ************                        GEOMETRY OF THE CROSS SECTION                        ************
%  ******************************************************************************************************
fprintf('- define the cross section...\n');

%define the nodes of the section
SECTION.nodes = [
                  0,      0;
                  B/2,    0;
                  B,      0;
                  0,      H;
                  B/2,    H;
                  B,      H;
                  ];

%define the element of the section
SECTION.elem = [
                  1,  2;
                  2,  3;
                  2,  5;
                  4,  5;
                  5,  6;
                  ];

%define the thickness of the element of the section
SECTION.thickness = [tw, tw, tf, tw, tw];

%define the closed path of the section (a cell array of vectors containing the labels of the nodes of the path)
SECTION.closedPath = {};

%define the reference frame in which the geometrical properties are computed
SECTION.reference = 'originalReferenceFrame';

```

DEFINIZIONE dei NODI del PROFILO → nella i-ESIMA RIGA LE COORDINATE del NODO i-ESIMO

DEFINIZIONE degli ELEMENTI della SEZIONE → nella j-ESIMA RIGA LE ETICHETTE dei NODI COLLEGATI

VETTORE CON LO SPESSORE degli ELEMENTI → nel posto i-ESIMO, SPESSORE dell'ELEMENTO i-ESIMO

NOME PER RICORDARSI IL SIST. di RIFERIMENTO...



3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 | Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```
%%
%% ***************************************************************************************************
%   ************                         INERTIAL PROPERTIES                            ************
%   ***************************************************************************************************
fprintf('- computing the inertial properties...\n');

%compute the length and the angle of the section
SECTION = SaintVenant.lengthElementSection(SECTION);
SECTION = SaintVenant.angleElementSection(SECTION);
SECTION = SaintVenant.simplifyFields(SECTION);

%compute the area of each thin web
SECTION = SaintVenant.computeAreaElem(SECTION);

%compute the centroids of each element of the thin section
SECTION = SaintVenant.computeCentroidElem(SECTION);

%compute the centroid of the entire cross-section
SECTION = SaintVenant.computeCentroid(SECTION);

%compute the first moment of area of each element with respect to the reference system of the SECTION
SECTION = SaintVenant.computeFirstMomentSection(SECTION);

%compute the principal second moment of area of each element with respect to the reference system of the SECTION
considerWeakAxis = false;
SECTION = SaintVenant.computeSecondMomentSection(SECTION, considerWeakAxis);
SECTION = SaintVenant.simplifyFields(SECTION);



%%
%% ***************************************************************************************************
%   ************                      CENTRAL REFERENCE FRAME                           ************
```

CALCOLO delle PROPRIETÀ GEOMETRICHE della SEZIONE

SEMPLIFICA LA SCRITTURA SIMBOLICA

SECTION è DATO STRUTTURATO che CONTIENE TUTTE LE INFORMAZIONI SULLA SEZIONE

SI PUÒ CONSIDERARE O NO IL J_m DEBOLE del PROFILO SOTTILE

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 | Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR | PUBLISH | VIEW

New | Open | Save | Compare | Print — FILE
Go To | Find | Bookmark — NAVIGATE
Refactor — CODE
Profiler | Analyze — ANALYZE
Run Section | Section Break | Run and Advance | Run to End — SECTION
Run | Step | Stop — RUN

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

QUI SI TROVA IL CALCOLO del SISTEMA CENTRALE D'INERZIA

```
%%
%% ************************************************************************************
%  ************                  CENTRAL REFERENCE FRAME                  ************
%  ************************************************************************************
fprintf('- defining the central reference frame...\n');

%translate the reference system
newRef = 'translatedRefSystemInG';
coordNewOriginX = simplify(SECTION.centroid(1));
coordNewOriginY = simplify(SECTION.centroid(2));
SECTION_2 = SaintVenant.translateReferenceSystem(SECTION, coordNewOriginX, coordNewOriginY, newRef);

%compute the angle of the principal reference frame
angle = SaintVenant.computeAnglePrincipalRef(SECTION_2);

%translate the reference system in the principal reference frame
newRef = 'rotatedRefSystemInG';
angleRot = angle;
SECTION_3 = SaintVenant.rotateReferenceSystem(SECTION_2, angleRot, newRef);

%define the final section that must be used for the analysis
SECTION_F = SECTION_3;

%%
%% ************************************************************************************
%  ************                  SIMPLIFY THE OUTPUT                  ************
%  ************************************************************************************
% SECTION
% SaintVenant.simplifyFields(SECTION)
% SaintVenant.numSimplifyFields(SECTION)
%
% SECTION_2
% SaintVenant.simplifyFields(SECTION_2)
% SaintVenant.numSimplifyFields(SECTION_2)
```

