

ANNEXE 2. CODI FONT

INDEX

<i>cme2.m</i>	1
<i>cme2_calcul_estatic.m</i>	36
<i>dibuix_diagrames.m</i>	45
<i>dlgnus.m</i>	54
<i>dlgbarra.m</i>	58
<i>dlgcarreganus.m</i>	61
<i>dlgcarregabarra.m</i>	63
<i>dlgopcions.m</i>	65
<i>dibuix_contorn.m</i>	69
<i>dibuix_carreganus.m</i>	71
<i>dibuix_carregabarra.m</i>	74
<i>dibuix_projectada.m</i>	77
<i>dibuix_repartida.m</i>	79
<i>nus.m</i>	81
<i>barra.m</i>	82
<i>diagrama.m</i>	83

```

% CÀLCUL MATRICIAL D'ESTRUCTURES 2.0 "AKA CME2"
%
% PER SERGIO GUERRERO MIRALLES
%
% TREBALL DE FINAL DE CARRERA D'ENGINYERIA TÈCNICA D'OBRES PÚBLIQUES
% UNIVERSITAT POLITÈCNICA DE CATALUNYA, CURS 2011-2012
%
%
% Creació, edició, càlcul estàtic i mostra de resultats amb interfase
% gràfica d'estructures de barres articulades i reticulades, 2D i 3D, amb
% carregues a les barres.
%
% Darrera versió 13 d'octubre de 2.012.
%
function cme2()
%% FUNCIO QUE S'EXECUTA A L'INICI, DEFINICIO DE L'ENTORN GRAFIC:

% Inicialitza variables del programa i carrega les opcions.
[nusos,contnus,barres,diagrames,hgreaccions,contbarra,vista,flagbarra,primernusbarra,artprimernusbarra,artnusan,
hbarra,flagclic,flagmourenus,posini,flagcalculat,s,xmin,xmax,ymin,ymax,zmin,zmax,xm,ym,zm,xmed,ymed,zmed,angle3d,grafica,sD,mov_extrems,TL2G,Retorn_reaccions,Retorn_moviments,Retorn_esforcos_barres,diag_presxy,diag_presxz,nom,fmax,qmax]=inicialitza_variables;
[editalcrear,barraalcrearnus,fontnusos,tamanyfontnusos,fontcarregues,tamanyfontcarregues,factorzoom,tamanynus,distnumnus,precini,XLimini,YLimini,ZLimini,colormodificar,coloresborrar,colorcarregues,colorreaccions,colordeformada,colorN,colorTy,colorTz,colorMx,colorMy,colorMz,direct]=loadopcions; % Carrega les opcions.

[figcme2,tm,bg]=creacio_finestra; % Crea la finestra principal del programa.
[barraeines,botobarracrear,botobarramodificar,botobarraesborrar,botobarranus,botobarrabarra,botobarracarreganus,botobarracarregabarra,botobarrarotula,botobarramoure,botobarraapropar,botobarraallunyar]=creacio_barra_eines; % Crea la barra d'eines.
creacio_menus % Crea la barra de menús.
[panell,logo,grafica_resxy,grafica_resxz,txtbox_res,grafica_est,editx,edity,editz,editprec]=creacio_panell_grafica; % Crea el panell amb les grafiques i les caixes amb les coordenades i la precisió.
[panel2,botoedicio,botoxy,botoxz,botoyz,boto3d]=panell_botons_vista; % Crea el panel amb els botons que controlen les vistes i els modes d'edició/presentació de resultats.
[panel3,botocarregues,botoreaccions,botodeformada,botoaxils,bototallantY,bototallantZ,botomomentX,botomomentY,botomomentZ]=panell_botons_grafica; % Panell amb els botons per seleccionar els elements que es mostrin en la presentació de resultats.
botocalcular1=boto_calcular; % Botó per executar el càlcul.

% Un cop creats els objectes, el programa espera la interacció amb l'usuari
% i executa les accions programades als diferents objectes.

%% CREACIO I ACCIONS SOBRE LA FINESTRA PRINCIPAL, REDIMENSIONAMENT, CONTROL DEL RATOLÍ I TECLAT, ETC:

function [nusos,contnus,barres,diagrames,hgreaccions,contbarra,vista,flagbarra,primernusbarra,artprimernusbarra,artnusan,hbarra,flagclic,flagmourenus,posini,flagcalculat,s,xmin,xmax,ymin,ymax,zmin,zmax,xm,ym,zm,xmed,ymed,zmed,angle3d,grafica,sD,mov_extrems,TL2G,Retorn_reaccions,Retorn_moviments,Retorn_esforcos_barres,diag_presxy,diag_presxz,nom,fmax,qmax]=inicialitza_variables
% Funció que defineix les variables globals dels programa amb els
% valors inicials. Aquestes variables són accessibles dins de l'espai de
% treball de la funció principal del programa i les seves sub-funcions.
% Apareixen de color blau a l'editor de Matlab.

nusos=nus; % Llista de nusos.
contnus=0; % Comptador de nusos.
barres=barra; % Llista de barres.
contbarra=0; % Comptador de barres.
diagrames=diagrama; % Llista de diagrames (1 objecte diagrama per a cada barra).
hgreaccions=[]; % Llista d'objectes gràfics de reaccions.
vista='xy'; % Mode de vista actual.
flagbarra=0; % Flag que controla el proces de crear barres. S'activa quan s'ha clicat el primer nus a l'espera del segon.
primernusbarra=0; % Primer nus clicat en el proces de crear barres.
artprimernusbarra=0; % Estat del botó articulació en el moment de clicar el primer nus d'una barra.
artnusan=0; % Estat del botó articulació en el moment de crear l'ultim nus creat.
hbarra=0; % Línia temporal que representa la barra en construcció.
flagclic=0; % Flag que indica si s'ha clicat un boto del ratoli i quin ha estat.
flagmourenus=0; % Falg que indica a la funcio que detecta moviment del nus que s'ha clicat un nus amb el botó dret i s'ha de desplaçar.
posini=zeros(2,6); % Posicio inicial del ratoli al moure la pantalla.
flagcalculat=0; % Flag que indica que hi ha una estructura calculada.
s=0; % Factor d'escala per als objectes gràfics que es mostren en pantalla: nusos, carregues, etc.

xmin=[]; % Coordenada x minima de l'estructura.
xmax=[]; % Coordenada x maxima de l'estructura.
ymin=[]; % Coordenada y minima de l'estructura.
ymax=[]; % Coordenada y maxima de l'estructura.

```

```

zmin=[]; % Coordenada z minima de l'estructura.
zmax=[]; % Coordenada z maxima de l'estructura.
xmed=[]; % Mitjana de les coordenades x mínima i màxima de l'estructura.
ymed=[]; % Mitjana de les coordenades y mínima i màxima de l'estructura.
zmed=[]; % Mitjana de les coordenades z mínima i màxima de l'estructura.
angle3d=0; % Angle de rotació a la vista 3d.
xm=0; % Mitjana de les coordenades x de l'estructura.
ym=0; % Mitjana de les coordenades y de l'estructura.
zm=0; % Mitjana de les coordenades z de l'estructura.
grafica=0; % grafica que s'esta mostrant, grafica de l'estructura o grafiques de resultats
detallats per barra.
SD=1; % Escala de la deformada.
mov_extrems=[]; % Moviments en extrems de barra en eixos locals.
TL2G=[]; % Matrius de canvi de base de les barres.
Retorn_reaccions=[]; % Reaccions en eixos globals.
Retorn_moviments=[]; % Moviments en eixos locals.
Retorn_esforços_barres=[]; % Esforços en extrems de barra en eixos locals.
diag_presxy=diagrama; % Objecte diagrama que conte els diagrames detallats per barra en xy.
diag_presxz=diagrama; % Objecte diagrama que conte els diagrames detallats per barra en xz.
nom=''; % Nom de l'estructura actual
fmax=0; % Força aplicada als nusos o puntual maxima.
qmax=0; % Càrrega repartida o projectada maxima.
end

function [figcme2,tm,bg]=creacio_finestra
% Creació de la finestra:
set(0,'Units','pixels') % Per obtenir el tamany de la pantalla en pixels i no en cap altre
messura extraña.
sc=get(0,'ScreenSize'); % Obté el tamany de la pantalla.
tm=[1272 618]; % Tamany de la finestra a la pantalla.
ps=(sc(3:4)-tm)/2-[0,18]; % Posicio de la finestra, centrada a la pantalla.
bg=get(0,'defaultUicontrolBackgroundColor'); % Obté el color de fons per defecte.
figcme2=figure('Units','pixels','Position',[ps,tm],'NumberTitle','off','Name','Càlcul Matricial
d'Estructures 2.0','Resize','on','color',bg,'ToolBar','figure','MenuBar','none','ResizeFcn',
@figcme2_resize,'WindowButtonMotionFcn',@moviment_ratoli,'WindowButtonDownFcn',
@clic_figure,'WindowButtonUpFcn',@unclic_figure,'WindowScrollWheelFcn',@roda_ratoli,'CloseRequestFcn',
@sortir,'KeyPressFcn',@tecla); % Crea la finestra principal del programa.
end

function figcme2_rename
if nom
set(figcme2,'Name',['Càlcul Matricial d'Estructures 2.0 - ',nom])
else
set(figcme2,'Name','Càlcul Matricial d'Estructures 2.0')
end
end

function figcme2_resize(hObject,eventdata,handles)
% Recoloca els panells quan es redimensiona la finestra principal del
% programa.
sc=get(figcme2,'Position'); % Posicio i tamany de la pantalla despres de redimensionar.
tm=sc(3:4); % Tamany de la finestra a la pantalla.
if tm(1)<500 % Limita el mínim horitzontal a 500 píxels.
tm(1)=500;
set(figcme2,'Position',[sc(1:2),tm]);
end
if tm(2)<425 % Limita el mínim Vertical a 425 píxels.
sc(2)=sc(2)+tm(2)-425;
tm(2)=425;
set(figcme2,'Position',[sc(1:2),tm]);
end
% Canvia posició i tamany dels panells i elements.
set(panel1,'Position',[5,5,tm-[130,10]]); % Panell de la grafica.
set(panel2,'Position',[tm-[120,190],115,195]); % Panell dels botons de les vistes.
set(panel3,'Position',[tm-[120,390],115,200]); % Panell dels botons de la gràfica.
set(botocalcular1,'Position',[tm(1)-120,5,115,25]); % Boto calcular.
set(grafica_est,'Position',[36,50,tm-[176,70]]);
set(logo,'Position',[36,50,tm-[176,70]]);
set(grafica_resxy,'Position',[36,(tm(2)+30)/2,(tm-[182,70])/2]);
set(grafica_resxz,'Position',[36,50,(tm-[182,70])/2]);
set(txtbox_res,'Position',[tm(1)/2-49,50,tm(1)/2-87,tm(2)-70]);
centrar
end

function clic_figure(hObject,eventdata,handles)
% S'executa sempre que es premi el botó del ratoli.
% No confondre amb la funció clic_ratoli ni altres, clic_figure
% s'executa sempre abans i es cliqueji sobre la grafica o qualsevol
% altre objecte, és més genèrica i complementa les altres funcions.
% Discrimina segons el tipus de pulsació i l'enmagatzema a la
% variable flagclic de manera que al pitjar sobre un element gràfic
% es puguin programar diferents efectes.

```

```

switch get(figcme2, 'SelectionType')
    case 'normal' % Botó esquerre
        flagclic=1;
        if strcmp(get(botobarramoure, 'State'), 'on')
            posini=get(grafica_est, 'currentpoint');
        end
    case 'open' % Doble clic
        flagclic=2;
    case 'alt'
        flagclic=3; % Control i botó esquerre o botó dret
    case 'extend'
        flagclic=4; % Shift i botó esquerre, botó central o ambdós botons
        posini=get(grafica_est, 'currentpoint');
        set(figcme2, 'Pointer', 'hand');
end
end

function uncllic_figure(hObject,eventdata,handles)
    % S'executa a l'alliberar el botó del ratolí.

    flagclic=0; % Restitueix la variable que defineix el tipus de clic.
    flagmourenus=0; % Si s'estava movent un nus, restitueix el flag perquè no es mogui el nus amb
el moviment del ratolí.

    % Per si estava movent la pantalla (cursor amb forma de ma) torna
    % al cursor de fletxa o al que correspongui.
    if strcmp(get(botobarramoure, 'State'), 'on')
        set(figcme2, 'Pointer', 'hand');
    elseif strcmp(get(botobarraapropar, 'State'), 'on') || strcmp(get
(botobarraallunyar, 'State'), 'on')
        set(figcme2, 'Pointer', 'circle');
    else
        set(figcme2, 'Pointer', 'arrow');
    end
end

function roda_ratoli(hObject,eventdata)
    % Amplia i redueix la gràfica al fer rodar la rodeta del ratolí.

    zom(eventdata.VerticalScrollCount/factorzoom)
end

function [] = zom(factor)
    % Amplia i redueix la gràfica.

    %disp('Zoom')
    %tic_zom=tic;
    if grafica==grafica_est
        lim=[get(grafica_est, 'XLim');get(grafica_est, 'YLim');get(grafica_est, 'ZLim')]; % Llegeix
l'escala dels eixos.
        med=(lim(:,2)+lim(:,1))/2; % Punt mig.
        an=(lim(:,2)-lim(:,1))*2^factor; % Amplada multiplicada per 2 elevat al factor d'ampliació.
        lim=[med-an/2, med+an/2]; % Nova escala dels eixos.
        set(grafica_est, 'XLim', lim(1,:), 'YLim', lim(2,:), 'ZLim', lim(3,:)); % Fixa la nova escala.
        escala % Calcula el factor d'escala s.
        escalar_nusos % Reescala els nusos i les seves carregues.
        redibuixar_carregues_barres % Redibuixa les càrregues sobre les barres amb la nova escala.
        clicable % Estableix quins objectes responen al ratolí segons els botons de la barra.
    end
    %disp([' /Zoom: ', num2str(toc(tic_zom))])
end

function moviment_ratoli(hObject,eventdata,handles)
    % S'executa quan es mou el ratolí sobre la finestra de l'aplicació.
    % Actualitza els valors de les caixes de text que mostren la posició
    % del ratolí.
    % Al crear una barra crea l'efecte de que la línia segueix el cursor.
    % Si es prem la rodeta permet moure la gràfica.
    % Si es prem el botó dret mou nusos.

    posbruta=get(grafica_est, 'currentpoint'); % Posició del cursor.
    precisió=str2double(get(editprec, 'String')); % Precisió definida a la seva caixa de text.
    posició=round(posbruta(1,1:3)/precisió)*precisió; % Redondeja la posició a la precisió
establerta.

    % Actualitza la posició a les caixes de text.
    switch vista
        case 'xy'
            set(editx, 'String', num2str(posició(1))); % A la vista al plànol xy actualitza posicions
x i y.
            set(edity, 'String', num2str(posició(2)));

```

```

        posicio(3)=str2double(get(editz, 'String'));
    case 'xz'
        set(editx, 'String', num2str(posicio(1))); % A la vista al plànol xz actualitza posicions
x i z.
        posicio(2)=str2double(get(edity, 'String'));
        set(editz, 'String', num2str(posicio(3)));
    case 'yz'
        posicio(1)=str2double(get(editx, 'String')); % A la vista al plànol yz actualitza
posicions y i z.
        set(edity, 'String', num2str(posicio(2)));
        set(editz, 'String', num2str(posicio(3)));
    case '3d'
        set(editx, 'String', '');
        set(edity, 'String', '');
        set(editz, 'String', '');
    end

    % Si s'està creant una bara actualitza les coordenades amb la posició del cursor.
    if flagbarra
        x=get(hbarra, 'XData');
        y=get(hbarra, 'YData');
        z=get(hbarra, 'ZData');
        set(hbarra, 'XData', [x(1) posbruta(1,1)], 'YData', [y(1) posbruta(1,2)], 'ZData', [z(1) posbruta
(1,3)]);
    end

    % Si s'ha pres la rodeta del ratolí o el botó moure modifica els
    % límits de la gràfica creant l'efecte de moviment.
    if flagclic ==4 || (flagclic ==1 && strcmp(get(botobarramoure, 'State'), 'on'))
        mov=posbruta-posini; % posini es la posició anterior.
        mov=mov(1, :);
        if ~strcmp(vista, '3d')
            xl=get(grafica_est, 'XLim');
            xl=xl-[mov(1), mov(1)];
            yl=get(grafica_est, 'YLim')-[mov(2), mov(2)];
            zl=get(grafica_est, 'ZLim')-[mov(3), mov(3)];
            set(grafica_est, 'XLim', xl, 'YLim', yl, 'ZLim', zl);
        end
    end

    % Amb el botó dret pres sobre un nus permet moure'l sobre la
    % gràfica.
    if flagmourenus && ~get(boto3d, 'Value')
        nusos(flagmourenus).posicio=posicio; % flagmourenus conte el handle del nus.
        escala % Calcula el factor d'escala s.
        redibuixar_nusos % Redibuixa els nusos per tal de que es reorientin segons els nous
extrems.
        redibuixar_barres % Actualitza el dibuix de les barres.
        escalar_nusos % Reescala els nusos i les seves carregues.
        redibuixar_carregues_barres % Redibuixa les càrregues sobre les barres amb la nova escala.
        clicable % Estableix quins objectes responen al ratolí segons els botons de la barra.
    end
end

function tecla(hObject, eventdata, handles)
    % S'executa cada cop que es pitja una tecla.
    % Permet diferents accions en diferents moments.

    % Si s'està creant una barra permet abortar prement escapament.
    if flagbarra && strcmp(eventdata.Key, 'escape')
        flagbarra=0; % Posa a 0 el flag de que s'està creant barra.
        delete(hbarra) % Esborra la línia.
        hbarra=[];
    end

    % Amplia o redueix l'escala de la deformada prement '+' o '-'.
    if ~get(botoedicio, 'Value') && get(botodeformada, 'Value') && flagcalculat
        if strcmp(eventdata.Character, '-')
            sD=sD/2^(1/4); % Divideix l'escala per l'arrel quarta de 2, amb 4 pulsacions es
divideix l'escala per 2.
        elseif strcmp(eventdata.Character, '+')
            sD=sD*2^(1/4); % Multiplica l'escala per l'arrel quarta de 2, amb 4 pulsacions es
duplica l'escala.
        end
        for i=1:numel(diagrames)
            escala_deformada(TL2G(:, :, i), i, sD); % Actualitza la deformada a totes les barres.
        end
    end

    % Si es pitja return en qualsevol moment equival a pitjar el botó
    % calcular.
    if strcmp(eventdata.Key, 'return')

```

```

        calcular
    end

    % Moviment dels cursor al mode de vista 3d per rotar, apropar i
    % allunyar l'estructura.
    if strcmp(vista,'3d')
        switch eventdata.Key
            case 'leftarrow' % Si es prem la tecla de fletxa esquerra.
                mov=-pi/36; % Gira l'estructura 5°.
                angle3d=angle3d+mov; % angle de rotació.
                ct=[xm,ym,zm]; % Punt 'camera target' centre de l'estructura.
                set(grafica_est,'CameraPosition',ct+[cos(angle3d),1,sin(angle3d)]); % Vista
isomètrica.
            case 'rightarrow' % Si es prem la tecla de fletxa dreta.
                mov=pi/36; % Gira l'estructura 5°.
                angle3d=angle3d+mov; % angle de rotació.
                ct=[xm,ym,zm]; % Punt 'camera target' centre de l'estructura.
                set(grafica_est,'CameraPosition',ct+[cos(angle3d),1,sin(angle3d)]); % Vista
isomètrica.
            case 'uparrow' % Si es prem la tecla de fletxa a dalt.
                zom(-3/factorzoom); % Apropra l'estructura.
            case 'downarrow' % Si es prem la tecla de fletxa a baix.
                zom(3/factorzoom); % Allunya l'estructura.
        end
    end

end

end

end

%% CREACIÓ I ACCIONS DE LA BARRA D'EINES I ELS SEUS BOTONS:

function [barraeines,botobarracrear,botobarramodificar,botobarraesborrar,botobarranus,
botobarrabarra,botobarracarreganus,botobarracarregabarra,botobarrarotula,botobarramoure,
botobarraapropar,botobarraallunyar]=creacio_barra_eines
    % Crea una nova barra d'eines i defineix els botons.
    % Per als botons de zoom i moviment en copia l'imatge de la barra
    % per defecte de Matlab, però només la imatge, les accions estan
    % reprogramades per tal de tenir el control i que Matlab no faci
    % accions extrañes.

    barraperdefecte=findall(figcme2,'Type','uitoolbar'); % Handle de la barra d'eines per defecte
de Matlab, es manté activa mentre es copien alguns botons.
    barraeines=uitoolbar; % Creació d'una barra d'eines pròpia.
    botobarracrear=uitoggletool('TooltipString','Crear','CData',imread('lapiz.
png','BackgroundColor',bg),'State','on','ClickedCallback',@botocrear); % Defineix el botó crear.
    botobarramodificar=uitoggletool('TooltipString','Modificar','Cdata',imread('editar.
png','BackgroundColor',bg),'ClickedCallback',@botomodificar); % Defineix el botó modificar.
    botobarraesborrar=uitoggletool('TooltipString','Esborrar','CData',imread('goma.
png','BackgroundColor',bg),'ClickedCallback',@botoesborrar); % Defineix el botó esborrar.
    botobarranus=uitoggletool('Separator','on','TooltipString','Nus','CData',imread('nudo.
png','BackgroundColor',bg),'State','on','ClickedCallback',@botonus); % Defineix el botó nus.
    botobarrabarra=uitoggletool('TooltipString','Barra','CData',imread('barra.
png','BackgroundColor',bg),'ClickedCallback',@botobarra); % Defineix el botó barres.
    botobarracarreganus=uitoggletool('TooltipString','Càrrega en nus','CData',imread('carganudo.
png','BackgroundColor',bg),'ClickedCallback',@botocarreganus); % Defineix el botó carrega als nusos.
    botobarracarregabarra=uitoggletool('TooltipString','Càrrega en barra','CData',imread
('cargabarra.png','BackgroundColor',bg),'ClickedCallback',@botocarregabarra); % Defineix el botó
càrrega a les barres.
    botobarrarotula=uitoggletool('Separator','on','TooltipString','Extrems de barra
articulats','CData',imread('rotula.png','BackgroundColor',bg)); % Extrem de barra articulats.
    botobarramoure=uitoggletool('Separator','on','TooltipString','Moure','s','CData',get(findall
(barraperdefecte,'TooltipString','Pan'),'CData'),'ClickedCallback',@botomoure); % Defineix el botó
moure copiant l'aspecte de la barra d'eines original.
    botobarraapropar=uitoggletool('TooltipString','Apropar-se','CData',get(findall
(barraperdefecte,'TooltipString','Zoom In'),'CData'),'ClickedCallback',@botoapropar); % Defineix el
botó apropar copiant l'aspecte de la barra d'eines original.
    botobarraallunyar=uitoggletool('TooltipString','Allunyar-se','CData',get(findall
(barraperdefecte,'TooltipString','Zoom Out'),'CData'),'ClickedCallback',@botoallunyar); % Defineix el
botó allunyar copiant l'aspecte de la barra d'eines original.
    set(figcme2,'Toolbar','none'); % Es desactiva la barra d'eines original.
end

function botocrear(hObject,eventdata,handles)
    % S'executa quan s'activa el botó crear.
    %disp('boto crear')
    botons_accions('on','off','off','off','off','off')
end
function botomodificar(hObject,eventdata,handles)
    % S'executa quan s'activa el botó modificar.
    %disp('boto modificar')
    botons_accions('off','on','off','off','off','off')
end
function botoesborrar(hObject,eventdata,handles)

```

```

    % S'executa quan s'activa el botó esborrar.
    %disp('boto esborrar')
    botons_accions('off','off','on','off','off','off')
end
function botomoure(hObject,eventdata,handles)
    % S'executa quan s'activa el botó moure.
    botons_accions('off','off','off','on','off','off')
end
function botoapropar(hObject,eventdata,handles)
    % S'executa quan s'activa el botó zoom apropar.
    botons_accions('off','off','off','off','on','off')
end
function botoallunyar(hObject,eventdata,handles)
    % S'executa quan s'activa el botó zoom allunyar.
    botons_accions('off','off','off','off','off','on')
end
function botonus(hObject,eventdata,handles)
    % S'executa quan s'activa el botó nus.
    botons_elements('on','off','off','off')
end
function botobarra(hObject,eventdata,handles)
    % S'executa quan s'activa el botó barra.
    botons_elements('off','on','off','off')
end
function botocarreganus(hObject,eventdata,handles)
    % S'executa quan s'activa el botó càrrega sobre nus.
    botons_elements('off','off','on','off')
end
function botocarregabarra(hObject,eventdata,handles)
    % S'executa quan s'activa el botó càrrega sobre barra.
    botons_elements('off','off','off','on')
end
function botons_accions(bcrear,bmodificar,besborrar,bmoure,bapropar,ballunyar)
    % Es desactiven la resta de botons del grup.
    set(botobarracrear,'State',bcrear);
    set(botobarramodificar,'State',bmodificar);
    set(botobarraesborrar,'State',besborrar);
    set(botobarramoure,'State',bmoure);
    set(botobarraapropar,'State',bapropar);
    set(botobarraallunyar,'State',ballunyar);
    colors
    clicable % Modifica colors dels elements gràfics i hi defineix si es poden cliquejar, segons el
cas.
end
function botons_elements(bnus,bbarra,bcarreganus,bcarregabarra)
    % Esdesactiven la resta de botons del grup.
    set(botobarranus,'State',bnus);
    set(botobarrabarra,'State',bbarra);
    set(botobarracarreganus,'State',bcarreganus);
    set(botobarracarregabarra,'State',bcarregabarra)
    colors
    clicable % Modifica colors dels elements gràfics i hi defineix si es poden cliquejar, segons el
cas.
end
function clicable(hObject,eventdata,handles)
    % Modifica l'estat dels objectes gràfics que representen nusos,
    % barres i càrregues, perquè acceptin clic de ratolí o no, de
    % manera que els que no s'hagin de clicar no interfereixin
    % entorpint el programa.

    %disp('Clicable')
    %tic_clicable=tic;
    if get(botoedicio,'Value')
        if strcmp(get(botobarracrear,'State'),'on') % Si s'esta creant...
            if strcmp(get(botobarranus,'State'),'on') % nusos, es cliqueja directament sobre la
gràfica, es desactiven els clics als objectes gràfics.
                canviestat('off','off','off','off')
            elseif strcmp(get(botobarrabarra,'State'),'on') % barres, s'activen els clics als
nusos.
                canviestat('on','off','off','off')
            elseif strcmp(get(botobarracarreganus,'State'),'on') % càrregues en nusos, s'activen
els clics als nusos.
                canviestat('on','off','off','off')
            elseif strcmp(get(botobarracarregabarra,'State'),'on') % càrregues a les barres,
s'activen clics a les barres.
                canviestat('off','on','off','off')
            end
            elseif strcmp(get(botobarramodificar,'State'),'on') || strcmp(get
(botobarraesborrar,'State'),'on') % Si s'està modificant o esborrant...
                if strcmp(get(botobarranus,'State'),'on') % nusos, s'activen els clics als nusos.
                    canviestat('on','off','off','off')
                elseif strcmp(get(botobarrabarra,'State'),'on') % barres, s'activen els clics a les

```



```

barres.
    canviestat('off','on','off','off')
    elseif strcmp(get(botobarracarreganus,'State'),'on') % càrregues en nusos, s'activen
els clics a les càrregues als nusos.
    canviestat('off','off','on','off')
    elseif strcmp(get(botobarracarregabarra,'State'),'on') % càrregues a les barres,
s'activen els clics a les càrregues a les barres.
    canviestat('off','off','off','on')
    end
    else
    canviestat('on','off','off','off') % Per defecte, que es pugui cliquejar als nusos per
moure'ls.
    end
    else
    canviestat('off','on','off','on') % S'activen clics a les barres.
    end
    if strcmp(get(botobarramoure,'State'),'on') % Canvia el cursor.
    set(figcme2,'Pointer','hand')
    elseif strcmp(get(botobarraapropar,'State'),'on') || strcmp(get
(botobarraallunyar,'State'),'on')
    set(figcme2,'Pointer','circle')
    else
    set(figcme2,'Pointer','arrow')
    end
    %colors % Canvia els colors dels elements.
    %disp(['/Clicable: ',num2str(toc(tic_clicable))])
end

function canviestat(clicnus,clicbarra,cliccarreganus,cliccarregabarra)
% Canvia l'estat de si es poden clicar o no els objectes gràfics.
for j=1:contnus
    set(nusos(j).hg,'HitTest',clicnus) % Objectes que representen els nusos.
    set(nusos(j).hgc,'HitTest',cliccarreganus) % Objectes que representen les càrregues als
nusos.
end
for j=1:contbarra
    set(barres(j).hg,'HitTest',clicbarra) % Objectes línies que representen les barres.
    set(barres(j).hgpuntual,'HitTest',cliccarregabarra); % Objectes que representen les
càrregues puntuals sobre les barres.
    set(barres(j).hgmoment,'HitTest',cliccarregabarra); % Objectes que representen les
càrregues moment sobre les barres.
    set(barres(j).hgpartida,'HitTest',cliccarregabarra); % Objectes que representen les
càrregues repartides sobre les barres.
    set(barres(j).hgprojectada,'HitTest',cliccarregabarra); % Objectes que representen les
càrregues projectades sobre les barres.
end
end

function colornusos(hObject,eventdata,handles)
% Actualitza el color dels nusos segons la situació.

%disp('Color nusos')
%tic_cn=tic;
if strcmp(get(botobarranus,'State'),'on') && get(botoedicio,'Value') % Si esta activat el botó
nusos i el mode edició.
    if strcmp(get(botobarramodificar,'State'),'on') % Si està activat el botó modificar.
    color=colormodificar; % Assigna als nusos amb el color definit.
    elseif strcmp(get(botobarraesborrar,'State'),'on') % Si està activat el botó esborrar.
    color=coloresborrar; % Assigna als nusos amb el color definit.
    else
    color='k'; % Color d'edició per defecte negre.
    end
else
    color='k'; % Color presentacio per defecte negre.
end
for i=1:contnus
    chgcolor(nusos(i).hg,color) % Canvia el color.
end
%disp(['/Color nusos: ',num2str(toc(tic_cn))])
end

function colorbarres(hObject,eventdata,handles)
% Actualitza el color de les barres segons la situació.

%disp('Color barres')
%tic_cb=tic;
if strcmp(get(botobarrabarra,'State'),'on') && get(botoedicio,'Value') % Si esta activat el
botó barres i el mode edició.
    if strcmp(get(botobarramodificar,'State'),'on') % Si està activat el botó modificar.
    color=colormodificar; % Assigna a les barres el color definit.
    elseif strcmp(get(botobarraesborrar,'State'),'on') % Si està activat el botó esborrar.
    color=coloresborrar; % Assigna a les barres el color definit.

```



```

        else
            color='k'; % Color per defecte al mode d'edicio.
        end
    else
        color='k'; % Color per defecte al mode de presentació.
    end
    if strcmp(color,'k') % Si s'ha assignat el color per defecte, el substitueix pel color definit
a les propietats de la barra.
        for i=1:contbarra
            set(barres(i).hg,'color',barres(i).color) % Canvia el color.
        end
    else
        for i=1:contbarra
            set(barres(i).hg,'color',color) % Canvia el color pel negre.
        end
    end
    %disp(['/Color barres: ',num2str(toc(tic_cb))])
end

function colorcarreganusos(hObject,eventdata,handles)
% Actualitza el color de les càrregues sobre els nusos segons la
% situació.

%disp('Color carrega nusos')
%tic_cn=tic;
if strcmp(get(botobarracarreganus,'State'),'on') && get(botoedicio,'Value') % Si està activat
el botó de càrregues als nusos i el mode edició.
    if strcmp(get(botobarramodificar,'State'),'on') % Si està activat el botó modificar.
        color=colormodificar; % Assigna a les càrregues el color definit.
    elseif strcmp(get(botobarraesborrar,'State'),'on') % Si està activat el botó esborrar.
        color=coloresborrar; % Assigna a les càrregues el botó definit.
    else
        color='k'; % Color d'edicio per defecte negre.
    end
elseif get(botoedicio,'Value')
    color='k'; % Color d'edicio per defecte negre.
else
    color=colorcarregues; % Color presentacio definit.
end
for i=1:contnus
    chgcolor(nusos(i).hgc,color) % Canvia el color.
end
%disp(['/Color carrega nusos: ',num2str(toc(tic_cn))])
end

function colorcarregabarres(hObject,eventdata,handles)
% Actualitza el color de les càrregues sobre les barres segons la
% situació.

%disp('Color carrega barres')
%tic_cb=tic;
if strcmp(get(botobarracarregabarra,'State'),'on') && get(botoedicio,'Value') % Si està activat
el botó de càrregues a les barres i el mode edició.
    if strcmp(get(botobarramodificar,'State'),'on') % Si està activat el botó modificar.
        color=colormodificar; % Assigna a les càrregues el color definit.
    elseif strcmp(get(botobarraesborrar,'State'),'on') % Si està activat el botó esborrar.
        color=coloresborrar; % Assigna a les càrregues el botó definit.
    else
        color='k'; % Color d'edicio per defecte negre.
    end
elseif get(botoedicio,'Value')
    color='k'; % Color d'edicio per defecte negre.
else
    color=colorcarregues; % Color presentacio definit.
end
for i=1:contbarra
    chgcolor(barres(i).hgpuntual,color) % Canvia el color de les càrregues puntuals.
    chgcolor(barres(i).hgmoment,color) % Canvia el color dels moments.
    chgcolor(barres(i).hgpartida,color) % Canvia el color de les càrregues repartides.
    chgcolor(barres(i).hgprojectada,color) % Canvia el color de les càrregues projectades.
end
%disp(['/Color carrega barres: ',num2str(toc(tic_cb))])
end

function colors(hObject,eventdata,handles)
% Actualitza color de nusos, barres i càrregues.

%disp('Colors')
%tic_col=tic;
colornusos
colorbarres
colorcarreganusos

```

```

    colorcarregarbarres
    %disp(['Colors: ',num2str(toc(tic_col))])
end

function chgcolor(h,color)
% Funció que canvia els colors dels objectes continguts en un
% objecte hgtransform.

set(findobj(h,'Type','line'),'color',color) % Color de les línies.
set(findobj(h,'Type','text'),'color',color) % Color dels textos.
set(findobj(h,'Type','patch'),'FaceColor',color,'EdgeColor',color) % Color dels patch.
end

%% CREACIÓ I ACCIONS DE LA BARRA DE MENÚS:

function creacio_menus
% Crea la barra de menús.

menuarxiu=uimenu(figcme2,'Label','Arxiu'); % Menú principal 'arxiu'.
uimenu(menuarxiu,'Label','Nou','Callback',@menunou); % Sub-menú 'nou'.
uimenu(menuarxiu,'Label','Carregar','Callback',@menucarregar); % Sub-menú 'carregar'.
uimenu(menuarxiu,'Label','Guardar','Callback',@menuguardar,'Separator','on'); % Sub-menú
'guardar'.
uimenu(menuarxiu,'Label','Guardar com a...','Callback',@menuguardarcom); % Sub-menu 'guardar
com'.
uimenu(menuarxiu,'Label','Sortir','Callback',@sortir,'Separator','on'); % Sub-menú 'sortir'.
menuopcions=uimenu(figcme2,'Label','Opcions'); % Menú principi 'opcions'.
uimenu(menuopcions,'Label','Editar opcions','Callback',@opcions); % Sub-menú 'editar opcions'.
end

function menunou(hObject,eventdata,handles)
% S'executa amb el menú 'nou'
% Pregunta si es vol guardar i esborra l'estructura.

resp=questdlg('Guardar abans d'esborrar l'estructura?','Nova
estructura','Si','No','Cancelar','Si');
switch resp
    case 'Si'
        menuguardar
        nou
    case 'No'
        nou
end
end

function nou
% Esborra les dades.

%disp('Nou')
%tic_nou=tic;
if grafica~=grafica_est % Si s'esta presentant resultats de barra.
    clic_grafica_res
end
nusus=nus; % nova llista d'objectes nus.
contnus=0; % comptador de nusos a zero.
barres=barra; % nova llista d'objectes barra.
contbarra=0; % comptador de barres a zero.
hgreaccions=[]; % esborra la llista de reaccions.
diagrames=diagrama; % nova llista d'objectes diagrama.
delete(get(grafica,'Children')) % esborra tots els objectes gràfics.
set(grafica_est,'XLim',XLimini,'YLim',YLimini,'ZLim',ZLimini) % Estableix els límits
originals.
centrar
set(botoedicio,'Value',1) % Torna al mode edició.
edicio % Esborra diagrames, etc..
nom=''; % Esborra el nom de l'estructura.
figcme2_rename % Canvia el títol.
fmax=0;
qmax=0;
%disp(['Nou: ',num2str(toc(tic_nou))])
end

function menucarregar(hObject,eventdata,handles)
% Carrega una estructura previament guardada.

carpetaactual=cd(direct); % Es col·loca a la carpeta on es guarden els treballs i guarda la
carpeta actual de Matlab.
[arxiu,carpeta]=uigetfile('*.cme2','Carregar'); % Obre la finestra per seleccionar arxiu.
cd(carpetaactual) % Un cop seleccionat l'arxiu torna a la carpeta de Matlab.

%disp('Carregar')
%tic_carregar=tic;

```

```

if arxiu % Si s'ha seleccionat un arxiu...
    if grafica~=grafica_est % Si s'esta presentant resultats de barra.
        clic_grafica_res
    end
    [nusos,barres,contnus,contbarra]=carregararxiu(fullfile(carpeta,arxiu)); % Carrega les
dades.
    % Esborra les dades de treballs anteriors.
    delete(get(grafica,'Children'))
    hgreaccions=[];
    diagrames=diagrama;
    for i=1:contnus
        nusos(i).hg=[];
        nusos(i).hgc=[];
    end

    for i=1:contbarra
        barres(i).hg=[];
        barres(i).hgpuntual=[];
        barres(i).hgmoment=[];
        barres(i).hgrepartida=[];
        barres(i).hgprojectada=[];
    end
    flagcalculat=0;
    maxpuntual % Calcula carrega puntual maxima.
    maxrepartida % Calcula carrega repartida o projectada maxima.
    redibuixar_nusos % Dibuixa els nusos.
    redibuixar_barres % Dibuixa les barres.
    redibuixa_carreganusos % Dibuixa les càrregues als nusos.
    centrar % Centra l'estructura a la pantalla.
    set(botoedicio,'Value',1) % Torna al mode edició.
    edicio
    nom=arxiu; % Nom de l'estructura.
    figcme2_rename % Canvia el títol.
end
figure(figcme2) % Retorna el focus a la finestra del programa, si no es fa el programa no
respondrà fins que no es cliqueji sobre la finestra.
%disp(['Carregar: ',num2str(toc(tic_carregar))])
end

function menuguardar(hObject,eventdata,handles)
    % Guarda les dades sobre l'estructura actual per recuperar-la més
    % endavant.

    %disp('Guardar')
    %tic_guardar=tic;
    if nom
        carpetaactual=cd(direct); % Es coloca a la carpeta on es guarden els treballs i guarda la
carpeta actual de Matlab.
        save(nom,'nusos','barres','-mat'); % Guarda les llistes de nusos i de barres que formen
l'estructura.
        cd(carpetaactual) % Torna a la carpeta de Matlab.
        figure(figcme2) % Retorna el focus a la finestra del programa, si no es fa el programa no
respondrà fins que no es cliqueji sobre la finestra.
    else
        menuguardarcom
    end
    %disp(['Guardar: ',num2str(toc(tic_guardar))])
end

function menuguardarcom(hObject,eventdata,handles)
    % Guarda les dades sobre l'estructura actual per recuperar-la més
    % endavant.

    %disp('Guardar com')
    %tic_guardar=tic;
    carpetaactual=cd(direct); % Es coloca a la carpeta on es guarden els treballs i guarda la
carpeta actual de Matlab.
    arxiu=uiinputfile('*.*cme2','Guardar com a...',nom); % Obre el diàleg per seleccionar o crear un
arxiu.
    if arxiu % Si s'ha seleccionat un arxiu
        nom=arxiu; % N'assigna el nom a l'estructura
        figcme2_rename % Canvia el títol.
        save(nom,'nusos','barres','-mat'); % Guarda les llistes de nusos i de barres que formen
l'estructura.
    end
    cd(carpetaactual) % Torna a la carpeta de Matlab.
    figure(figcme2) % Retorna el focus a la finestra del programa, si no es fa el programa no
respondrà fins que no es cliqueji sobre la finestra.
    %disp(['Guardar com: ',num2str(toc(tic_guardar))])
end

function sortir(hObject,eventdata,handles)