TRASLAZIONE della SEZIONE NEL SISTEMA BARICENTRALE

ANGOLO del SIST. CENTRALE

SECTION_3 è RIFERITA AL SISTEMA CENTRALE D'INERZIA

QUI SI DEFINISCE LA SEZIONE DA USARE NEI CALCOLI SUCCESSIVI

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR PUBLISH VIEW

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```matlab
%% ******************************************************************************************
% ************                          DEFINE LOADS                          ************
% ******************************************************************************************
fprintf('- defining the loads...\n');
% - - CASE 1: completely symbolic values (they are assumed to be referred to the section in the central
% reference frame)

%define the symbolic values
syms NN MMx MMy TTx TTy MMt

%define the six components of the internal forces
N = NN;
Mx = MMx;
My = MMy;
Tx = TTx;
Ty = TTy;
Mt = MMt;



% - - CASE 2: symbolic values function of parameters (they are assumed to be referred to the section
% in the central reference frame)

%define the symbolic values
syms F W

% %define the six components of the internal forces
% N = F;
% Mx = 2 * W;
% My = 3 * W;
% Tx = 4 * F;
% Ty = -3 * F;
% Mt = 1.5 * W;



% % - - CASE 3: numerical values (referred to the section in the central reference frame)
% N = 10;
% Mx = 15;
% My = 30;
% Tx = 45;
```

DEFINIZIONE dei CARICHI

4 MODI PER DEFINIRE CARICHI:

i) TOTALMENTE SIMBOLICO e GENERICO

ii) SIMBOLICO MA CON PARAMETRI DATI

iii) NUMERICO

iv) DANDO CENTRO DI PRESSIONE (TRASLAZIONE del CENTRO !!!)

N.B.: SE È NOTO che ALCUNE AZIONI INTERNE SONO NULLE, BISOGNA DEFINIRLO GIÀ QUI

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 | Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR PUBLISH VIEW

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```
%%
%% ************************************************************************************************************
%  ************                               SOLVE STATIC PROBLEM                               ************
%  ************************************************************************************************************
fprintf('- solving the static problem...\n');

%find the normal stress sigma due to N, Mx and My
[sigma, C, axisN, axisS] = SaintVenant.computeSigma(SECTION_F, N, Mx, My);
sigmaMaxMin = SaintVenant.valueSigmaMinMax(SECTION_F, sigma);

%find the tangential stresses tau due to shear forces Tx and Ty
[funSX, funSY, funIntSX, funIntSY] = SaintVenant.solveStaticMomentJourawski(SECTION_F);
tauS = SaintVenant.solveTauShearJourawski(SECTION_F, funSX, funSY, Tx, Ty);
fluxShear = SaintVenant.solveFluxShearJourawski(SECTION_F, funIntSX, funIntSY, Tx, Ty);

%find the tangential stresses tau due to torsional moment Mt
[tauT, tauMaxT, Jtelem, funtFlux, funmTors] = SaintVenant.solveTorsion(SECTION_F, Mt);

%find center of shear
centerS = vpa(SaintVenant.shearCenter(SECTION_F, funIntSX, funIntSY, Tx, Ty));

%find the value of the ideal monoaxial stress according to Tresca/Von Mises criteria
tauTot = tauS + tauT;
crit = 'VonMises';
sigmaIdVM = SaintVenant.yieldingCriteria(SECTION, sigma, tauTot, crit);
crit = 'Tresca';
sigmaIdTR = SaintVenant.yieldingCriteria(SECTION, sigma, tauTot, crit);

%display the solution
fprintf(' \t\t - sigma:\n');
sigma
% fprintf(' \t\t - value of max and min sigma:\n');
% sigmaMaxMin
fprintf(' \t\t - tau (shear):\n');
tauS
fprintf(' \t\t - tau (torsion):\n');
tauT
fprintf(' \t\t - center of pressure:\n');
C
% fprintf(' \t\t - center of shear:\n');
% centerS
```

CALCOLO delle σ_z (N, M_x, M_y)

CALCOLO delle τ (T_x, T_y)

CALCOLO delle τ (M_t)

CENTRO di TAGLIO

σ_{id} di TRESCA/VON MISES

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR PUBLISH VIEW

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```matlab
% centerS



%%
%% **************************************************************************************************************
%  ************                          COMPUTE NUMERICAL VALUES                                  ************
%  **************************************************************************************************************
fprintf('- computing the numerical values...\n');

%set the value of the parameters
listOfParameters = [B, H, tw, tf, NN, MMx, MMy, TTx, TTy, MMt];
listOfValues = [15, 30, 0.7, 1, 1, 1, 1, 2, 2, 1];
% listOfParameters = [a, t, F, W];
% listOfValues = [1, 0.1, 1, 1];