```

```

% Surt del programa.
% S'executa al pitjar sobre l'aspa o al premer el menú sortir.
% Executa un msgbox per confirmar i guardar l'estructura.

resp=questdlg('Guardar abans de sortir?','Sortir de CME2','Si','No','Cancelar','Si');
switch resp
    case 'Si'
        menuguardar
        delete(figcme2) % Esborra la finestra.
    case 'No'
        delete(figcme2) % Esborra la finestra.
end
end

function opcions(hObject,eventdata,handles)
% S'executa al seleccionar el menú d'edició de les opcions del
% programa.

dlgopcions % Obre la finestra per modificar les opcions, que les guardara a l'arxiu opcions.
mat.

[editallcrear,barraalcrearnus,fontnusus,tamanyfontnusus,fontcarregues,tamanyfontcarregues,
factorzoom,tamanyusus,distnumnus,precini,XLimini,YLimini,ZLimini,colormodificar,coloresborrar,
colorcarregues,colorreaccions,colordeformada,colorN,colorTy,colorTz,colorMx,colorMy,colorMz,direct]
=loadopcions; % Carrega les opcions de l'arxiu opcions.mat.

%disp('Opcions')
%tic_opcions=tic;
% Modifica els colors dels botons segons les opcions.
set(botocarregues,'ForegroundColor',colorcarregues);
set(botoreaccions,'ForegroundColor',colorreaccions);
set(botodeformada,'ForegroundColor',colordeformada);
set(botoaxils,'ForegroundColor',colorN);
set(bototallantY,'ForegroundColor',colorTy);
set(bototallantZ,'ForegroundColor',colorTz);
set(botomomentX,'ForegroundColor',colorMx);
set(botomomentY,'ForegroundColor',colorMy);
set(botomomentZ,'ForegroundColor',colorMz);

for i=1:contnus
    set(findobj(nusus(i).hg,'type','text'),'FontName',fontnusus,'FontSize',tamanyfontnusus)
    set(findobj(nusus(i).hgc,'type','text'),'FontName',fontcarregues,'FontSize',
tamanyfontcarregues)
    if flagcalculat
        set(findobj(hgreaccions(i),'type','text'),'FontName',fontcarregues,'FontSize',
tamanyfontcarregues)
    end
end

colors
clicable % Estableix quins objectes responen al ratoli segons els botons de la barra.
%disp(['/Opcions: ',num2str(toc(tic_opcions))])
end

% CREACIÓ I ACCIONS SOBRE EL PANELL QUE CONTÉ LES GRÀFIQUES, LES COORDENADES I LA PRECISIÓ.

function [panell,logo,grafica_resxy,grafica_resxz,txtbox_res,grafica_est,editx,edity,editz,
editprec]=creacio_panell_grafica
% Creació de la pantalla gràfica:
panell=uipanel(figcme2,'Units','pixels','Position',[5,5,tm-
[130,10]],'BorderStyle','etchedin','BorderWidth',1); % Creacio del panell.
logo=axes('Parent',panell,'Units','pixels','Position',[36,50,tm-[176,70]]); % Objecte axes on
es carreguen els logos.
imagecsc(imread('logo.jpg'),'ButtonDownFcn',@clic_grafica_res); % Carrega dels logos.
set(logo,'handlevisibility','off','Visible','off'); % S'oculten els marges i es fa
inaccessible. Queda com a imatge de fons sense alterar al treballar amb l'estructura, ampliar, reduir,
moure, etc.
grafica_resxy=axes('Parent',panell,'Units','pixels','Position',[36,(tm(2)+30)/2,(tm-[182,70])
/2],'Box','on','color','none','NextPlot','add','Visible','off','ButtonDownFcn',@clic_grafica_res); %
Pantalla de presentació de resultats de barra en eixos xy locals.
grafica_resxz=axes('Parent',panell,'View',[0,0],'Units','pixels','Position',[36,50,(tm-
[182,70])/2],'Box','on','color','none','NextPlot','add','Visible','off','ButtonDownFcn',
@clic_grafica_res); % Pantalla de presentació de resultats de barra en eixos xz locals.
txtbox_res=icontrol(panell,'Units','pixels','Position',[tm(1)/2-49,50,tm(1)/2-182,tm(2)
-70],'Style','edit','max',2,'FontName','default','FontUnits','pixels','FontSize',
12,'HorizontalAlignment','left','BackgroundColor','white','Visible','off'); % Editbox on es presenta
l'informe de resultats.
grafica_est=axes('Parent',panell,'Units','pixels','Position',[36,50,tm-
[176,70]],'color','none','MinorGridLineStyle',':','ButtonDownFcn',@clic_ratoli,'NextPlot','add','XLim',
XLimini,'YLim',YLimini,'Box','on'); % Pantalla de treball amb la gràfica de l'estructura.
icontrol(panell,'Units','pixels','Position',
[36,5,15,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',

```

```

14, 'HorizontalAlignment', 'left', 'String', 'X:'); % Etiqueta.
    editx=icontrol(panel1, 'Units', 'pixels', 'Position', ↵
[51,5,50,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', ↵
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'Enable', 'off'); % Caixa on s'indica o ↵
introueix la coordenada x.
    uicontrol(panel1, 'Units', 'pixels', 'Position', ↵
[111,5,15,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', ↵
14, 'HorizontalAlignment', 'left', 'String', 'Y:'); % Etiqueta.
    edity=icontrol(panel1, 'Units', 'pixels', 'Position', ↵
[126,5,50,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', ↵
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'Enable', 'off'); % Caixa on s'indica o ↵
introdueix la coordenada y.
    uicontrol(panel1, 'Units', 'pixels', 'Position', ↵
[186,5,15,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', ↵
14, 'HorizontalAlignment', 'left', 'String', 'Z:'); % Etiqueta.
    editz=icontrol(panel1, 'Units', 'pixels', 'Position', ↵
[201,5,50,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', ↵
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', '0'); % Caixa on s'indica o ↵
introdueix la coordenada z.
    uicontrol(panel1, 'Units', 'pixels', 'Position', ↵
[256,5,59,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', ↵
14, 'HorizontalAlignment', 'left', 'String', 'Precisió:'); % Etiqueta.
    editprec=icontrol(panel1, 'Units', 'pixels', 'Position', ↵
[315,5,50,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', ↵
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(precini)); % Caixa on ↵
s'introdueix la precisió a l'introduir punts amb el ratolí.
    grafica=grafica_est;
end

function clic_ratoli(hObject,eventdata,handles)
% S'executa quan es cliqueja sobre la pantalla gràfica.
% flagclic es la variable que crea la funció clic_figure, que s'executa
% just abans i indica el botó que s'ha pres.

% Si es cliqueja el botó esquere sobre el planol amb els botons de
% la barra crear i nus en mode edició crea un nus.
if flagclic==1 && get(botoedicio, 'Value') && strcmp(get(botobarracrear, 'State'), 'on') && strcmp(↵
(get(botobarranus, 'State'), 'on') && ~strcmp(vista, '3d'))
    crearnus
% Si s'està en mode zoom amplia o redueix segons el cas.
elseif (flagclic==1 && strcmp(get(botobarraapropar, 'State'), 'on')) || (flagclic==3 && strcmp(↵
(get(botobarraallunyar, 'State'), 'on'))
    zom(-3/factorzoom);
elseif (flagclic==1 && strcmp(get(botobarraallunyar, 'State'), 'on')) || (flagclic==3 && strcmp(↵
(get(botobarraapropar, 'State'), 'on'))
    zom(3/factorzoom);
end
end

function clic_nus(gcbo,eventdata,handles)
% S'executa quan es cliqueja sobre un nus.
% Dins del camp 'UserData' de l'objecte gràfic que representa al
% nus hi ha enmagatzemat el seu número per tal d'identificar-lo.

n=get(gcbo, 'UserData'); % Número de nus.
if flagclic==1 && get(botoedicio, 'Value') % Si s'ha pres el botó esquerre i s'està en mode ↵
d'edició, depenent dels botons de la barra s'excuta una acció o una altra.
    if strcmp(get(botobarraesborrar, 'State'), 'on') && strcmp(get(botobarranus, 'State'), 'on')
        esborrarnus(n)
    elseif strcmp(get(botobarramodificar, 'State'), 'on') && strcmp(get(↵
(botobarranus, 'State'), 'on')
        modificarnus(n)
    elseif strcmp(get(botobarracrear, 'State'), 'on') && strcmp(get(botobarrabarra, 'State'), 'on')
        crearbarra(n)
    elseif strcmp(get(botobarracrear, 'State'), 'on') && strcmp(get(↵
(botobarracarreganus, 'State'), 'on')
        crearmodificarcarreganus(n)
    end
    elseif flagclic==3 && ~flagbarra && get(botoedicio, 'Value') % Si s'ha pres el botó dret sobre ↵
el nus permet moure'l.
        flagmourenus=n; % Activa el flag que indica que es modifiqui la posició del nus amb el ↵
moviment del ratolí, a mes el flag conté el handle de l'objecte grafic que representa el nus.
    end
end

function clic_carreganus(gcbo,eventdata,handles)
% S'executa quan es cliqueja sobre la carrega d'un nus.
% Dins del camp 'UserData' de l'objecte gràfic que representa la
% carrega hi ha enmagatzemat el número de nus per tal
% d'identificar-lo.

n=get(gcbo, 'UserData'); % Número de nus.

```

```

        if flagclic==1 && get(botoedicio,'Value') && strcmp(get(botobarracarreganus,'State'),'on')% Si
s'ha pres el botó esquerre i s'està en mode d'edició, depenent dels botons de la barra s'excuta una
acció o una altra.
        if strcmp(get(botobarraesborrar,'State'),'on')
            esborrarcarreganus(n)
        elseif strcmp(get(botobarramodificar,'State'),'on')
            crearmodificarcarreganus(n)
        end
    end
end

function clic_barra(gcbo,eventdata,handles)
% S'executa quan es cliqueja sobre una barra.
% Dins del camp 'UserData' de la línia que representa la
% barra hi ha enmagatzemat el seu número per tal d'identificar-la.

n=get(gcbo,'UserData'); % Número de barra.
if flagclic==1 && get(botoedicio,'Value') % Si s'ha pres el botó esquerre i s'està en mode
edició, depenent dels botons de la barra s'executa una acció o una altra.
    if strcmp(get(botobarraesborrar,'State'),'on') && strcmp(get(botobarrabarra,'State'),'on')
        esborrarbarra(n)
        redibuixar_nusos
        escalar_nusos
    elseif strcmp(get(botobarramodificar,'State'),'on') && strcmp(get
(botobarrabarra,'State'),'on')
        modificarbarra(n)
    elseif strcmp(get(botobarracrear,'State'),'on') && strcmp(get
(botobarracarregabarra,'State'),'on')
        crearmodificarcarregabarra(n)
    end
    elseif flagclic==1 % Si es pren sobre la barra en el mode de presentacio de resultats,
presenta els resultats per a la barra seleccionada.
        resultat_barra(n)
    end
end

function clic_carregabarra(gcbo,eventdata,handles)
% S'executa quan es cliqueja sobre la càrrega d'una barra.
% Dins del camp 'UserData' de l'objecte grafic que representa la
% càrrega hi ha enmagatzemat el seu número per tal d'identificar-la.

n=get(gcbo,'UserData'); % Número de barra.
if flagclic==1 && get(botoedicio,'Value') % Si s'ha pres el botó esquerre i s'està en mode
edició, depenent dels botons de la barra s'executa una acció o una altra.
    if strcmp(get(botobarraesborrar,'State'),'on') && strcmp(get
(botobarracarregabarra,'State'),'on')
        esborrarcarregabarra(n)
    elseif strcmp(get(botobarramodificar,'State'),'on') && strcmp(get
(botobarracarregabarra,'State'),'on')
        crearmodificarcarregabarra(n)
    end
    elseif flagclic==1 % Si s'ha pres el botó esquerre en la presentacio de resultats.
        resultat_barra(n)
    end
end

%% CREACIÓ I ACCIONS SOBRE EL PANELL QUE CONTÉ ELS BOTONS QUE CONTROLLEN LES VISTES I EL MODE
EDICIÓ/PRESENTACIÓ DE RESULTATS.

function [panel2,botoedicio,botoxy,botoxz,botoyz,boto3d]=panell_botons_vista
% Creació del panell amb els botons.
panel2=uipanel(figcme2,'Units','pixels','Position',[tm-[120,190],
115,195],'BorderType','etchedin','BorderWidth',1,'FontName','default','FontUnits','pixels','FontSize',
14,'Title','Vista:'); % Creació del panell.
botoedicio=icontrol(panel2,'Units','pixels','Position',
[5,150,105,25],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Edició','Value',1,'Callback',@edicio,'KeyPressFcn',@tecla); % Creació botó edició.
botoreixeta=icontrol(panel2,'Units','pixels','Position',
[5,120,105,25],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Reixeta','Callback',@reixeta,'KeyPressFcn',@tecla); % Creació boto reixeta.
botocentrar=icontrol(panel2,'Units','pixels','Position',
[5,90,105,25],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Centrar','Callback',@centrar,'KeyPressFcn',@tecla); % Creació botó centrar.
botoxy=icontrol(panel2,'Units','pixels','Position',
[5,45,50,35],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','XY','Callback',@vistaxy,'Value',1,'KeyPressFcn',@tecla); % Creació botó vista XY.
botoxz=icontrol(panel2,'Units','pixels','Position',
[60,45,50,35],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','XZ','Callback',@vistaxz,'KeyPressFcn',@tecla); % Creació botó vista XZ.
botoyz=icontrol(panel2,'Units','pixels','Position',
[5,5,50,35],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','YZ','Callback',@vistayz,'KeyPressFcn',@tecla); % Creació botó vista YZ.

```



```

        boto3d=uicontrol(panel2,'Units','pixels','Position',
[60,5,50,35],'Style','togglebutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','3D','Callback',@vista3d,'KeyPressFcn',@tecla); % Creació vista 3D.
    end

function edicio(hObject,eventdata,handles)
    % Mode edicio/presentació
    % Si s'ha pitjat el botó edició esborra els diagrames, fa visibles
    % les carregues i torna als colors d'edició.

    %disp('Edicio')
    %tic_edicio=tic;
    if get(botoedicio,'Value') % Si el boto edició esta pitjat.
        flagcalculat=0; % Es restableix el flag que indica que hi ha una estructura ja calculada.
        set(get(panel3,'Children'),'Enable','off') % Es desactiven els botons que controlen els
resultats.
        set(get(barraeines,'Children'),'Enable','on') % S'activen els botons de la barra d'eines.
        esborra_diagramesireaccions % Esborra els diagrames
        maxpuntual % Calcula maxima força puntual sense tenir en compte reaccions
        redibuixar_carregues_barres % Redibuixa les carregues a les barres
        redibuixa_carreganusos % Redibuixa les carregues als nusos
        escalar_nusos % Escala els nusos
        clicable % Estableix quins objectes son susceptibles de ser cliquejats segons l'estat de la
bara d'eines
    else % Si el botó edició no està pitjat...
        set(get(panel3,'Children'),'Enable','on') % Activa els botons que controlen la mostra de
resultats.
        set(botobarracrear,'Enable','off') % Desactiven els botons de la barra d'eines.
        set(botobarramodificar,'Enable','off')
        set(botobarraesborrar,'Enable','off')
        set(botobarranus,'Enable','off')
        set(botobarrabarra,'Enable','off')
        set(botobarracarreganus,'Enable','off')
        set(botobarracarregabarra,'Enable','off')

        diagcarregues % Mostra les càrregues si estan seleccionades.
        diagreaccions % Mostra les reaccions si estan seleccionades.
        visibilitat_diagrames % Mostra els diagrames i la deformade si estan seleccionats.
    end
    colors
    %disp(['Edicio: ',num2str(toc(tic_edicio))])
end

function reixeta(hObject,eventdata,handles)
    % Activa i desactiva la reixeta.

    if get(hObject,'Value')
        grid on
        %grid minor % Reixeta més tupida.
    else
        grid off
    end
end

function centrar(hObject,eventdata,handles)
    % Centra l'estructura a la gràfica i estableix els límits.

    %disp('Centrar')
    %tic_centrar=tic;
    if grafica==grafica_est % Si s'estan mostrant els resultats en detall per barra no es fa res.
        calcul_cord_extrems % Càlcul de les coordenades extremes de la estructura.

        tam=get(grafica_est,'Position'); % Posició i tamany de la grafica. Valors de tamany
horitzontal en tam(3), i en Vertical en tam(4).

        switch vista
            case 'xy'
                if (xmax-xmin)*tam(4)/tam(3)<(ymax-ymin) % Si la relació alçada/amplada de
l'estructura es mes gran que la de la gràfica...
                    amp=(ymax-ymin)*tam(3)/tam(4); % Amplada horitzontal en funció de l'alçada
Vertical.
                    ampy=(ymax-ymin);
                    XL=[xmed-ampx/1.5,xmed+ampx/1.5];
                    YL=[ymed-ampy/1.5,ymed+ampy/1.5];
                elseif xmax==xmin && ymax==ymin
                    XL=XLimini+[xmed,xmed];
                    YL=XLimini*tam(4)/tam(3)+[ymed,ymed];
                else % Si la relació alçada/amplada és menor que la de la gràfica...
                    amp=(xmax-xmin);
                    ampy=(xmax-xmin)*tam(4)/tam(3); % Alçada en funció de l'amplada
                    XL=[xmed-ampx/1.5,xmed+ampx/1.5];
                    YL=[ymed-ampy/1.5,ymed+ampy/1.5];
                end
            end
        end
    end
end

```



```

end
set(grafica_est, 'XLim',XL, 'YLim',YL, 'ZLim',ZLimini); % Assigna els valors als eixos
de la gràfica.
case 'xz'
if (xmax-xmin)*tam(4)/tam(3)<(zmax-zmin) % Si la relació alçada/amplada de
l'estructura es mes gran que la de la gràfica...
ampx=(zmax-zmin)*tam(3)/tam(4); % Amplada horitzontal en funció de l'alçada
Vertical.

ampz=zmax-zmin;
XL=[xmed-ampx/1.5,xmed+ampx/1.5];
ZL=[zmed-ampz/1.5,zmed+ampz/1.5];
elseif xmax==xmin && zmax==zmin
XL=XLimini+[xmed,xmed];
ZL=XLimini*tam(4)/tam(3)+[zmed,zmed];
else % Si la relació alçada/amplada és menor que la de la gràfica...
ampx=xmax-xmin;
ampz=(xmax-xmin)*tam(4)/tam(3); % Alçada en funció de l'amplada
XL=[xmed-ampx/1.5,xmed+ampx/1.5];
ZL=[zmed-ampz/1.5,zmed+ampz/1.5];
end
set(grafica_est, 'XLim',XL, 'YLim',YLimini, 'ZLim',ZL); % Assigna els valors als eixos
de la gràfica.
case 'yz'
if (ymax-ymin)*tam(4)/tam(3)<(zmax-zmin) % Si la relació alçada/amplada de
l'estructura es mes gran que la de la gràfica...
ampy=(zmax-zmin)*tam(3)/tam(4); % Amplada horitzontal en funció de l'alçada
Vertical.

ampz=zmax-zmin;
YL=[ymed-ampy/1.5,ymed+ampy/1.5];
ZL=[zmed-ampz/1.5,zmed+ampz/1.5];
elseif ymax==ymin && zmax==zmin
YL=YLimini+[ymed,ymed];
ZL=YLimini*tam(4)/tam(3)+[zmed,zmed];
else % Si la relació alçada/amplada és menor que la de la gràfica...
ampy=ymax-ymin;
ampz=(ymax-ymin)*tam(4)/tam(3); % Alçada en funció de l'amplada
YL=[ymed-ampy/1.5,ymed+ampy/1.5];
ZL=[zmed-ampz/1.5,zmed+ampz/1.5];
end
set(grafica_est, 'XLim',XLimini, 'YLim',YL, 'ZLim',ZL); % Assigna els valors als eixos
de la gràfica.
case '3d'
ct=[xm,ym,zm]; % Punt 'camera target' centre de l'estructura.
minlim=min([xmin,ymin,zmin]); % Mínim
maxlim=max([xmax,ymax,zmax]); % Màxim
angle3d=pi/4;
set(grafica_est, 'XLim',[minlim,maxlim], 'YLim',[minlim,maxlim], 'ZLim',[minlim,
maxlim], 'CameraPosition',ct+[cos(angle3d),1,sin(angle3d)]); % Vista isomètrica.
end
escala % Calcula el factor d'escala s.
escalar_nusos % Reescala els nusos i les seves carregues.
redibuixar_carregues_barres % Redibuixa les càrregues sobre les barres amb la nova escala.
clicable % Estableix quins objectes responen al ratolí segons els botons de la barra.
end
%disp(['Centrar: ',num2str(toc(tic_centrar))])
end

function calcul_cord_extrems
% Calcula les coordenades extremes i mitjes de l'estructura.

if contnus % Si hi ha nusos definits...
xmin=nusos(1).posicio(1);
xmax=nusos(1).posicio(1);
ymin=nusos(1).posicio(2);
ymax=nusos(1).posicio(2);
zmin=nusos(1).posicio(3);
zmax=nusos(1).posicio(3);
xm=nusos(1).posicio(1);
ym=nusos(1).posicio(2);
zm=nusos(1).posicio(3);
for i=2:contnus % Si hi ha més d'un nus...
xmin=min(xmin,nusos(i).posicio(1)); % x mínima
xmax=max(xmax,nusos(i).posicio(1)); % x màxima
xm=xm+nusos(i).posicio(1); % sumatori de x
ymin=min(ymin,nusos(i).posicio(2));
ymax=max(ymax,nusos(i).posicio(2));
ym=ym+nusos(i).posicio(2);
zmin=min(zmin,nusos(i).posicio(3));
zmax=max(zmax,nusos(i).posicio(3));
zm=zm+nusos(i).posicio(3);
end
xm=xm/contnus; % crea la mitja de x

```

```

        ym=ym/contnus;
        zm=zm/contnus;
    else % Si no hi ha nusos definits...
        xmin=XLimini(1)*0.75;
        xmax=XLimini(2)*0.75;
        ymin=YLimini(1)*0.75;
        ymax=YLimini(2)*0.75;
        zmin=ZLimini(1)*0.75;
        zmax=ZLimini(2)*0.75;
        xm=(XLimini(1)+XLimini(2))*0.75/2;
        ym=(YLimini(1)+YLimini(2))*0.75/2;
        zm=(ZLimini(1)+ZLimini(2))*0.75/2;
    end
    xmed=(xmin+xmax)/2; % mitja dels extrems de x
    ymed=(ymin+ymax)/2;
    zmed=(zmin+zmax)/2;
end

function vistaxy(hObject,eventdata,handles)
    % S'executa quan es selecciona la vista XY.

    %disp('Vista XY')
    %tic_xy=tic;
    if get(hObject,'Value') && grafica==grafica_est
        vista='xy'; % Variable global vista controla el mode de vista actual del programa.
        set(botoxz,'Value',0); % Deselecciona les altres vistes.
        set(botoyz,'Value',0);
        set(boto3d,'Value',0);
        set(grafica_est,'View',[0,90],'Box','on'); % Vista del plànol XY.
        set(editx,'Enable','off'); % Desactiva la caixa de text amb la coordenada en x perque no es
pugui editar.
        set(edity,'Enable','off'); % Desactiva la caixa de text amb la coordenada en y perque no es
pugui editar.
        set(editz,'Enable','on','String','0'); % Activa la caixa de text amb la coordenada z per
editar la profunditat.
        redibuixar_nusos
        redibuixar_barres
        redibuixa_carreganusos
        %redibuixar_carregues_barres
        if flagcalculat
            esborra_diagramesireaccions
            dibuixa_reaccions
            dibuixa_diagramesideformada
            diagcarregues
            diagreaccions
            visibilitat_diagrames % Mostra els diagrames i la deformed si estan seleccionats.
        end
        centrar % Centra l'estructura a la gràfica.
    else
        set(hObject,'Value',abs(get(hObject,'Value')-1)); % Evita que es deseleccioni la vista, el
que faria que no hi hagués cap seleccionada.
    end
    %disp(['Vista XY: ',num2str(toc(tic_xy))])
end

function vistaxz(hObject,eventdata,handles)
    % S'executa quan es selecciona la vista XZ.

    %disp('Vista XZ')
    %tic_xz=tic;
    if get(hObject,'Value') && grafica==grafica_est
        vista='xz'; % Variable global vista controla el mode de vista actual del programa.
        set(botoxy,'Value',0); % Deselecciona les altres vistes.
        set(botoyz,'Value',0);
        set(boto3d,'Value',0);
        set(grafica_est,'View',[0,0],'Box','on'); % Vista del plànol XZ.
        set(editx,'Enable','off'); % Desactiva la caixa de text amb la coordenada en x perque no es
pugui editar.
        set(edity,'Enable','on','String','0'); % Activa la caixa de text amb la coordenada y per
editar la profunditat.
        set(editz,'Enable','off'); % Desactiva la caixa de text amb la coordenada en z perque no es
pugui editar.
        redibuixar_nusos
        redibuixar_barres
        redibuixa_carreganusos
        %redibuixar_carregues_barres
        if flagcalculat
            esborra_diagramesireaccions
            dibuixa_reaccions
            dibuixa_diagramesideformada
            diagcarregues
            diagreaccions

```

```

        visibilitat_diagrames % Mostra els diagrames i la deformed si estan seleccionats.
    end
    centrar % Centra l'estructura a la grafica.
else
    set(hObject, 'Value',abs(get(hObject, 'Value')-1)); % Evita que es deseleccioni la vista, el
que faria que no hi hagués cap seleccionada.
    end
    %disp(['Vista XY: ',num2str(toc(tic_xz))])
end

function vistayz(hObject,eventdata,handles)
    % S'executa quan es selecciona la vista YZ.

    %disp('Vista YZ')
    %tic_yz=tic;
    if get(hObject, 'Value') && grafica==grafica_est
        vista='yz'; % Variable global vista controla el mode de vista actual del programa.
        set(botoxy, 'Value',0); % Deselecciona les altres vistes.
        set(botoxz, 'Value',0);
        set(boto3d, 'Value',0);
        set(grafica_est, 'View',[90,0], 'Box','on'); % Vista del plànol YZ.
        set(editx, 'Enable','on','String','0'); % Activa la caixa de text amb la coordenada x per
editar la profunditat.
        set(edity, 'Enable','off'); % Desactiva la caixa de text amb la coordenada en y porque no es
pugui editar.
        set(editz, 'Enable','off'); % Desactiva la caixa de text amb la coordenada en z porque no es
pugui editar.
        redibuixar_nusos
        redibuixar_barres
        redibuixa_carreganusos
        %redibuixar_carregues_barres
        if flagcalculat
            esborra_diagramesireaccions
            dibuixa_reaccions
            dibuixa_diagramesideformada
            diagcarregues
            diagreaccions
            visibilitat_diagrames % Mostra els diagrames i la deformed si estan seleccionats.
        end
        centrar % Centra l'estructura a la grafica.
    else
        set(hObject, 'Value',abs(get(hObject, 'Value')-1)); % Evita que es deseleccioni la vista, el
que faria que no hi hagués cap seleccionada.
    end
    %disp(['Vista XY: ',num2str(toc(tic_yz))])
end

function vista3d(hObject,eventdata,handles)
    % S'executa quan es selecciona la vista 3D.
    % Es defineix que la càmera apunti al punt mig de l'estructura, i el punt d'observació a la
mateixa distància a tots els eixos per obtenir una vista isomètrica.
    % Es defineixen els límits iguals a tots els eixos.

    %disp('Vista 3D')
    %tic_3d=tic;
    if get(hObject, 'Value') && grafica==grafica_est
        vista='3d'; % Variable global vista controla el mode de vista actual del programa.
        set(botoxy, 'Value',0); % Deselecciona les altres vistes.
        set(botoxz, 'Value',0);
        set(botoyz, 'Value',0);

        set(editx, 'Enable','off'); % Desactiva les caixes de text amb les coordenades.
        set(edity, 'Enable','off');
        set(editz, 'Enable','off');
        redibuixar_nusos
        redibuixar_barres
        redibuixa_carreganusos
        %redibuixar_carregues_barres
        if flagcalculat
            esborra_diagramesireaccions
            dibuixa_reaccions
            dibuixa_diagramesideformada
            diagcarregues
            diagreaccions
            visibilitat_diagrames % Mostra els diagrames i la deformed si estan seleccionats.
        end
    end

    ct=[xm,ym,zm]; % Punt 'camera target' centre de l'estructura.
    minlim=min([xmin,ymin,zmin]); % Mínim
    maxlim=max([xmax,ymax,zmax]); % Màxim
    angle3d=pi/4;

```

```

        set(grafica_est, 'XLim', [minlim, maxlim], 'YLim', [minlim, maxlim], 'ZLim', [minlim, maxlim], 'CameraTarget', ct, 'CameraPosition', ct+[cos(angle3d), 1, sin(angle3d)], 'CameraUpVector', [0, 1, 0], 'Box', 'off'); % Vista isomètrica.
        axis equal % Mante la relació entre els eixos.

        centrar
    else
        set(hObject, 'Value', abs(get(hObject, 'Value')-1)); % Evita que es deseleccioni la vista, el que faria que no hi hagués cap seleccionada.
    end
    %disp([' /Vista XY: ', num2str(toc(tic_3d))])
end

% CREACIÓ I ACCIONS SOBRE EL PANELL QUE CONTÉ ELS BOTONS QUE CONTROLLEN ELS RESULTATS QUE ES MOSTREN.

function [panel3, botocarregues, botoreaccions, botodeformada, botoaxils, bototallantY, bototallantZ, botomomentX, botomomentY, botomomentZ]=panell_botons_grafica
    % Creació del panell amb els botons.
    panel3=uipanel(figcme2, 'Units', 'pixels', 'Position', [tm-[120, 390], 115, 200], 'BorderType', 'etchedin', 'BorderWidth', 1, 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'Title', 'Gràfica:'); % Creació del panell.
    botocarregues=icontrol(panel3, 'Units', 'pixels', 'Position', [5, 155, 105, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'Càrregues', 'ForegroundColor', colorcarregues, 'Enable', 'off', 'Callback', @diagcarregues, 'Value', 1, 'KeyPressFcn', @tecla); % Creació botó vista de càrregues.
    botoreaccions=icontrol(panel3, 'Units', 'pixels', 'Position', [5, 125, 105, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'Reaccions', 'ForegroundColor', colorreaccions, 'Enable', 'off', 'Callback', @diagreaccions, 'Value', 1, 'KeyPressFcn', @tecla); % Creació botó vista de reaccions.
    botodeformada=icontrol(panel3, 'Units', 'pixels', 'Position', [5, 95, 105, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'Deformada', 'ForegroundColor', colordeformada, 'Enable', 'off', 'Callback', @diagdeformada, 'Value', 1, 'KeyPressFcn', @tecla); % Creació botó vista de deformada.
    botoaxils=icontrol(panel3, 'Units', 'pixels', 'Position', [5, 65, 50, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'N', 'ForegroundColor', colorN, 'Enable', 'off', 'Callback', @diagN, 'Value', 1, 'KeyPressFcn', @tecla); % Creació botó vista de diagrama d'axils.
    bototallantY=icontrol(panel3, 'Units', 'pixels', 'Position', [5, 35, 50, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'Ty', 'ForegroundColor', colorTy, 'FontWeight', 'demi', 'Enable', 'off', 'Callback', @diagTy, 'Value', 1, 'KeyPressFcn', @tecla); % Creació botó vista de diagrama de tallants en y.
    bototallantZ=icontrol(panel3, 'Units', 'pixels', 'Position', [5, 5, 50, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'Tz', 'ForegroundColor', colorTz, 'FontWeight', 'demi', 'Enable', 'off', 'Callback', @diagTz, 'KeyPressFcn', @tecla); % Creació botó vista de diagrama de tallants en z.
    botomomentX=icontrol(panel3, 'Units', 'pixels', 'Position', [60, 65, 50, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'Mx', 'ForegroundColor', colorMx, 'FontWeight', 'demi', 'Enable', 'off', 'Callback', @diagMx, 'KeyPressFcn', @tecla); % Creació botó vista de diagrama de moments en x.
    botomomentY=icontrol(panel3, 'Units', 'pixels', 'Position', [60, 35, 50, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'My', 'ForegroundColor', colorMy, 'FontWeight', 'demi', 'Enable', 'off', 'Callback', @diagMy, 'KeyPressFcn', @tecla); % Creació botó vista de diagrames de moments en Y.
    botomomentZ=icontrol(panel3, 'Units', 'pixels', 'Position', [60, 5, 50, 25], 'Style', 'togglebutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'String', 'Mz', 'ForegroundColor', colorMz, 'FontWeight', 'demi', 'Enable', 'off', 'Callback', @diagMz, 'Value', 1, 'KeyPressFcn', @tecla); % Creació botó vista de diagrames de moments en z.
end

function diagcarregues(hObject, eventdata, handles)
    % Oculta o mostra les càrregues a la representació de resultats amb el seu color corresponent.

    %disp('Oculta o mostra diagrames carregues')
    %tic_dc=tic;

    if ~get(botoedicio, 'Value')
        if get(botocarregues, 'Value') % Si les càrregues estan seleccionades...
            vis='on';
        else
            vis='off';
        end
    end
    if grafica==grafica_est
        for i=1:contnus
            set(nusos(i).hgc, 'Visible', vis) % Visibilitat de les càrregues als nusos.
        end

        %redibuixar_carregues_barres % Redibuixa les càrregues.
        for i=1:contbarra
            set(barres(i).hgpuntual, 'Visible', vis) % Visibilitat de les càrregues a les barres.
            set(barres(i).hgmoment, 'Visible', vis)
            set(barres(i).hgpartida, 'Visible', vis)
        end
    end
end

```

```

        set(barres(i).hgprojectada, 'Visible', vis)
    end
end
end
end
%disp(['/Ocultat o mostra diagrames carregues: ', num2str(toc(tic_dc))])
end

function diagreaccions(hObject, eventdata, handles)
    % Oculta o mostra les reaccions a la representaci6 de resultats.

    %disp('Oculta o mostra reaccions')
    %tic_r=tic;
    if grafica==grafica_est
        if get(botoreaccions, 'Value')
            set(hgreaccions, 'Visible', 'on')
        else
            set(hgreaccions, 'Visible', 'off')
        end
    end
end
%disp(['/Ocultat o mostra reaccions: ', num2str(toc(tic_r))])
end

function diagdeformada(hObject, eventdata, handles)
    % Oculta o mostra la deformada a la representaci6 de resultats.

    %disp('Oculta o mostra la deformada')
    %tic_d=tic;
    if get(botodeformada, 'Value')
        vis='on';
    else
        vis='off';
    end
    if grafica==grafica_est
        for i=1:numel(diagrames)
            set(diagrames(i).hgdeformada, 'Visible', vis)
        end
    else
        set(diag_presxy.hgdeformada, 'Visible', vis)
        set(diag_presxz.hgdeformada, 'Visible', vis)
    end
end
%disp(['/Ocultat o mostra la deformada: ', num2str(toc(tic_d))])
end

function diagN(hObject, eventdata, handles)
    % Oculta o mostra el diagrama d'axils a la representaci6 de
    % resultats.

    %disp('Oculta o mostra el diagrama d'axils')
    %tic_N=tic;
    if get(botoaxils, 'Value')
        vis='on';
    else
        vis='off';
    end
    if grafica==grafica_est
        for i=1:numel(diagrames)
            set(diagrames(i).hgN, 'Visible', vis)
        end
    else
        set(diag_presxy.hgN, 'Visible', vis)
        set(diag_presxz.hgN, 'Visible', vis)
    end
end
%disp(['/Ocultat o mostra el diagrama d'axils: ', num2str(toc(tic_N))])
end

function diagTy(hObject, eventdata, handles)
    % Oculta o mostra el diagrama de tallants en y a la representaci6
    % de resultats.

    %disp('Oculta o mostra el diagrama de tallants en y')
    %tic_Ty=tic;
    if get(bototallantY, 'Value')
        vis='on';
    else
        vis='off';
    end
    if grafica==grafica_est
        for i=1:numel(diagrames)
            set(diagrames(i).hgTy, 'Visible', vis)
        end
    else
        set(diag_presxy.hgTy, 'Visible', vis)
    end
end

```

```

end
%disp(['Oculto o mostra el diagrama de tallants en y: ',num2str(toc(tic_Ty))])
end

function diagTz(hObject,eventdata,handles)
% Oculta o muestra el diagrama de tallants en z a la representació
% de resultats.

%disp('Oculto o mostra el diagrama de tallants en z')
%tic_Tz=tic;
if get(bototallantZ,'Value')
    vis='on';
else
    vis='off';
end
if grafica==grafica_est
    for i=1:numel(diagrames)
        set(diagrames(i).hgTz,'Visible',vis)
    end
else
    set(diag_presxz.hgTz,'Visible',vis)
end
%disp(['Oculto o mostra el diagrama de tallants en z: ',num2str(toc(tic_Tz))])
end

function diagMx(hObject,eventdata,handles)
% Oculta o muestra el diagrama de moments en x a la representació
% de resultats.

%disp('Oculto o mostra el diagrama de moments torsors')
%tic_Mx=tic;
if get(botomomentX,'Value')
    vis='on';
else
    vis='off';
end
if grafica==grafica_est
    for i=1:numel(diagrames)
        set(diagrames(i).hgMx,'Visible',vis)
    end
else
    set(diag_presxy.hgMx,'Visible',vis)
    set(diag_presxz.hgMx,'Visible',vis)
end
%disp(['Oculto o mostra el diagrama de moments torsors: ',num2str(toc(tic_Mx))])
end

function diagMy(hObject,eventdata,handles)
% Oculta o muestra el diagrama de moments en y a la representació
% de resultats.

%disp('Oculto o mostra el diagrama de moments flectors en y')
%tic_My=tic;
if get(botomomentY,'Value')
    vis='on';
else
    vis='off';
end
if grafica==grafica_est
    for i=1:numel(diagrames)
        set(diagrames(i).hgMy,'Visible',vis)
    end
else
    set(diag_presxz.hgMy,'Visible',vis)
end
%disp(['Oculto o mostra el diagrama de moments flectors en y: ',num2str(toc(tic_My))])
end

function diagMz(hObject,eventdata,handles)
% Oculta o muestra el diagrama de moments en z a la representació
% de resultats.

%disp('Oculto o mostra el diagrama de moments flectors en z')
%tic_Mz=tic;
if get(botomomentZ,'Value')
    vis='on';
else
    vis='off';
end
if grafica==grafica_est
    for i=1:numel(diagrames)
        set(diagrames(i).hgMz,'Visible',vis)
    end
end

```

```

        end
    else
        set(diag_presxy.hgMz, 'Visible', vis)
    end
    %disp(['Oculto o mostra el diagrama de moments flectors en z: ', num2str(toc(tic_Mz))])
end

function visibilitat_diagrames
    % Oculta o mostra el conjunt de diagrames i deformada.

    %disp('Visibilitat diagrames')
    %tic_vd=tic;
    diagN
    diagTy
    diagTz
    diagMx
    diagMy
    diagMz
    diagdeformada
    %disp(['Visibilitat diagrames: ', num2str(toc(tic_vd))])
end

%% CREACIÓ I ACCIONS QUE S'EXECUTEN AMB EL BOTÓ CALCULAR.

function botocalcular1=boto_calcular
    % Creació del boto calcular.
    botocalcular1=uicontrol(figcme2, 'Units', 'pixels', 'Position', [tm(1)
-120,5,115,25], 'Style', 'pushbutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'String', 'CÀLCUL', 'Callback', @calcular, 'KeyPressFcn', @tecla);
end

function calcular(hObject, eventdata, handles)
    % Surt del mode edició, elimina digrames anteriors.
    % Prepara les dades per cridar a la funcio de càlcul estàtic.
    % Representa les reaccions.
    % Amb el resultats del càlcul estàtic crida a la funció de càlcul i
    % dibuix dels diagrames i la deformada, els escala i col·loca al
    % seu lloc.

    %disp('Calcular')
    %tic_calc=tic;
    if grafica~=grafica_est % Si no s'esta presentant resultats de barra.
        clic_grafica_res
    end
    if contbarra % Si hi ha barres.
        esborra_diagramesireaccions % Esborra els diagrames i les reaccions anteriors si n'hi han.
        [Ncoord, Ncond, Nextfor, Lnods, Lartyp, Lmecp, barra_carrega_rep, barra_carrega_rep_proj,
barra_carrega_punt, barra_carrega_mom]=prepara_dades_calcul; % Adapta les dades al format de la funcio
de càlcul.
        [Retorn_reaccions, Retorn_moviments, Retorn_esforços_barres, TL2G] = cme2_calcul_estatic
(contnus, contbarra, Ncoord, Ncond, Nextfor, Lnods, Lartyp, Lmecp, barra_carrega_rep, barra_carrega_rep_proj,
barra_carrega_punt, barra_carrega_mom); % Crida a la funció de càlcul estàtic.
        flagcalculat=1; % Flag que indica que s'ha calculat una estructura. Permet a la funció que
controla l'escala de la deformada amb el teclat conèixer si hi ha deformada.
        maxpuntual % Calcula forces puntuals o reaccions maximes.
        dibuixa_reaccions % Dibuixa les reaccions.
        redibuixar_carregues_barres
        redibuixa_carreganusos
        escalar_nusos
        dibuixa_diagramesideformada % Dibuixa els diagrames i la deformada.
        set(botoedicio, 'Value', 0); % Botó edició en mode presentacio.
        edicio % Desactiva botons de la barra d'eines per a l'edició, activa els de mostra de
resultats, etc.
    end
    %disp(['Calcular: ', num2str(toc(tic_calc))])
end

function esborra_diagramesireaccions
    % Esborra els diagrames i reaccions anteriors.

    %disp('Esborra diagrames i reaccions')
    %tic_esb=tic;
    for i=1:length(diagrames) % Esborra els diagrames anteriors.
        delete(diagrames(i).hgN)
        delete(diagrames(i).hgTy)
        delete(diagrames(i).hgTz)
        delete(diagrames(i).hgMx)
        delete(diagrames(i).hgMy)
        delete(diagrames(i).hgMz)
        delete(diagrames(i).hgdeformada)
    end
    for i=1:length(hgreaccions) % Esborra les reaccions anteriors.
        if hgreaccions(i)