%define the section containing only numerical values and variables x, y, xi, eta (to plot the section
%the numerical values of the quantities are necessary)
SECTION_P = SaintVenant.numSimplifyFields( ...
                            SaintVenant.subsNumericalValueFields(SECTION_F, listOfParameters, listOfValues));

%numerical value of the results
C_N = vpa(subs(C, listOfParameters, listOfValues));
S_N = vpa(subs(centerS, listOfParameters, listOfValues));
axisN_N = vpa(subs(axisN, listOfParameters, listOfValues));
axisS_N = vpa(subs(axisS, listOfParameters, listOfValues));
sigma_N = simplify(vpa(subs(sigma, listOfParameters, listOfValues)));
sigmaMaxMin_N = simplify(vpa(subs(sigmaMaxMin, listOfParameters, listOfValues)));
funSX_N = simplify(vpa(subs(funSX, listOfParameters, listOfValues)));
funSY_N = simplify(vpa(subs(funSY, listOfParameters, listOfValues)));
funIntSX_N = simplify(vpa(subs(funIntSX, listOfParameters, listOfValues)));
funIntSY_N = simplify(vpa(subs(funIntSY, listOfParameters, listOfValues)));
tauS_N = simplify(vpa(subs(tauS, listOfParameters, listOfValues)));
fluxShear_N = simplify(vpa(subs(fluxShear, listOfParameters, listOfValues)));
tauT_N = simplify(vpa(subs(tauT, listOfParameters, listOfValues)));
tauMaxT_N = simplify(vpa(subs(tauMaxT, listOfParameters, listOfValues)));
funtFlux_N = simplify(vpa(subs(funtFlux, listOfParameters, listOfValues)));
funmTors_N = simplify(vpa(subs(funmTors, listOfParameters, listOfValues)));
sigmaIdVM_N = simplify(vpa(subs(sigmaIdVM, listOfParameters, listOfValues)));
sigmaIdTR_N = simplify(vpa(subs(sigmaIdTR, listOfParameters, listOfValues)));
```

PER I GRAFICI SERVONO DEI VALORI NUMERICI!! BISOGNA INSERIRLI QUA...

CALCOLO DI TUTTI GLI OUTPUT PRECEDENTI IN FORMA NUMERICA

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR PUBLISH VIEW

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```
%% ******************************************************************************************************
%   ************                                PLOTS                                      ************
%   ******************************************************************************************************
fprintf('- plotting...\n');

% plot of the cross section
graphicOptions = Graphic.libGraphicStyle('STYLE1_SECTION');
graphicOptionsPt = Graphic.libGraphicStyle('STYLE1_POINT');


figure(1)
hold on
title('Plot of the cross section');
Graphic.plotSection(SECTION_P, graphicOptions);
Graphic.plotCentroid(SECTION_P, graphicOptionsPt);
% set(gca, 'XDir', 'reverse');
% set(gca, 'YDir', 'reverse');
axis equal




%%
%plot of the results of normal stress
graphicOptions = Graphic.libGraphicStyle('STYLE1_SECTION');
graphicOptionsPt = Graphic.libGraphicStyle('STYLE1_POINT');
graphicOptionsEll = Graphic.libGraphicStyle('STYLE1_LINES');
graphicOptionsSigma = Graphic.libGraphicStyle('STYLE1_OUTSIGMA');
scaleSigma = 0.8;

figure(2)
hold on
title('Normal stress analysis');
Graphic.plotSection(SECTION_P, graphicOptions);
Graphic.plotCentroid(SECTION_P, graphicOptionsPt);
Graphic.plotEllipseOfInertia(SECTION_P, graphicOptionsEll);
Graphic.plotNormalForceOutput(SECTION_P, C_N, axisN_N, axisS_N, sigma_N, scaleSigma, graphicOptionsSigma);
% set(gca, 'XDir', 'reverse');
% set(gca, 'YDir', 'reverse');
axis equal
```

DEFINIZIONE dello STILE (già fatto in Graphic.libGraphicStyle)

→ PLOT della SEZIONE e/o del GRAFICO VOLUTO

PER VISUALIZZARE MEGLIO SI PUÒ CAMBIARE LA SCALA DEL GRAFICO

SE SERVE SI POSSONO RUOTARE GLI ASSI

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 | Col 1

Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR PUBLISH VIEW

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```matlab



%%
%plot of the results of tau due to shear - first moment of area Sx
graphicOptions = Graphic.libGraphicStyle('STYLE3_SECTION');
graphicOptionsPt = Graphic.libGraphicStyle('STYLE1_POINT');
graphicOptions2 = Graphic.libGraphicStyle('STYLE4_DIAGRAMS');
numSubs = 40;
scale = 0.02;

figure(3)
hold on
title('First moment of area (shear) Sx');
Graphic.plotDiagramsOnSection(SECTION_P, funSX_N, numSubs, scale, graphicOptions2);
Graphic.plotSection(SECTION_P, graphicOptions);
Graphic.plotCentroid(SECTION_P, graphicOptionsPt);
% set(gca, 'XDir', 'reverse');
% set(gca, 'YDir', 'reverse');
axis equal