```



```

        delete(hgreaccions(i))
    end
end
hgreaccions=[];
diagrames=diagrama;
%disp(['/Esborra diagrames i reaccions: ',num2str(toc(tic_esb))])
end

function [Ncoord,Ncond,Nextfor,Lnodes,Lartyp,Lmecp,barra_carrega_rep,barra_carrega_rep_proj,
barra_carrega_punt,barra_carrega_mom]=prepara_dades_calcul
% Prepara les dades al format d'entrada de la funció de càlcul estàtic.

%disp('Prepara dades calcul')
%tic_prep=tic;
Ncoord=zeros(contnus,3); % Coordenades dels nusos.
Ncond=zeros(contnus,7); % Condicions de contorn.
Nextfor=zeros(contnus*6,1); % Càrregues als nusos.
for i=1:contnus
    Ncoord(i,:)=nusos(i).posicio; % Coordenades dels nusos.
    Ncond(i,1)=nusos(i).contorn; % Condicions de contorn.
    switch nusos(i).contorn
        case 1 % rodet
            Ncond(i,2:4)=nusos(i).rodet; % Condició de contorn rodet.
        case {4,5,6} % moviments elàstics
            Ncond(i,2:7)=nusos(i).rig; % Condició de contorn elàsticitat.
        case {7,8,9} % moviments imposats
            Ncond(i,2:7)=nusos(i).desp; % Condició de contorn moviment.
    end
    Nextfor(i*6-5:i*6-3)=nusos(i).puntual; % Càrregues puntuals als nusos.
    Nextfor(i*6-2:i*6)=nusos(i).moment; % Càrregues moment als nusos.
end

Lnodes=zeros(contbarra,2); % Connectivitat dels extrems de barra.
Lartyp=zeros(contbarra,2); % articulació dels extrems de barra.
Lmecp=zeros(contbarra,9); % Propietats mecàniques de les barres.
barra_carrega_rep=zeros(contbarra,6); % Càrregues repartides.
barra_carrega_rep_proj=zeros(contbarra,6); % Càrregues projectades.
barra_carrega_punt=zeros(contbarra,5); % Càrregues puntuals.
barra_carrega_mom=zeros(contbarra,5); % Càrregues moment.
for i=1:contbarra
    Lnodes(i,1)=barres(i).nusinf; % Connectivitat de l'extrem inferior de barra.
    Lnodes(i,2)=barres(i).nussup; % Connectivitat de l'extrem superior de barra.
    Lartyp(i,1)=barres(i).artinf; % articulació de l'extrem inferior de barra.
    Lartyp(i,2)=barres(i).artsup; % articulació de l'extrem superior de barra.
    Lmecp(i,1)=barres(i).E; % Modul de Young.
    Lmecp(i,2)=9.81; % Gravetat, herència de CME, de fet no s'utilitza.
    Lmecp(i,3)=barres(i).A; % Àrea de la secció.
    Lmecp(i,4)=barres(i).Iz; % Moment d'inèrcia Iz.
    Lmecp(i,5)=barres(i).Iy; % Moment d'inèrcia Iy.
    Lmecp(i,6:8)=barres(i).vectorIz; % Vector direcció inèrcia Iz.
    Lmecp(i,9)=barres(i).GIT; % Rigidesa a torsió.
    barra_carrega_rep(i,:)=barres(i).repartida; % Càrrega repartida.
    barra_carrega_rep_proj(i,:)=barres(i).projectada; % Càrrega projectada.
    barra_carrega_punt(i,:)=barres(i).puntual; % Càrrega puntual.
    barra_carrega_mom(i,:)=barres(i).moment; % Càrrega moment.
end
%disp(['/Prepara dades calcul: ',num2str(toc(tic_prep))])
end

function dibuixa_reaccions
% Dibuixa les reaccions.

%disp('Dibuixa reaccions')
%tic_dr=tic;
hgreaccions=zeros(contnus,1); % Guarda l'objecte gràfic en una llista.
for i=1:contnus % Repasa els nusos.
    if any(Return_reaccions(6*i-5:6*i)) % Si hi ha alguna reacció al nus...
        hgreaccions(i)=dibuix_carreganus(i,nusos(i).posicio,[xm,ym,zm],Return_reaccions(6*i-5:
6*i-3)',Return_reaccions(6*i-2:6*i)',vista,fontcarregues,tamanyfontcarregues,fmax); % Dibuixa la
reacció com una càrrega al nus.
        m=makehgtform('translate',nusos(i).posicio,'scale',s); % Escala i col·loca la càrrega a
la posició del nus.
        set(hgreaccions(i),'UserData',i,'Matrix',m);
        chgcolor(hgreaccions(i),colorreaccions) % Dóna color a les reaccions.
        if strcmp(vista,'3d')
            set(get(hgreaccions(i),'Children'),'Clipping','off') % Fa que els objectes gràfics
sobresurtin de la grafica.
        else
            set(get(hgreaccions(i),'Children'),'Clipping','on') % Fa que els objectes gràfics
NO sobresurtin de la grafica.
        end
    end
end
end
end

```

```

end
%disp(['Dibuixa reaccions: ',num2str(toc(tic_dr))])
end

function dibuixa_diagramesideformada
% Dibuixa els diagrames i la deformada.

%disp('Dibuixa diagrames i deformada')
%tic_dd=tic;
Tmax=0; % Tallant maxim.
Mmax=0; % Moment maxim.
Dmax=0; % Deformació màxima.
mov_extrems=zeros(12,contbarra);
for i=1:contbarra % Per a totes les barres...
    [long,esforcos,moviments,rep,proj,punt,mom,Iz,Iy]=dades_diagrama(i);
    diagrames(i)=dibuix_diagrames(long,esforcos,moviments,rep,proj,punt,mom,barres(i).E,barres
(i).A,Iz,Iy); % Calcula i dibuixa els diagrames.
    diagrames(i).num=i; % Número de barra a la qual pertany el diagrama.
    Tmax=max([Tmax,diagrames(i).Tmax/long]); % Calcula la relació tallant o axil/longitud
màxima de totes les barres.
    Mmax=max([Mmax,diagrames(i).Mmax/long]); % Calcula la relació moment/longitud màxim de
totes les barres.
    Dmax=max([Dmax,diagrames(i).Dmax/long]); % Calcula la relació deformació/longitud màxima de
totes les barres.
end

sT=1/(2*Tmax); % Càlcul de l'escala dels diagrames d'axils i tallants.
sM=1/(2*Mmax); % Càlcul de l'escala dels diagrames de moments.
sD=1/(5*Dmax); % Càlcul de l'escala de la deformada.

% Coloca, escala i coloreja els diagrames i la deformada.
for i=1: numel(diagrames)
    pos=nusos(barres(i).nusinf).posicio; % El diagrama es col·loca a la posició del nus
inferior.
    vector_barra=nusos(barres(i).nussup).posicio-pos; % Vector direcció de la barra, per a
calcular la matriu de rotació dels diagrames. No s'utilitzen els eixos locals, perquè poden no
coincidir l'eix z local i global.
    mT=matbarra(vector_barra,pos,sT); % Càlcul de la matriu de rotació, posició i escala dels
diagrames de tallant.
    mM=matbarra(vector_barra,pos,sM); % Càlcul de la matriu de rotació, posició i escala dels
diagrames de moments.
    set(diagrames(i).hgN,'Matrix',mT); % Col·locació del diagrama d'axils.
    set(diagrames(i).hgTy,'Matrix',mT); % Col·locació del diagrama de tallants en y.
    set(diagrames(i).hgTz,'Matrix',mT); % Col·locació del diagrama de tallants en z.
    set(diagrames(i).hgMx,'Matrix',mM); % Col·locació del diagrama de moments en x.
    set(diagrames(i).hgMy,'Matrix',mM); % Col·locació del diagrama de moments en y.
    set(diagrames(i).hgMz,'Matrix',mM); % Col·locació del diagrama de moments en z.
    if strcmp(vista,'3d')
        clip3d='off'; % Fa que els objectes gràfics sobresurtin de la grafica.
    else
        clip3d='on'; % Fa que els objectes gràfics NO sobresurtin de la grafica.
    end
    set(get(diagrames(i).hgN,'Children'),'Clipping',clip3d)
    set(get(diagrames(i).hgTy,'Children'),'Clipping',clip3d)
    set(get(diagrames(i).hgTz,'Children'),'Clipping',clip3d)
    set(get(diagrames(i).hgMx,'Children'),'Clipping',clip3d)
    set(get(diagrames(i).hgMy,'Children'),'Clipping',clip3d)
    set(get(diagrames(i).hgMz,'Children'),'Clipping',clip3d)
    set(get(diagrames(i).hgdeformada,'Children'),'Clipping',clip3d)
    escala_deformada(TL2G(:, :, i), i, sD) % Col·locació i rotació de la deformada. Aquí SI que
s'utilitzen els eixos locals, ja que la deformació es calcula en eixos locals.
    coloridiagrama(diagrames(i))
end
%disp(['Dibuixa diagrames i deformada: ',num2str(toc(tic_dd))])
end

function [long,esforcos,moviments,rep,proj,punt,mom,Iz,Iy]=dades_diagrama(n)

%disp(['Prepara dades calcul diagrames barra n° ',num2str(n)])
%tic_dd=tic;
n1=barres(n).nusinf;
n2=barres(n).nussup;
T=TL2G(:, :, n); % Matriu de canvi de base d'exos globals a locals.
esforcos(1:12)=Retorn_esforcos_barres(n,1:12); % Esforços en extrems de barra en eixos locals.
moviments=[T*Retorn_moviments(n1*6-5:n1*6-3);T*Retorn_moviments(n1*6-2:n1*6);T*Retorn_moviments
(n2*6-5:n2*6-3);T*Retorn_moviments(n2*6-2:n2*6)]; % Moviments dels extrems de barra en eixos locals.
mov_extrems(:,n)=moviments; % Guarda els moviments en extrems de barra en eixos locals per a la
funcio que mostra l'informe de resultats.
vbarra=nusos(n2).posicio-nusos(n1).posicio; % Vector barra.
long=norm(vbarra); % Longitud de la barra.
rep=[barres(n).repartida(1),(T*barres(n).repartida(2:4))',barres(n).repartida(5:6)]; % Carrega
repartida en eixos locals.

```

```

    proj=[barres(n).projectada(1),(T*barres(n).projectada(2:4)'),'barres(n).projectada(5:6)]; % C
Càrrega projectada en eixos locals.
    punt=[barres(n).puntual(1),(T*barres(n).puntual(2:4)'),'barres(n).puntual(5)]; % Carrega en
eixos locals.
    mom=[barres(n).moment(1),(T*barres(n).moment(2:4)'),'barres(n).moment(5)]; % Moments en eixos
locals.
    Iz=barres(n).Iz; % Inercia Iz.
    if all(barres(n).vectorIz==0) % Si no esta definit el vector direcció d'Iz, Iy=Iz.
        Iy=Iz;
    else
        Iy=barres(n).Iy;
    end
    %disp(['/Prepara dades calcul diagrames barra n° ',num2str(n),' ': ',num2str(toc(tic_dd))])
end

function colorddiagrames
    % Coloreja tots els diagrames i la deformada de l'estructura.

    %disp('Color diagrames')
    %tic_cd=tic;
    for i=1:contbarra
        colordiagrama(diagrames(i))
    end
    %disp(['/Color diagrames: ',num2str(toc(tic_cd))])
end

function colordiagrama(diag)
    % Coloreja els diagrames i la deformada de l'objecte diagrama.

    %disp(['/Color diagrama ',num2str(diag.num)])
    %tic_cd=tic;
    chgcolor(diag.hgN,colorN); % Color diagrama d'axils.
    chgcolor(diag.hgTy,colorTy); % Color diagrama de tallants en y.
    chgcolor(diag.hgTz,colorTz); % Color diagrama de tallants en z.
    chgcolor(diag.hgMx,colorMx); % Color diagrama de moments en x.
    chgcolor(diag.hgMy,colorMy); % Color diagrama de moments en y.
    chgcolor(diag.hgMz,colorMz); % Color diagrama de moments en z.
    chgcolor(diag.hgdeformada,colordeformada); % Color deformada.
    %disp(['/Color diagrama ',num2str(diag.num),' ': ',num2str(toc(tic_cd))])
end

function mat=matbarra(vector_barra,posicio,escala)
    % Calcula la matriu de gir, escala i posició per a representar
    % els diagrames al plànol de la vista.
    % Calcula la matriu per tal que els diagrames es representin al
    % plà format per la barra i l'eix Vertical segons la vista.

    vector_barra=vector_barra/norm(vector_barra);
    if abs(vector_barra(3))==1
        vector3=[vector_barra(3),0,0];
        vector2=[0,1,0];
    else
        vector2=cross([0,0,1],vector_barra); % Calcula dos vectors perpendiculars al primer i entre
ells per crear la matriu de canvi de base.
        vector2=vector2/norm(vector2);
        vector3=cross(vector_barra,vector2);
        vector3=vector3/norm(vector3);
    end

    if strcmp(vista,'xz') || strcmp(vista,'yz') % Si s'esta a les vistes amb eix Vertical z,
s'intercanvien els vectors per mostrar els diagrames adequadament.
        vector2=vector3;
        vector3=vector2;
        vector2=vector2_;
    end
    mat(1:4,1)=[vector_barra';0]; % Matriu de gir, posició i escala.
    mat(1:4,2)=[vector2'*escala;0];
    mat(1:4,3)=[vector3'*escala;0];
    mat(1:4,4)=[posicio';1];
end

function escala_deformada(TL2G,n,escala)
    % Escala, gira i col·loca la deformada.
    % Afegeix els scatter3 que marquen la fletxa màxima.

    mat(1:4,1)=[TL2G(:,1);0]; % Direcció i escala x.
    mat(1:4,2)=[TL2G(:,2)*escala;0]; % Direcció i escala y.
    mat(1:4,3)=[TL2G(:,3)*escala;0]; % Direcció i escala z.
    mat(1:4,4)=[nunos(barres(n).nusunf).posicio';1]; % Posició.

    diag=findobj(diagrames(n).hgdeformada,'type','line'); % Línia que representa la deformada.
    defx=get(diag,'UserData'); % Recupera els valors per al recàlcul de la deformada en x escalada.

```

```

x=defx.x+defx.e*escala; % Recalcula la deformada en x escalada.
set(diag, 'X', x) % Modifica el valors en x de la linia que representa la deformada, amb la nova
deformació calculada.
set(diagrames(n).hgdeformada, 'Matrix', mat) % Col·loca i escala la deformada a l'extrem
inferior de la barra.
end

function resultat_barra(n)
% S'executa quan es cliqueja sobre una barra a la presentació de
% resultats de tota l'estructura.
% Mostra els resultats detallats per a la barra seleccionada.
% Mostra dos gràfiques amb la representació dels eixos locals xy i
% xz on es presenten els diagrames i la deformada amb valors
% numerics a la meitat esquerra. A la meitat dreta mostra una caixa
% de text amb un informe de propietats de la barra, lleis
% d'esforços i deformació.

grafica=grafica_resxy;
long=norm(nusos(barres(n).nussup).posicio-nusos(barres(n).nusinf).posicio); % Longitud de la
barra.

set(grafica_est, 'Visible', 'off')

set(txtbox_res, 'Visible', 'on') % Fa visible el textbox que conté l'informe.
set(grafica_resxy, 'XLim', [0-long/10, long*1.1], 'YLim', [-1.1, 1.4]) % Dimensiona la presentació en
xy perquè hi capiga la barra, els diagrames es dimensionaran entre -1 i 1.
set(grafica_resxz, 'XLim', [0-long/10, long*1.1], 'ZLim', [-1.1, 1.4]) % Dimensiona la presentació en
xz perquè hi capiga la barra, els diagrames es dimensionaran entre -1 i 1.
set(get(grafica_est, 'Children'), 'Visible', 'off') % Oculta la gràfica de l'estructura.

txtnum=[num2str(barres(n).num), ' (' , num2str(barres(n).nusinf), '- ', num2str(barres(n).
nussup), ')']; % Text número de barra i nusos.
text(0-long/15, 1.4, 0, ['X-Y local barra n°', txtnum, ':'], 'VerticalAlignment', 'top', 'Parent',
grafica_resxy) % Text títol vista xy.
text(0-long/15, 0, 1.4, ['X-Z local barra n°', txtnum, ':'], 'VerticalAlignment', 'top', 'Parent',
grafica_resxz) % Text títol vista xz.

line([0, long], [0, 0], [0, 0], 'color', 'k', 'LineWidth', 2, 'Parent', grafica_resxy) % Dibuixa la barra
a la vista xy.
line([0, long], [0, 0], [0, 0], 'color', 'k', 'LineWidth', 2, 'Parent', grafica_resxz) % Dibuixa la barra
a la vista xz.

[long, esforcos, moviments, rep, proj, punt, mom, Iz, Iy]=dades_diagrama(n);
[diag_presxy, informe]=dibuix_diagrames(long, esforcos, moviments, rep, proj, punt, mom, barres(n).E,
barres(n).A, Iz, Iy); % Calcula i dibuixa els diagrames.
diag_presxy.Mmax=diag_presxy.Mmax+(diag_presxy.Mmax==0); % Impedeix que Mmax sigui zero.
diag_presxy.Tmax=diag_presxy.Tmax+(diag_presxy.Tmax==0); % Impedeix que Mmax sigui zero.
diag_presxy.Dmax=diag_presxy.Dmax+(diag_presxy.Dmax==0); % Impedeix que Mmax sigui zero.

colordiagrama(diag_presxy) % Coloreja els diagrames.
diag_presxz=diag_presxy; % Copia els handles (no els objectes, només la referència) dels
dibuixos dels diagrames en representació xy a la representació xz.
diag_presxz.hgdeformada=copyobj(diag_presxy.hgdeformada, grafica_resxz); % Copia la deformada
(crea un objecte còpia independent) a la representació xz.

set(diag_presxy.hgN, 'Parent', grafica_resxy, 'Matrix', makehgtform('scale', [1, 1/diag_presxy.Tmax,
1])); % Assigna el diagrama axils a la representació xy i l'escala.
set(diag_presxy.hgTy, 'Parent', grafica_resxy, 'Matrix', makehgtform('scale', [1, 1/diag_presxy.Tmax,
1])); % Assigna el diagrama tallants en y a la representació xy i l'escala.
set(diag_presxy.hgMz, 'Parent', grafica_resxy, 'Matrix', makehgtform('scale', [1, 1/diag_presxy.Mmax,
1])); % Assigna el diagrama moments en z a la representació xy i l'escala.
set(diag_presxy.hgdeformada, 'Parent', grafica_resxy, 'Matrix', makehgtform('scale', [1, .
5/diag_presxy.Dmax, 1])); % Assigna una de les còpies de la deformada a la representació xy i l'escala.
delete(findobj(diag_presxy.hgdeformada, 'UserData', 'dz')) % Elimina les referències a la
deformada en z a la representació xy.

set(diag_presxz.hgTz, 'Parent', grafica_resxz, 'Matrix', makehgtform('xrotate', pi/2, 'scale', [1, 1
/diag_presxz.Tmax, 1])); % Assigna el diagrama de tallants en z a la representació xz i l'escala.
set(diag_presxz.hgMx, 'Parent', grafica_resxz, 'Matrix', makehgtform('xrotate', pi/2, 'scale', [1, 1
/diag_presxz.Mmax, 1])); % Assigna el diagrama de moments en x a la representació xz i l'escala.
set(diag_presxz.hgMy, 'Parent', grafica_resxz, 'Matrix', makehgtform('xrotate', pi/2, 'scale', [1, 1
/diag_presxz.Mmax, 1])); % Assigna el diagrama moments en y a la representació xz i l'escala.
set(diag_presxz.hgdeformada, 'Parent', grafica_resxz, 'Matrix', makehgtform('scale', [1, 1, .
5/diag_presxz.Dmax])); % Assigna una de les còpies de la deformada a la representació xz i l'escala.
delete(findobj(diag_presxz.hgdeformada, 'UserData', 'dy')) % Elimina les referències a la
deformada en y a la representació xz.

visibilitat_diagrames % Mostra els diagrames i la deformada si estan seleccionats.

txtposinf=vector2txt(nusos(barres(n).nusinf).posicio); % Posició nus inferior.
txtpossup=vector2txt(nusos(barres(n).nussup).posicio); % Posició nus superior.
if any(barres(n).vectorIz) % Si està definit el vector direcció Iz...

```

```

        txtinerciaIz=num2str(barres(n).Iz), ' , direcció ',vector2txt(barres(n).vectorIz), ' eix Z
local']; % Texte vector direcció Iz..
        txtinerciaIy=num2str(barres(n).Iy); % Iz i Iy són diferents.
    else
        txtinerciaIz=num2str(barres(n).Iz); % Texte Iz.
        txtinerciaIy=['Iz= ',num2str(barres(n).Iz)]; % Si no està definit el vector Iz Iy=Iz.
    end
    if barres(n).artinf % Si l'extrem inferior és articulat.
        txtartinf='articulat';
        txtgirs1={['Girs: ',vector2txt([0,diagrames(n).Oini]), ' / ',vector2txt(TL2G(:, :,n)*[0;
diagrames(n).Oini'])]}; % El gir de l'extrem de barra articulat no coincideix amb el gir del nus! Gir
calculat amb la deformada.
    else
        txtartinf='rígid';
        txtgirs1={['Girs: ',vector2txt(mov_extrems(4:6,n)), ' / ',vector2txt(TL2G(:, :,n)*mov_extrems
(4:6,n))]}; % Si l'extrem és rígid el seu gir és el gir del nus.
    end
    if barres(n).artsup % Si l'extrem superior es articulat.
        txtartsup='articulat';
        txtgirs2={['Girs: ',vector2txt([0,diagrames(n).Ofin]), ' / ',vector2txt(TL2G(:, :,n)*[0;
diagrames(n).Ofin'])]}; % El gir de l'extrem de barra articulat no coincideix amb el gir del nus! Gir
calculat amb la deformada.
    else
        txtartsup='rígid';
        txtgirs2={['Girs: ',vector2txt(mov_extrems(10:12,n)), ' / ',vector2txt(TL2G(:, :,n)
*mov_extrems(10:12,n))]}; % Si l'extrem és rígid el seu gir és el gir del nus.
    end
        txtttitol={['INFORME DE RESULTATS PER A LA BARRA N°: ',txtnum,':']; % Texte del títol.
        txtpropietats={['Longitud= ',num2str(long)];['Mòdul de Young E= ',num2str(barres(n).E)];['Àrea
de la secció= ',num2str(barres(n).A)];['Moment d'Inèrcia Iz= ',txtinerciaIz];['Moment d'Inèrcia Iy=
',txtinerciaIy];['Rigidesa a torsió GIT= ',num2str(barres(n).GIT)]; % Texte propietats mecàniques de
la barra.
        txteixos={['x'=' ',vector2txt(TL2G(:, 1,n)), ' ; y'=' ',vector2txt(TL2G(:, 2,n)), ' ; z'=' ',vector2txt
(TL2G(:, 3,n))]};
        txtextrem1={['Extrem ',txtartinf, ' al nus n° ',num2str(barres(n).nusinf), ', posició en eixos
globals: ',txtposinf]}; % Texte extrem inferior.
        txtesf1={['Forces: ',vector2txt(Return_esforcos_barres(n,1:3)), ' / ',vector2txt(TL2G(:, :,n)
*Return_esforcos_barres(n,1:3))]}; % Texte esforços a l'extrem inferior.
        txtmom1={['Moments: ',vector2txt(Return_esforcos_barres(n,4:6)), ' / ',vector2txt(TL2G(:, :,n)
*Return_esforcos_barres(n,4:6))]}; % Texte moments a l'extrem inferior.
        txtdesp1={['Desplaçaments: ',vector2txt(mov_extrems(1:3,n)), ' / ',vector2txt(TL2G(:, :,n)
*mov_extrems(1:3,n))]}; % Texte desplaçaments a l'extrem inferior.
        txtextrem2={['Extrem ',txtartsup, ' al nus n° ',num2str(barres(n).nussup), ', posició en eixos
globals: ',txtpossup]}; % Texte extrem superior.
        txtesf2={['Forces: ',vector2txt(Return_esforcos_barres(n,7:9)), ' / ',vector2txt(TL2G(:, :,n)
*Return_esforcos_barres(n,7:9))]}; % Texte esforços a l'extrem superior.
        txtmom2={['Moments: ',vector2txt(Return_esforcos_barres(n,10:12)), ' / ',vector2txt(TL2G(:, :,n)
*Return_esforcos_barres(n,10:12))]}; % Texte moments a l'extrem superior.
        txtdesp2={['Desplaçaments: ',vector2txt(mov_extrems(7:9,n)), ' / ',vector2txt(TL2G(:, :,n)
*mov_extrems(7:9,n))]}; % Texte desplaçaments a l'extrem superior.

        txt=[txtttitol;{'PROPIETATS: ';};txtpropietats;{' ';EIXOS LOCALS: ';};txteixos;{' ';ESFORÇOS I
MOVIMENTS A L'EXTREM INFERIOR EN EIXOS LOCALS/GLOBALS: ';};txtextrem1;txtesf1;txtmom1;txtdesp1;
txtgirs1;{' ';ESFORÇOS I MOVIMENTS A L'EXTREM SUPERIOR EN EIXOS LOCALS/GLOBALS: ';};txtextrem2;
txtesf2;txtmom2;txtdesp2;txtgirs2;{' ';DIAGRAMES D'ESFORÇOS I DEFORMADA EN EIXOS LOCALS: ';};
informe]; % Pega els textos.
        set(txtbox_res,'String',txt); % Mostra el text.
    end

function txt=vector2txt(vector)
    % Converteix un vector en text.

    txt=['(',num2str(vector(1)), ', ',num2str(vector(2)), ', ',num2str(vector(3)), ')'];
end

function clic_grafica_res(hObject,eventdata,handles)
    % S'executa quan es cliqueja sobre la representació de resultats en
    % detall per una barra. Torna a mostrar els resultats per a tota
    % l'estructura.

    if grafica~=grafica_est
        grafica=grafica_est; % gràfica de treball és la gràfica de l'estructura.
        axes(grafica_est) % gràfica de treball.
        set(grafica_est,'Visible','on') % Fa visible la gràfica de l'estructura.
        set(txtbox_res,'Visible','off') % Oculta el text amb l'informe de resultats de la barra.
        %set(grafica_resxy,'Visible','off') % Oculta la gràfica de la barra en xy.
        %set(grafica_resxz,'Visible','off') % Oculta la gràfica de la barra en xz.
        delete(get(grafica_resxy,'Children')); % Elimina el contingut de les gràfiques de les
barres.

        delete(get(grafica_resxz,'Children'));
        diag_presxy=diagrama; % Esborra les referències als diagrames mostrats en detall.
        diag_presxz=diagrama; % Esborra les referències als diagrames mostrats en detall.

```



```

    for i=1:contnus
        set(nusos(i).hg, 'Visible', 'on') % Fa visibles els nusos.
    end
    for i=1:contbarra
        set(barres(i).hg, 'Visible', 'on') % Fa visibles les barres.
    end
    diagcarregues % Fa visibles les càrregues.
    diagreaccions % Fa visibles les reaccions.
    visibilitat_diagrames % Mostra els diagrames i la deformeda si estan seleccionats.
end
end

%% CREACIÓ I MODIFICACIÓ DINÀMICA DE NUSOS
% Funcions relacionades amb la creació, modificació, eliminació, dibuix,
% orientació dels nusos i les seves càrregues.

function crearnus
    % Crea i dibuixa un nou nus.

    %disp('Crear nus')
    %tic_cn=tic;
    posicio=[str2double(get(editx, 'String')),str2double(get(edity, 'String')),str2double(get
(editz, 'String'))]; % Obtè les coordenades de la posició de les caixes de text, ja arrodonides a la
precisió.
    contnus=contnus+1; % Incrementa el comptador de nusos en un.
    nusos(contnus)=nus; % Nou objecte nus.
    nusos(contnus).num=contnus; % Una de les dades que conte l'estructura de la classe nus es el
seu número.
    nusos(contnus).posicio=posicio; % posició del nus.
    calcul_cord_extrems % Recalcula màxims, mínims de les coordenades dels nusos per tal de que els
nous nusos s'orientin correctament.
    if editalcrear
        modificarnus(contnus) % Si està seleccionada l'opció, edita el nus després de crear-lo.
    end
    if barraalcrearnus && contnus > 1 % Si està seleccionada l'opció crea una barra al crear el nus
connectada al nus anterior.
        novabarra(contnus-1,contnus,artnusant,strcmp(get(botobarrarotula, 'State'), 'on')) % Crea una
barra des del nus anterior al nus creat.
    end
    artnusant=strcmp(get(botobarrarotula, 'State'), 'on'); % Apunta l'estat del botó ròtula al crear
l'últim nus.
    redibuixar_nusos % Dibuixa i gira els nusos segons les barres connectades o la seva posició
relativa.
    escala % Calcula el factor d'escala s.
    escalar_nusos % Reescala els nusos i les seves carregues.
    %disp(['/Crear nus: ',num2str(toc(tic_cn))])
end

function esborrarnus(n)
    % Elimina un nus i la seva representació gràfica.
    % Elimina les barres connectades al nus i actualitza la numeració.

    %disp(['Esborrar nus ',num2str(n)])
    %tic_en=tic;

    esborrarcarganus(n) % elimina la carrega al nus
    if contbarra>0 % Si hi han barres definides...
        for i=contbarra:-1:1 % Compte endarrera (endarrera per tal de que l'acció d'esborrar barres
modifica la llista barres).
            if barres(i).nusinf==n || barres(i).nussup==n % Si algun dels extrems estava connectat
al nus que s'ha eliminat...
                esborrarbarra(i) % elimina també la barra
            end
        end
    end
    delete(nusos(n).hg) % Destruïeix l'objecte gràfic que representa el nus.
    nusos2=nus; % Crea una nova llista de nusos.
    if n>1
        nusos2(1:n-1)=nusos(1:n-1); % Copia els nusos anteriors a la nova llista.
    end
    if n<contnus
        nusos2(n:contnus-1)=nusos(n+1:contnus); % Copia els nusos posteriors a la nova llista.
        for i=n:contnus-1
            nusos2(i).num=i; % Actualitza número de nusos dels nusos posteriors a l'eliminat.
            set(nusos2(i).hg, 'UserData', i); % Actualitza camp 'UserData' amb el nou número de nus.
            set(findobj(nusos2(i).hg, 'Type', 'text'), 'String', num2str(i)); % Actualitza el text.
        end
    end
    nusos=nusos2; % Substitueix la llista antiga de nusos per la nova.
    contnus=contnus-1; % Actualitza el comptador de nusos.

    if contbarra>0 % Si hi han definides...

```

```

for i=1:contbarra
    if barres(i).nusinf>n % Si el nus inferior té número superior a l'elimitat...
        barres(i).nusinf=barres(i).nusinf-1; % Actualitza el número.
    end
    if barres(i).nussup>n % Si el nus superior té número superior a l'elimitat...
        barres(i).nussup=barres(i).nussup-1; % Actualitza el número.
    end
end
end

calcul_cord_extrems % Recalcula màxims, mínims de les coordenades dels nusos per tal de que els
nous nusos s'orientin correctament.
escala % Calcula el factor d'escala s.
redibuixar_nusos % Reorienta, gira, els nusos segons les barres connectades o la seva posició
relativa.
escalar_nusos % Adequa l'escala.
clicable % Activa o desactiva la pulsaio dels elements grafics segons els botons de la barra.

%disp(['Esborrar nus ',num2str(n),': ',num2str(toc(tic_en))])
end

function modificarnus(n)
    % Modifica un nus.

    %disp(['Modificar nus ',num2str(n)])
    %tic_mn=tic;
    nusos(n)=dlgnus(nusos(n)); % Obre la finestra d'edició del nus.
    calcul_cord_extrems % Recalcula màxims, mínims de les coordenades dels nusos per si s'han
modificat les coordenades.
    escala % Calcula el factor d'escala s.
    redibuixar_nusos % Reorienta, gira, els nusos segons les barres connectades o la seva posició
relativa.
    redibuixar_barres % Redibuixa les barres, per si el nus te barres connectades i s'ha modificat
la posició.
    redibuixar_carregues_barres % Redibuixa les carregues sobre les barres.
    escalar_nusos % Adequa l'escala.
    clicable % Activa o desactiva la pulsaio dels elements grafics segons els botons de la barra.
    %disp(['Modificar nus ',num2str(n),': ',num2str(toc(tic_mn))])
end

function redibuixar_nus(n)
    % Dibuixa el nus, elimina el dibuix anterior.
    % Gira el dibuix en funcio de les barres connectades o la posició
    % relativa.

    %disp(['Redibuixar nus ',num2str(n)])
    %tic_rn=tic;
    con=[]; % Llistat amb els angles de les barres connectades al nus.
    for i=1:contbarra
        if barres(i).nusinf==n % Si el nus està connectat a l'extrem inferior d'una barra...
            vec=nusos(barres(i).nussup).posicio-nusos(n).posicio; % Vector direcció de la barra.
            con(length(con)+1)=atan2(vec(2),vec(1)); % Llistat amb els angles de les barres
connectades al nus.
        elseif barres(i).nussup==n % Si el nus està connectat a l'extrem superior d'una barra...
            vec=nusos(barres(i).nusinf).posicio-nusos(n).posicio; % Vector direcció de la barra.
            con(length(con)+1)=atan2(vec(2),vec(1)); % Llistat amb els angles de les barres
connectades al nus.
        end
    end
end

switch nusos(n).contorn % Discrimina segons el tipus de condicio de contorn.
    case 0 % Nus lliure
        ang=0; % No es gira.
    case 1 % Rodet
        ang=atan2(nusos(n).rodet(2),nusos(n).rodet(1))-pi/2; % Es gira segons el seu pla
definit.
    case {2,4} % Recolzaments
        % Per defecte no es gira. Si té una barra que travesa el recolzament el gira per evitar
l'efecte anti-estètic.
        if isempty(con) % Si no hi han barres connectades...
            ang=0; % No es gira.
        else
            alfa=ones(1,numel(con))*5*pi/6;
            con2=con+alfa;
            if all(con2 > 2*pi/3 | con2 < 0) % Si no hi ha una barra sota el recolzament...
                ang=0; % No es gira.
            else
                beta=ones(1,numel(con))*pi/2;
                con2=con2-beta;
                con2=con2+2*pi*(con2 < -pi);
                if all(con2 > 2*pi/3 | con2 < 0) % Si no hi ha una barra a la dreta del
recolzament...

```



```

        ang=pi/2; % Es gira 90°.
    else
        con2=con2+pi;
        con2=con2-2*pi*(con2 > pi);
        if all(con2 > 2*pi/3 | con2 < 0) % Si no hi ha una barra a l'esquerra del
recolzament...
            ang=3*pi/2; % Es gira 270°.
        else
            con2=con2+beta;
            con2=con2-2*pi*(con2 > pi);
            if all(con2 > 2*pi/3 | con2 < 0) % Si no hi ha una barra per sobre del
recolzament...
                ang=pi; % Es gira 180°.
            else
                ang=0; % Si hi ha barres que interfereixen en totes direccions, no
es gira.
            end
        end
    end
end
end
end
end
case 3 % Encastaments
if numel(con)==1 % Si només està connectat a una barra...
    ang=con-pi/2; % Segueix la direcció de la barra.
else % Si està connectat a més d'una barra...
    ang=atan2(sum(sin(con)),sum(cos(con)))-pi/2; % Gira l'angle mig.
    ang=round(ang*12/pi)*pi/12; % Arrodoneix l'angle.
end
case {5,6} % Girs elàstics amb desplaçaments nuls, girs elàstics amb desplaçaments elàstics
tics
    if isempty(con) % si no hi han barres connectades...
        ang=0; % No es gira.
    else % Si hi han connectades una barra o més...
        ang=con(1)-pi/2; % Segueix la direcció de la primera.
    end
case 8 % Girs imposats i desplaçaments lliures.
    if isempty(con) % Si no hi han barres connectades...
        ang=0; % No es gira.
    else % Si hi han connectades una barra o més...
        ang=con(1); % Segueix la direcció de la primera.
    end
otherwise
    ang=0; % En qualsevol altre cas no es gira.
end

nusos(n).gir=ang; % Apunta l'angle de gir.
hg=dibuix_contorn(nusos(n).contorn,distnumnus,fontnusos,tamanyfontnusos); % Dibuixa el nus.
set(hg,'ButtonDownFcn',@clic_nus,'UserData',n); % Assigna la funció que s'executarà quan es
clicueji.
ht=findobj(hg,'Type','text'); % Objecte que conté el text de numeració del nus dins de
l'objecte gràfic que representa el nus.
pos=get(ht,'Position'); % N'obté la posició.
angt=atan2(nusos(n).posicio(2)-ym,nusos(n).posicio(1)-xm); % Angle de situació del text al
voltant del nus, segons la posició relativa del nus.
angt=round((angt-pi/4)*2/pi)*pi/2+pi/4; % Arrodoneix l'angle.
angt=angt-ang; % Resta l'angle de gir del nus per tal de que al girar-lo el text vagi a la
seva posició calculada.
pos(1:2)=[cos(angt),sin(angt)]*distnumnus; % Calcula en coordenades la posició del text.
set(ht,'String',num2str(n),'Position',
pos,'HorizontalAlignment','center','VerticalAlignment','middle'); % Assigna la posició al text.
if strcmp(vista,'3d')
    set(get(hg,'Children'),'Clippin','off') % Permet que en 3d sobresurti de la gràfica.
else
    set(get(hg,'Children'),'Clippin','on') % NO permet que en 3d sobresurti de la gràfica.
end
delete(nusos(n).hg) % Esborra el dibuix anàtic.
nusos(n).hg=hg; % Guarda el handle del nou objecte gràfic a la llista de nusos.
%disp(['Redibuixar nus ',num2str(n),': ',num2str(toc(tic_rn))])
end

function redibuixar_nusos
% Redibuixa tots els nusos.
%disp('Redibuixar nusos')
%tic_rn=tic;
for i=1:contnus
    redibuixar_nus(i) % Redibuixa cada nus.
end
colornusos % Coloreja ls nusos.
%disp(['Redibuixar nusos: ',num2str(toc(tic_rn))])
end

function escala