%%
%plot of the results of tau due to shear - first moment of area Sy
graphicOptions = Graphic.libGraphicStyle('STYLE3_SECTION');
graphicOptionsPt = Graphic.libGraphicStyle('STYLE1_POINT');
graphicOptions2 = Graphic.libGraphicStyle('STYLE1_DIAGRAMS');
numSubs = 40;
scale = 0.1;

figure(4)
hold on
title('First moment of area (shear) Sy');
Graphic.plotDiagramsOnSection(SECTION_P, funSY_N, numSubs, scale, graphicOptions2);
Graphic.plotSection(SECTION_P, graphicOptions);
Graphic.plotCentroid(SECTION_P, graphicOptionsPt);
% set(gca, 'XDir', 'reverse');
% set(gca, 'YDir', 'reverse');
axis equal
```

SI POSSONO DEFINIRE PIU' STILI GRAFICI IN BASE A QUELLO CHE SI VUOLE DISEGNARE

PER I GRAFICI BISOGNA DIRE ANCHE IL NUMERO DI PUNTI IN CUI SI DIVIDE IL DOMINIO

I PLOT SONO PENSATI PER ESSERE ESEGUITI CON LA STESSA SCALA SUGLI ASSI

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 | Col 1

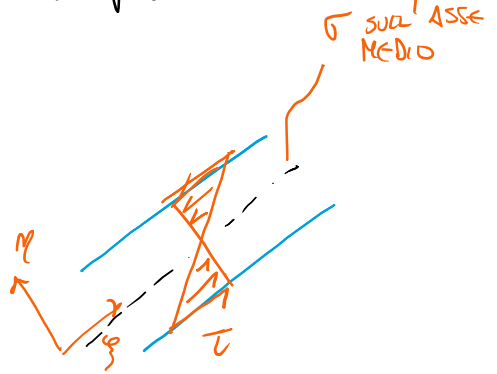
Editor - C:\Users\marot\Desktop\UniTS\Didattica\SaintVenant\script_sezIPE.m

EDITOR PUBLISH VIEW

SaintVenant.m | Graphic.m | script1.m | script_sezIPE.m

```
%%
%plot of the results of safety check - Von Mises
graphicOptions = Graphic.libGraphicStyle('STYLE3_SECTION');
graphicOptions1 = Graphic.libGraphicStyle('STYLE1_DIAGRAMS');
graphicOptions2 = Graphic.libGraphicStyle('STYLE2_DIAGRAMS');
graphicOptions3 = Graphic.libGraphicStyle('STYLE3_DIAGRAMS');
graphicOptions4 = Graphic.libGraphicStyle('STYLE5_DIAGRAMS');
graphicOptions1.faceAlpha = 0.4;
graphicOptions2.faceAlpha = 0.4;
graphicOptions3.faceAlpha = 0.4;
graphicOptions4.faceAlpha = 0.3;
numSubs = 40;
scale = 20;

%compute the value of the ideal sigma and define the value of fy
matSigmaIdVM = SaintVenant.computeSigmaIdThickness(SECTION_P, sigmaIdVM_N);
fyeild = 0.2;
fyeildVec = (fyeild + (0 * xi)) * ones(length(matSigmaIdVM(1, :)));

figure(8)
hold on
title('distribution of the ideal stress \sigma_{id} (Von Mises yielding criterion)');
Graphic.plotDiagramsOnSection(SECTION_P, matSigmaIdVM(1, :), numSubs, scale, graphicOptions1);
Graphic.plotDiagramsOnSection(SECTION_P, matSigmaIdVM(2, :), numSubs, scale, graphicOptions2);
Graphic.plotDiagramsOnSection(SECTION_P, matSigmaIdVM(3, :), numSubs, scale, graphicOptions3);
Graphic.plotDiagramsOnSection(SECTION_P, fyeildVec, numSubs, scale, graphicOptions4);
Graphic.plotSection(SECTION_P, graphicOptions);
axis equal
```

$$\sigma_{id} = \sqrt{\sigma^2 + 3\tau^2}$$



QUANDO C'È TORSIONE, NON È FACILE CAPIRE DOVE SI TROVA LA $\sigma_{id,MAX}$

→ SI CALCOLA NEL PUNTO CENTRALE E NEGLI ESTREMI dello SPESSORE...

PLOT ANCHE della TENSIONE DI SNERVAMENTO $\sigma_{id} \leq f_y$

3 usages of "funSX" found | Zoom: 125% | UTF-8 | CRLF | script | Ln 1 Col 1

— ESEMPI SEZIONI —