```

```

% Relaciona el tamany de la representació dels objectes gràfics amb
% el tamany de la gràfica, per tal de que sigui constant encara que
% s'amplii i redueixi. Els objectes gràfics sempre es mostraran del
% mateix tamany.

% Pren com a referència l'eix horitzontal.
%disp('Escala')
%tic_s=tic;
switch vista
    case 'yz'
        xl=get(grafica_est,'YLim'); % Al plànol yz l'eix horitzontal és l'eix y.
    case '3d'
        xl=get(grafica_est,'XLim')*3; % En 3D els eixos són iguals, s'aplica un factor de
multiplicació per compensar que la proporcionalitat amb el tamany de la pantalla canvia.
    otherwise
        xl=get(grafica_est,'XLim'); % A la resta de vistes es pren l'eix x.
end

tam=get(grafica_est,'Position'); % tam(3) és el tamany horitzontal de la grafica, en píxels.
s=(xl(2)-xl(1))*tamanyus/tam(3); % Calcul del factor escala s.
%disp(['Escala: ',num2str(toc(tic_s))])
end

function escalar_nusos
% Ecala el nusos i les seves carregues i reaccions segons la variable d'escala s calculada.

%disp('Escalar_nusos')
%tic_en=tic;
for i=1:contnus % Per a tots els nusos.
    t=makehgtform('translate',nusos(i).posicio,'zrotate',nusos(i).gir,'scale',s); % Matriu de
gir, desplaçament i escala.
    set(nusos(i).hg,'Matrix',t); % Assigna a l'objecte que representa el nus la matriu de gir,
desplaçament i escala.
    if any(nusos(i).puntual) || any(nusos(i).moment) % Si hi han forces aplicades al nus...
        t=makehgtform('translate',nusos(i).posicio,'scale',s); % Matriu de desplaçament i
escala.
        set(nusos(i).hgc,'Matrix',t); % Assigna a l'objecte que representa les forces la matriu
de desplaçament i escala.
    end
    if numel(hgreaccions) % Si hi han reaccions...
        if hgreaccions(i)
            t=makehgtform('translate',nusos(i).posicio,'scale',s); % Matriu de desplaçament i
escala.
            set(hgreaccions(i),'Matrix',t); % Assigna a l'objecte que representa les reaccions
la matriu de desplaçament i escala.
        end
    end
end
end

%disp(['Escalar nusos: ',num2str(toc(tic_en))])
end

function crearmodificarcarreganus(n)
% Crea o modifica la carrega d'un nus.

%disp(['Crea o modifica carrega en nus ',num2str(n)])
%tic_cn=tic;
nusos(n)=dlgcarreganus(nusos(n)); % Obre la finestra d'edició de les càrregues dels nusos.
maxpuntual % Recalcula la càrrega maxima.
redibuixa_carreganus(n); % Dibuixa de nou la càrrega.
escalar_nusos % Escala la carrega i la resta d'objectes gràfics.
colorcarreganusos
%disp(['Crea o modifica carrega en nus ',num2str(n),': ',num2str(toc(tic_cn))])
end

function redibuixa_carreganus(n)
% Crea els objectes gràfics que representen les carregues als
% nusos.

%disp(['Redibuixa carrega nus ',num2str(n)])
%tic_rc=tic;
delete(nusos(n).hgc); % Elimina l'objecte grafic que representa la càrrega.
nusos(n).hgc=dibuix_carreganus(n,nusos(n).posicio,[xm,ym,zm],nusos(n).puntual,nusos(n).moment,
vista,fontcarregues,tamanyfontcarregues,fmax); % Crea l'objecte gràfic.
set(nusos(n).hgc,'ButtonDownFcn',@clic_carreganus,'UserData',n); % Assigna a l'objecte la
funció que s'executara quan el cliqueji sobre la càrrega i el número de nus per identificar-lo.
if strcmp(vista,'3d')
    set(get(nusos(n).hgc,'Children'),'Clippin','off') % Permet que en 3d sobresurti de la
gràfica.
else
    set(get(nusos(n).hgc,'Children'),'Clippin','on') % NO permet que en 3d sobresurti de la
gràfica.
end
end

```

```

end
%clicable % Estableix els colors i quins objectes gràfics responen als clics segons l'estat
dels botons de la barra d'eines.
%disp(['Redibuixa carrega nus ',num2str(n),': ',num2str(toc(tic_rc))])
end

function redibuixa_carreganusos
% Redibuixa totes les càrregues als nusos.

%disp('Redibuixa carrega nusos')
%tic_rc=tic;
for i=1:contnus
    redibuixa_carreganusos(i) % Redibuixa les càrregues.
end
colorcarreganusos
escalar_nusos
%disp(['Redibuixar carrega nusos: ',num2str(toc(tic_rc))])
end

function esborrrarcarreganusos(n)
% Esborra la carrega d'un nus.

%disp(['Esborrar carrega nus ',num2str(n)])
%tic_ec=tic;
%delete(nusos(n).hgc); % Elimina l'objecte grafic que representa la càrrega.
%nusos(n).hgc=[]; % Elimina la referència a l'objecte eliminat.
nusos(n).puntual=[0,0,0]; % Restableix a zeros els valors de la càrrega.
nusos(n).moment=[0,0,0]; % Restableix a zero els valors del moment.
maxpuntual % Recalcula la càrrega maxima.
redibuixa_carreganusos
%disp(['Esborrar carrega nus ',num2str(n),': ',num2str(toc(tic_ec))])
end

%% CREACIÓ I MODIFICACIÓ DINÀMICA DE BARRES
% Funcions relacionades amb la creació, modificacio, eliminació, dibuix,
% de les barres i les seves càrregues.

function crearbarra(n)
% Crea una barra a l'estructura.
% Si es clica sobre el primer nus crea una línia que segueix el
% cursor fins clicar el segon nus, o es canceli amb la tecla
% 'escape'.

%disp('Crear barra')
%tic_cb=tic;
if flagbarra % Si ja s'ha iniciat el proces de crear la barra, s'ha clicat sobre el segon nus.
    novabarra(primernusbarra,n,artprimernusbarra,strcmp(get(botobarrarotula,'State'),'on')) %
Crea la barra.
    flagbarra=0; % Reestableix a zero el flag.
    delete(hbarra); % Elimina la barra que segueix el cursor.
    hbarra=[];
else % Si s'ha clicat sobre el primer nus.
    primernusbarra=n; % Apunta el número de nus.
    artprimernusbarra=strcmp(get(botobarrarotula,'State'),'on'); % Apunta si està seleccionat
el botó articulació.
    posicio=[str2double(get(editx,'String')),str2double(get(edity,'String')),str2double(get
(editz,'String'))]; % Obté la posició del punter del ratolí.
    hbarra=line([nusos(n).posicio(1) posicio(1)],[nusos(n).posicio(2) posicio(2)],[nusos(n).
posicio(3) posicio(3)],'color','k','LineWidth',2,'Clipping','off','HitTest','off','Parent',
grafica_est); % Crea una línia des del nus clicat fins a la posició del ratolí.
    flagbarra=1; % Activa el flag per tal de que al clicar sobre un altre nus es completi el
procés de creació de la barra.
end
%disp(['Crear barra: ',num2str(toc(tic_cb))])
end

function novabarra(nusinf,nussup,artinf,artsup)
% Crea una barra des del nus inferior al superior.

%disp(['Nova barra ',num2str(nusinf),'-',num2str(nussup)])
%tic_nb=tic;
contbarra=contbarra+1; % Incrementa el comptador de barres.
barres(contbarra)=barra; % Nou objecte barra a la llista de barres.
barres(contbarra).nusinf=nusinf; % Nus inferior.
barres(contbarra).nussup=nussup; % Nus superior.
barres(contbarra).artinf=artinf; % articulació a l'extrem inferior.
barres(contbarra).artsup=artsup; % articulació a l'extrem superior.
barres(contbarra).num=contbarra; % Assigna número de barra a la barra.
if editalcrear % Si està seleccionada l'opció edita la barra després de crear-la.
    modificarbarra(contbarra) % Edita i dibuixa la barra.
else
    redibuixar_barra(contbarra) % Dibuixa la barra.
end

```

```

        colorbarres
        clicable % Estableix quins objectes gràfics responen als clics segons l'estat dels botons
de la barra d'eines.
    end
    %disp(['Nova barra ',num2str(nusinf),'-',num2str(nussup),' : ',num2str(toc(tic_nb))])
end

function redibuixar_barra(n)
    % Elimina el dibuix existent i el redibuixa.

    %disp(['Redibuixar barra ',num2str(n)])
    %tic_rb=tic;
    delete(barres(n).hg); % Esborra l'objecte gràfic.
    posinf=nusos(barres(n).nusinf).posicio; % Posició nus inferior.
    possup=nusos(barres(n).nussup).posicio; % Posició nus superior.
    barres(n).hg=line([posinf(1) possup(1)],[posinf(2) possup(2)],[posinf(3) possup(3)
(3)],'LineWidth',2,'UserData',n,'ButtonDownFcn',@clic_barra,'Parent',grafica_est); % Crea la línia.
    if strcmp(vista,'3d')
        set(barres(n).hg,'Clippin','off') % Permet que en 3d sobresurti de la gràfica.
    else
        set(barres(n).hg,'Clippin','on') % NO permet que en 3d sobresurti de la gràfica.
    end
    %disp(['Redibuixar barra ',num2str(n),' : ',num2str(toc(tic_rb))])
end

function redibuixar_barres
    % Elimina els objectes grafics que representen les barres i les
    % dibuixa de nou.
    %disp('Redibuixar barres')
    %tic_rb=tic;
    for i=1:contbarra % Per a totes les barres.
        redibuixar_barra(i);
    end
    %clicable % Estableix els colors i quins objectes gràfics responen als clics segons l'estat
dels botons de la barra d'eines.
    %disp(['Redibuixar barres: ',num2str(toc(tic_rb))])
    colorbarres
end

function esborrarbarra(n)
    % Esborra la barra indicada.
    % Elimina de la llista i elimina els objectes gràfics que
    % representen la barra i les seves càrregues.

    %disp(['Esborrar barra ',num2str(n)])
    %tic_eb=tic;
    nusinf=barres(n).nusinf; % Nus inferior.
    nussup=barres(n).nussup; % Nus superior.
    esborrarcarregabarra(n); % Esborra la carrega.
    delete(barres(n).hg) % Elimina l'objecte gràfic que representa la barra.
    barres2=barra; % Crea una nova llista de barres.
    if n>1
        barres2(1:n-1)=barres(1:n-1); % Copia les barres amb número inferior a la nova llista.
    end
    if n<contbarra
        barres2(n:contbarra-1)=barres(n+1:contbarra); % Copia les barres amb número superior a la
nova llista.
        for i=n:contbarra-1
            barres2(i).num=i; % Actualitza els números de barra.
            set(barres2(i).hg,'UserData',i); % Actualitza els números de barra al camp 'UserData'
dels objectes grafics.
        end
    end
    barres=barres2; % Elimina la llista de barres antiga i la substitueix per la nova.
    contbarra=contbarra-1; % Actualitza el comptador de barres.
    redibuixar_nus(nusinf) % Actualitza el nus inferior, la seva orientació por dependre de la
barra.
    redibuixar_nus(nussup) % Actualitza el nus superior, la seva orientació por dependre de la
barra.
    %clicable % Estableix els colors i quins objectes gràfics responen als clics segons l'estat
dels botons de la barra d'eines.
    %disp(['Esborrar barra ',num2str(n),' : ',num2str(toc(tic_eb))])
end

function modificarbarra(n)
    % Edita la barra.

    barres=dlgbarra(contnus,barres,n); % Diàleg de moifiació de les barres.
    %disp(['Modificar barra ',num2str(n)])
    %tic_mb=tic;
    redibuixar_nus(barres(n).nusinf) % Redibuixa el nus inferior, modificacions de la barra poden
afectar la seva orientacio.

```

```

redibuirar_nus(barres(n).nussup) % Redibuixa el nus superior, modificacions de la barra poden
afectar la seva orientacio.
escalar_nusos
redibuirar_barra(n) % Redibuixa la barra.
colorbarres % Coloreja els elements gràfics segons els botons de la barra.
clicable % Activa o desactiva la pulsaio dels elements grafics segons els botons de la barra.
%disp(['/Modificar barra ',num2str(n),': ',num2str(toc(tic_mb))])
end

function crearmodificarcarregabarra(n)
% Crea i modifica càrregues a les barres.

barres(n)=dlgcarregabarra(barres(n)); % Crida al diàleg d'edició de les càrregues sobre la
barra.
%disp(['Crear o modificar carregues en barra ',num2str(n)])
%tic_cc=tic;
maxpuntual % Calcula carrega puntual maxima.
maxrepartida % Calcula carrega repartida o projectada maxima.
redibuirar_carregues_barra(n) % Dibuixa les càrregues.
colorcarregabarres
clicable % Estableix els colors i quins objectes gràfics responen als clics segons l'estat dels
botons de la barra d'eines.
%disp(['/Crear o modificar carregues en barra ',num2str(n),': ',num2str(toc(tic_cc))])
end

function redibuirar_carregues_barra(n)
% Dibuixa les càrregues a les barres.
% Si existeixen objectes gràfics els elimina i els crea de nou.

%disp(['Redibuirar carregues en barra ',num2str(n)])
%tic_rc=tic;
delete(barres(n).hgpuntual) % Elimina objecte gràfic força puntual.
barres(n).hgpuntual=[];
delete(barres(n).hgmoment) % Elimina objecte gràfic moment.
barres(n).hgmoment=[];
delete(barres(n).hgrepartida) % Elimina objecte gràfic càrrega repartida.
barres(n).hgrepartida=[];
delete(barres(n).hgprojectada) % Elimina objecte gràfic càrrega projectada.
barres(n).hgprojectada=[];
cordinf=nusos(barres(n).nusinf).posicio; % Coordenades nus inferior.
cordsup=nusos(barres(n).nussup).posicio; % Coordenades nus superior.
vectorbarra=cordsup-cordinf; % Vector direcció de la barra.
vectorbarra=vectorbarra/norm(vectorbarra); % Vector direcció de la barra normalitzat.

if barres(n).puntual(1) || barres(n).moment(1) % Si s'han introduït càrregues puntuals o
moments...
[barres(n).hgpuntual,barres(n).hgmoment]=dibuix_carregabarra(barres(n),nusos(barres(n).
nussup).posicio-nusos(barres(n).nusinf).posicio,vista,fontcarregues,tamanyfontcarregues,fmax); %
Dibuixa carrega puntual i moment.
if barres(n).puntual(1) % Si s'ha introduït càrrega puntual...
pos=barres(n).puntual(5)*vectorbarra+cordinf; % Posició en eixos globals de la càrrega
puntual.
m=makehgtform('translate',pos,'scale',s); % Matriu de col·locació i escala de la
càrrega puntual.
set(barres(n).hgpuntual,'ButtonDownFcn',@clic_carregabarra,'Matrix',m,'Parent',
grafica_est); % Aplica la matriu a l'objecte gràfic que representa la càrrega puntual.
end
if barres(n).moment(1) % Si s'ha introduït moment...
pos=barres(n).moment(5)*vectorbarra+cordinf; % Posició en eixos globals del moment.
m=makehgtform('translate',pos,'scale',s); % Matriu de col·locació i escala del moment.
set(barres(n).hgmoment,'ButtonDownFcn',@clic_carregabarra,'Matrix',m,'Parent',
grafica_est); % Aplica la matriu a l'objecte gràfic que representa el moment.
end
end

if barres(n).repartida(1) && any(cordsup-cordinf) % Si s'ha introduït càrrega repartida...
puntinf=cordinf+vectorbarra*barres(n).repartida(5); % Punt inferior d'aplicació de la
càrrega en eixos globals.
puntsup=cordsup-vectorbarra*barres(n).repartida(6); % Punt superior d'aplicació de la
càrrega en eixos globals.
barres(n).hgrepartida=dibuix_repartida(n,puntinf,puntsup,barres(n).repartida,s,
fontcarregues,tamanyfontcarregues,qmax); % Dibuixa la càrrega repartida.
set(barres(n).hgrepartida,'ButtonDownFcn',@clic_carregabarra,'Parent',grafica_est); %
Assigna la funció que s'executarà quan es cliqueji sobre la càrrega.
if strcmp(vista,'3d')
set(get(barres(n).hgrepartida,'Children'),'Clipping','off') % Fa que els objectes
gràfics sobresurtin de la grafica en vista 3d.
else
set(get(barres(n).hgrepartida,'Children'),'Clipping','on') % Fa que els objectes
gràfics NO sobresurtin de la grafica en vista 3d.
end
end
end

```

```

    if barres(n).projectada(1) && any(cordsup-cordinf) % Si s'ha introduït càrrega projectada...
        puntinf=cordinf+vectorbarra*barres(n).projectada(5); % Punt inferior d'aplicació de la
càrrega en eixos globals.
        puntsup=cordsup-vectorbarra*barres(n).projectada(6); % Punt superior d'aplicació de la
càrrega en eixos globals.
        barres(n).hgprojectada=dibuix_projectada(n,puntinf,puntsup,barres(n).projectada,s,
fontcarregues,tamanyfontcarregues,qmax); % Dibuixa la càrrega projectada.
        set(barres(n).hgprojectada,'ButtonDownFcn',@clic_carregabarra,'Parent',grafica_est); %
Assigna la funció que s'executarà quan es cliqueji sobre la càrrega.
        if strcmp(vista,'3d')
            set(get(barres(n).hgprojectada,'Children'),'Clipping','off') % Fa que els objectes
gràfics sobresurtin de la grafica en vista 3d.
        end
    end
    %disp(['/Redibuixar carregues en barra ',num2str(n),': ',num2str(toc(tic_rc))])
end

function redibuixar_carregues_barres
%disp('Redibuixar carrega barres')
%tic_rcb=tic;
for i=1:contbarra
    redibuixar_carregues_barra(i)
end
colorcarregabarres
diagcarregues
%disp(['/Redibuixar carrega barres: ',num2str(toc(tic_rcb))])
end

function esborrarcarregabarra(n)
% Elimina els objectes gràfics que representen les càrregues sobre
% la barra. Restableix els valors per defecte a la llista de
% barres.

%disp(['/Esborrar carregues en barra ',num2str(n)])
%tic_ec=tic;
if barres(n).hgpuntual % Si hi ha un objecte que representa la força puntual...
    delete(barres(n).hgpuntual) % Elimina l'objecte.
    barres(n).hgpuntual=[]; % Elimina la seva referència.
    barres(n).puntual=[0,0,-1,0,0]; % Valor per defecte.
    maxpuntual % Calcula carrega puntual maxima
end

if barres(n).hgmoment % Si hi ha un objecte que representa el moment...
    delete(barres(n).hgmoment) % Elimina l'objecte.
    barres(n).hgmoment=[]; % Elimina la seva referència.
    barres(n).moment=[0,0,0,1,0]; % Valor per defecte.
end

if barres(n).hgrepartida % Si hi ha un objecte que representa la càrrega repartida...
    delete(barres(n).hgrepartida) % Elimina l'objecte.
    barres(n).hgrepartida=[]; % Elimina la seva referència.
    barres(n).repartida=[0,0,-1,0,0,0]; % Valor per defecte.
    maxrepartida % Calcula carrega repartida o projectada maxima.
end

if barres(n).hgprojectada % Si hi ha un objecte que representa la càrrega projectada...
    delete(barres(n).hgprojectada) % Elimina l'objecte.
    barres(n).hgprojectada=[]; % Elimina la seva referència.
    barres(n).projectada=[0,0,-1,0,0,0]; % Valor per defecte.
    maxrepartida % Calcula carrega repartida o projectada maxima.
end

%disp(['/Esborrar carregues en barra ',num2str(n),': ',num2str(toc(tic_ec))])
end

function maxpuntual
% Calcula la màxima força puntual o aplicada als nusos.

fmax=0; % Valor per defecte cero.
for i=1:contnus % Per a tots els nusos...
    fmax=max(abs([fmax,nusos(i).puntual])); % Màxim de les forces puntuals.
end
for i=1:contbarra % Per a totes les barres...
    fmax=max(abs([fmax,barres(i).puntual(1)])); % Màxim de les forces puntuals.
end
if flagcalculat % Si hi han reaccions...
    for i=1:contnus
        fmax=max(abs([fmax,Retorn_reaccions(i*6-5:i*6-3)]));
    end
end
end

```

```

end

function maxrepartida
    % Calcula la maxima carrega repartida o projectada.

    qmax=0; % Valor inicial cero.
    for i=1:contbarra % Per a totes les barres.
        qmax=max(abs([qmax,barres(i).repartida(1),barres(i).projectada(1)])); % Màxim de les forces
repartides i projectades.
    end
end

end

%% FUNCIONS DE CARREGA FORA DE LA FUNCIO PRINCIPAL

function [nusus,barres,contnus,contbarra]=carregararxiu(arxiu)
    % Carrega un arxiu amb dades de l'estructura i les retorna.
    % Aquesta funció no pot estar dins de la funció principal del programa,
    % Matlab no permet carregar les variables a l'espai de treball, ha
    % d'estar aïllada.

    load(arxiu,'-mat') % Carrega les dades de nusos i barres.
    contnus=numel(nusus); % Nombre de nusos.
    contbarra=numel(barres); % Nombre de barres.
end
function [editalcrear,barraalcrearnus,fontnusus,tamanyfontnusus,fontcarregues,tamanyfontcarregues,
factorzoom,tamanyynus,distnumnus,precini,XLimini,YLimini,ZLimini,colormodificar,coloresborrar,
colorcarregues,colorreaccions,colordeformada,colorN,colorTy,colorTz,colorMx,colorMy,colorMz,direct]
=loadopcions
    % Carrega les variables amb les opcions del programa i les retorna.
    % Si no troba l'arxiu crida al diàleg per editar les opcions. El diàleg
    % no permet cancel·lar sense guardarles a l'arxiu. Finalment carrega les
    % dades.
    % Aquesta funció no pot estar dins de la funció principal del programa,
    % Matlab no permet carregar les variables a l'espai de treball, ha
    % d'estar aïllada.

    if ~exist('opcions.mat','file') % Si no existeix l'arxiu d'opcions del programa obre la finestra
d'edició de les opcions. No es pot sortir de la finestra d'opcions sense crear l'arxiu.
        dlgopcions % Crida al diàleg d'edició de les opcions.
    end
    load('opcions.mat') % Carrega les opcions.
end
end

```



```

%% CÀLCUL ESTÀTIC PER A ESTRUCTURES BI O TRIDIMENSIONALS.
%
% SERGIO GUERRERO MIRALLES 2.011-2.012
% ENGINYERIA TÈCNICA D'OBRES PÚBLIQUES
% UNIVERSITAT POLITÈCNICA DE CATALUNYA
%
% Versió data 13/10/2.012
%
% Entrada:
% n = sencer nombre de nusos de l'estructura.
% N = sencer nombre de barres de l'estructura.
% Ncoord = matriu n*3 amb les coordenades x, y i z dels nusos.
% Ncond = condicions de contorn dels nusos: nus lliure =0; rodet inclinat
% =1; recolzament simple =2; encastament =3; desplaçaments elàstics i girs
% lliures =4; girs elàstics i desplaçaments nuls =5; desplaçaments i girs
% elàstics =6; desplaçaments imposats i girs lliures =7; girs imposats i
% desplaçaments lliures =8; desplaçaments i girs imposats =9.
% Nextfor = vector n*6 de forces aplicades als nusos.
% Lnodes = matriu N*2 amb nombre de nus inferior i superior de cada barra.
% Lartyp = matriu N*2 amb tipus d'extrem de barra inferior i superior,
% rígid =0; articulad =1.
% Lmecp = matriu N*9 de propietats mecàniques de cada barra: E, G, A, Iz,
% Iy, zx, zy, zz, J.
% barra_carrega_rep = matriu N*6 amb la càrrega repartida sobre les barres,
% mòdul, direcció x, y i z, distància a extrem inferior i superior.
% barra_carrega_rep_proj = matriu N*6 amb la càrrega repartida projectada
% sobre les barres (longitud de la càrrega = longitud de la barra), mòdul,
% direcció x, y i z, distància a extrem inferior i superior.
% barra_carrega_punt = matriu N*5 amb la càrrega puntual sobre les barres,
% mòdul, direcció x, y i z i distància al nus inferior.
% barra_carrega_mom = matriu N*5 amb el moment aplicat sobre les barres,
% mòdul, direcció x, y i z i distància al nus inferior.
%
% Sortida:
% Retorn_reaccions = vector n*6 amb les reaccions als nusos.
% Retorn_moviments = vector n*6 amb desplaçaments i girs.
% Retorn_esforços_barres = matriu N*12 amb els esforços als extrems de les
% barres.
% TL2G = matriu 3*3*N amb les matrius de canvi de base local a global de
% cada barra.
%
% Discrimina si el sistema és en 2 o 3 dimensions, eliminant dirèctament
% els moviments que no hi intervenen en vigues contínues, engrallats
% plans, girs de ròtula, etc.
% Reordena els nusos de manera que els recolzaments quedin al final,
% d'aquesta manera la matriu reduïda queda inscrita dins de la matriu
% completa i no cal duplicar-la a la memòria.
% sistema: llista amb l'ordre dels nusos a la matriu de rigidesa global.
% ns: dimensió de la matriu del sistema.
% nn: dimensió de la matriu reduïda. nn<ns.

function [Retorn_reaccions,Retorn_moviments,Retorn_esforços_barres,TL2G] = cme2_calcul_estatic(n,N,
Ncoord,Ncond,Nextfor,Lnodes,Lartyp,Lmecp,barra_carrega_rep,barra_carrega_rep_proj,barra_carrega_punt,
barra_carrega_mom)
[sistema,ns,nn]=reordenacio_nusos(Lnodes,Lartyp,barra_carrega_rep,barra_carrega_rep_proj,
barra_carrega_punt,barra_carrega_mom,Nextfor,Ncoord,Ncond,N,n); % Reordenació dels nusos per inscriure
la Matriu Reduïda dins de la Matriu de Rigidesa.
Matriu_Rigidesa_G=zeros(ns,ns); % Dimensiona la matriu de rigidesa global.
Nintfor=zeros(n*6,1); % Dimensiona vector d'esforços transmesos als nusos per les càrregues aplicades a
les barres.
Nepfor=zeros(N,12); % Matriu amb esforços en extrems de barra deguts a les càrregues, per a cada barra.
TL2G=zeros(3,3,N);
for i=1:N % Per a cada barra...
n1=Lnodes(i,1); % node inferior
n2=Lnodes(i,2); % node superior
extrems=Lartyp(i,1)*2+Lartyp(i,2); % rígid-rígid =0, rígid-articulat =1, articulad-rígid =2,
articulat-articulat =3.
vector_barra=Ncoord(n2,1:3)-Ncoord(n1,1:3); % Vector de la barra.
L=norm(vector_barra); % Longitud de la barra.
vector_Iz=Lmecp(i,6:8); % Vector inèrcia de la barra.
TL2G(:, :, i)=calcul_canvi_base(vector_barra,vector_Iz); % Càlcul i enmagatzematge de la matriu de
canvi de base d'eixos locals a globals.
T=[TL2G(:, :, i),zeros(3);zeros(3),TL2G(:, :, i)]; % Matriu canvi de base completa 6x6
KL=calcul_submatrius_rigidesa(Lmecp(i,:),extrems,L); % Càlcul de la submatriu de rigidesa en eixos
locals.
KG=[T*KL(1:6,1:6)*T',T*KL(1:6,7:12)*T';T*KL(7:12,1:6)*T',T*KL(7:12,7:12)*T']; % Submatriu de
rigidesa en eixos globals.
KG=contorn_rodet_inclinat(KG,n1,n2,Ncond); % Aplicació de la condició de contorn de rodet inclinat
a la submatriu de rigidesa, si s'escau.
Matriu_Rigidesa_G=Matriu_Rigidesa_G+ensamblatge_matriu_rigidesa(KG,ns,sistema,n1,n2); % Ensamblatge
de la submatriu dins de la Matriu de Rigidesa.
Nepfor(i,:)=calcul_càrregues_barra_local(barra_carrega_rep(i,:),barra_carrega_rep_proj(i,:),

```

```

barra_carrega_punt(i,:),barra_carrega_mom(i,:),TL2G(:, :, i),L,extrems); % Trasllat de les carregues a
les barres als extrems de barra, en eixos locals.
Nintfor(n1*6-5:n1*6)=Nintfor(n1*6-5:n1*6)+T*Nepfor(i,1:6)'; % Càlcul de l'estat de forces al nus
inferior provocades per les càrregues a les barres, en eixos globals.
Nintfor(n2*6-5:n2*6)=Nintfor(n2*6-5:n2*6)+T*Nepfor(i,7:12)'; % Càlcul de l'estat de forces al nus
superior provocades per les càrregues a les barres, en eixos globals.
end
Ntotfor=Nextfor+Nintfor; % Total de forces aplicades als nusos.
[Vector_esforços_prescrits,Vector_esforços,Vector_moviments]=contorn_prescrits_i_rodet
(Matriu_Rigidesa_G,n,ns,sistema,Ncond,Ntotfor); % Crea els vectors d'esforços i moviments aplicant les
condicions de contorn de moviments prescrits i rodets.
Matriu_Rigidesa_G=contorn_elastic(Matriu_Rigidesa_G,n,sistema,Ncond(:,1),Ncond(:,2:7)); % Modifica la
Matriu de Rigidesa amb les rigideses dels contorns elàstics.
Vector_moviments(1:nn,1)=Matriu_Rigidesa_G(1:nn,1:nn)\Vector_esforços_prescrits(1:nn,1); % Resol el
sistema amb la matriu reduïda amb els esforços calculats per als contorns prescrits sumats.
Matriu_Rigidesa_G=contorn_elastic(Matriu_Rigidesa_G,n,sistema,Ncond(:,1),-Ncond(:,2:7)); % Desfà la
modificació de la Matriu de Rigidesa amb les rigideses dels contorns elàstics.
Vector_reaccions=Matriu_Rigidesa_G*Vector_moviments-Vector_esforços; % Calcula les reaccions als
recolzaments de l'estructura.
[Retorn_reaccions,Retorn_moviments]=ordre_inicial_nusos(sistema,Vector_reaccions,Vector_moviments,n,
Ncond); % Retorn de reaccions i moviments en l'ordre original dels nusos i eixos globals.
Retorn_esforços_barres=zeros(N,12); % Càlcul dels esforços en extrems de barra.
for i=1:N
n1=Lnodes(i,1); % node inferior
n2=Lnodes(i,2); % node superior
extrems=Lartyp(i,1)*2+Lartyp(i,2); % rígid-rígid =0, rígid-articulat =1, articulat-rígid =2,
articulat-articulat =3.
vector_barra=Ncoord(n2,1:3)-Ncoord(n1,1:3); % Vector de la barra.
L=norm(vector_barra); % Longitud de la barra.
T=[TL2G(:, :, i),zeros(3);zeros(3),TL2G(:, :, i)]; % Matriu canvi de base completa 6x6
KL=calcul_submatriu_rigidesa(Lmecp(i,:),extrems,L); % Càlcul de la submatriu de rigidesa en eixos
locals.
vector_mov_local=[T'*Retorn_moviments(n1*6-5:n1*6);T'*Retorn_moviments(n2*6-5:n2*6)]; % Vector
moviments en eixos locals.
Retorn_esforços_barres(i,:)=KL*vector_mov_local-Nepfor(i,:)'; % Esforços en extrems de cada barra
en eixos locals.
end
end

%% REORDENACIÓ DE NUSOS.

% CRIVATGE I REORDENACIÓ D'ESFORÇOS/MOVIMENTS ALS NUSOS PER A LA
% CONSTRUCCIÓ DE LA MATRIU DE RIGIDESA GLOBAL.
% Elimina girs de ròtula i la tercera dimensió si la estructura és plana,
% agrupa els moviments prescrits per les condicions de contorn per tenir
% inscrita la Matriu Reduïda dins de la Global.
% Genera la llista 'sistema' amb dimensió 6 * número de nusos amb l'ordre a la
% matriu de rigidesa global.

function [sistema,ns,nn]=reordenacio_nusos(Lnodes,Lartyp,carrega_rep,carrega_rep_proj,carrega_punt,
carrega_mom,Nextfor,Ncoord,Ncond,N,n)
moviment_sistema=ones(6*n,1); % moviments per defecte tots permesos.
nus_rotula=zeros(6*n,1);
forces=[0,0,0];
moments=[0,0,0];
for i=1:N % Per a totes les barres...
n1=6*Lnodes(i,1);
n2=6*Lnodes(i,2);
if Lartyp(i,1)==0 % Si l'extrem inferior és rígid...
nus_rotula(n1-5:n1)=[1,1,1,1,1,1]; % es tindran en compte els girs en x, y i z del nus
inferior.
else
nus_rotula(n1-5:n1-3)=[1,1,1]; %si tots els extrems que arriben al nus son articulats no es
tindrà en compte a la Matriu de Rigidesa.
end
if Lartyp(i,2)==0 % Si l'extrem superior és rígid...
nus_rotula(n2-5:n2)=[1,1,1,1,1,1]; % es tindran en compte els girs en x, y i z del nus
inferior.
else
nus_rotula(n2-5:n2-3)=[1,1,1]; %si tots els extrems que arriben al nus son articulats no es
tindrà en compte a la Matriu de Rigidesa.
end
forces=forces|(carrega_rep(i,1)*carrega_rep(i,2:4)|carrega_rep_proj(i,1)*carrega_rep_proj(i,2:4)
|carrega_punt(i,1)*carrega_punt(i,2:4); % Direcció forces aplicades a les barres.
moments=moments|carrega_mom(i,1)*carrega_mom(i,2:4); % Direcció moments aplicats a les barres.
end
for i=1:n % Per a tots els nusos...
forces=forces|Nextfor(6*i-5:6*i-3)'; % Direcció forces a tota l'estructura.
moments=moments|Nextfor(6*i-2:6*i)'; % Direcció moments a tota l'estructura.
if Ncond(i,1)==1 % Rodet inclinat.
forces=forces|Ncond(i,2:4); % El rodet inclinat introdueix reaccions normals al seu pla.

```

```

end
if Ncond(i,1)==7 || Ncond(i,1)==9 % Desplaçaments imposats.
    forces=forces|Ncond(i,2:4);
end
if Ncond(i,1)==8 || Ncond(i,1)==9 % Girs imposats.
    moments=moments|Ncond(i,5:7);
end
end
dimx=any(Ncoord(:,1)~=Ncoord(1,1)); % Si tots els nusos tenen la mateixa coordenada x dimx=0.
dimy=any(Ncoord(:,2)~=Ncoord(1,2)); % Si tots els nusos tenen la mateixa coordenada y dimy=0.
dimz=any(Ncoord(:,3)~=Ncoord(1,3)); % Si tots els nusos tenen la mateixa coordenada z dimz=0.
moviments=[0,0,0,0,0,0];
if dimx && ~dimy && ~dimz % Biga en eix X.
    if forces(1) % Forces en sentit longitudinal eix X.
        moviments=[1,0,0,0,0,0]; % Només forces i moviments en eix X.
    end
    if moments(1) % Moments en eix X.
        moviments=[0,0,0,1,0,0];
    end
    if forces(2) || moments(3) % Forces transversals en eix Y.
        moviments=moviments | [0,1,0,0,0,1]; % Forces i moviments en eix Y i moments i girs en eix Z.
    end
    if forces(3) || moments(2) % Forces transversals en eix Z.
        moviments=moviments | [0,0,1,0,1,0]; % Forces i moviments en eix Z i moments i girs en eix Y.
    end
elseif ~dimx && dimy && ~dimz % Biga en eix Y.
    if forces(2) % Forces en sentit longitudinal eix Y.
        moviments=[0,1,0,0,0,0]; % Només forces i moviments en eix Y.
    end
    if moments(2) % Moments en eix Y.
        moviments=[0,0,0,0,1,0];
    end
    if forces(1) || moments(3) % Forces transversals en eix X.
        moviments=moviments | [1,0,0,0,0,1]; % Forces i moviments en eix X i moments i girs en eix Z.
    end
    if forces(3) || moments(1) % Forces transversals en eix Z.
        moviments=moviments | [0,0,1,1,0,0]; % Forces i moviments en eix Z i moments i girs en eix X.
    end
elseif ~dimx && ~dimy && dimz % Biga en eix Z.
    if forces(3) % Forces en sentit longitudinal eix Z.
        moviments=[0,0,1,0,0,0]; % Només forces i moviments en eix Z.
    end
    if moments(3) % Moments en eix Z.
        moviments=[0,0,0,0,0,1];
    end
    if forces(1) || moments(2) % Forces transversals en eix X.
        moviments=moviments | [1,0,0,0,1,0]; % Forces i moviments en eix X i moments i girs en eix Y.
    end
    if forces(2) || moments(1) % Forces transversals en eix Y.
        moviments=moviments | [0,1,0,1,0,0]; % Forces i moviments en eix Y i moments i girs en eix X.
    end
elseif dimx && dimy && ~dimz % Estructura plana XY.
    if ~forces(3) && ~moments(1) && ~moments(2) % Forces en el pla de l'estructura -> 2D.
        moviments=[1,1,0,0,0,1]; % Forces i moviments en eixos X i Y i moments i girs en eix Z.
    elseif ~forces(1) && ~forces(2) && ~moments(3) % Engraellat pla XY.
        moviments=[0,0,1,1,1,0]; % Forces i moviments en eix Z i moments i girs en eixos X i Y.
    else
        moviments=[1,1,1,1,1,1]; % Estructura 3D.
    end
elseif ~dimx && dimy && dimz % Estructura plana YZ.
    if ~forces(1) && ~moments(2) && ~moments(3) % Forces en el pla de l'estructura -> 2D.
        moviments=[0,1,1,1,0,0]; % Forces i moviments en eixos Y i Z i moments i girs en eix X.
    elseif ~forces(2) && ~forces(3) && ~moments(1) % Engraellat pla YZ.
        moviments=[1,0,0,0,1,1]; % Forces i moviments en eix X i moments i girs en eixos Y i Z.
    else
        moviments=[1,1,1,1,1,1]; % Estructura 3D.
    end
elseif dimx && ~dimy && dimz % Estructura plana XZ.
    if ~forces(2) && ~moments(1) && ~moments(3) % Forces en el pla de l'estructura -> 2D.
        moviments=[1,0,1,0,1,0]; % Forces i moviments en eixos X i Z i moments i girs en eix Y.
    elseif ~forces(1) && ~forces(3) && ~moments(2) % Engraellat pla XZ.
        moviments=[0,1,0,1,0,1]; % Forces i moviments en eix Y i moments i girs en eixos X i Z.
    else
        moviments=[1,1,1,1,1,1]; % Estructura 3D.
    end
else
    moviments=[1,1,1,1,1,1]; % Estructura 3D.
end
contorn=zeros(6*n,1);
for i=1:n % Per a tots els nusos, aplica les condicions de contorn.
    moviment_sistema(6*i-5:6*i)=moviments;
    switch Ncond(i,1)

```

```

    case 0 % Nus lliure
        contorn(6*i-5:6*i)=[1,1,1,1,1,1]; % No hi ha condició de contorn, no s'anula cap
desplaçament o gir.
    case 1 % Rodet inclinat
        contorn(6*i-5:6*i)=[0,1,1,1,1,1]; % S'anula el desplaçament en x (correspon al vector
perpendicular al pla del rodet als eixos locals del rodet).
        Tr=calcul_canvi_base(Ncond(i,2:4)); % Calcula matriu canvi de base del rodet.
        T=[Tr,zeros(3);zeros(3),Tr];
        moviment_sistema(6*i-5:6*i)=T*moviment_sistema(6*i-5:6*i);

    case 2 % Recolzament simple
        contorn(6*i-5:6*i)=[0,0,0,1,1,1]; % S'anulen els moviments.
    case 3 % Encastament
        % contorn(6*i-5:6*i)=[0,0,0,0,0,0]; % S'anulen tots els moviments i girs.
    case 4 % Desplaçaments elàstics i girs lliures
        contorn(6*i-5:6*i)=[1,1,1,1,1,1]; % No s'anula cap desplaçament o gir.
    case 5 % Girs elàstics i desplaçaments nuls
        contorn(6*i-5:6*i)=[0,0,0,1,1,1]; % S'anulen els desplaçaments.
    case 6 % Desplaçaments i girs elàstics
        contorn(6*i-5:6*i)=[1,1,1,1,1,1]; % No s'anula cap desplaçament o gir.
    case 7 % Desplaçaments imposats i girs lliures
        contorn(6*i-5:6*i)=[0,0,0,1,1,1]; % S'anulen els desplaçaments.
    case 8 % Girs imposats i desplaçaments lliures
        contorn(6*i-5:6*i)=[1,1,1,0,0,0]; % S'anulen els girs.
    case 9 % Desplaçaments i girs imposats
        %contorn(6*i-5:6*i)=[0,0,0,0,0,0]; % S'anulen tots els moviments i girs.
end
end
moviment_sistema=moviment_sistema & nus_rotula; % Vector on per a cada moviment o gir s'indica si es
tindra en compte o no en la construcció de la Matriu de Rigidesa.
contorn=contorn & moviment_sistema; % Vector on per a cada moviment o gir s'indica si s'anula per les
condicions de contorn.
ns=sum(moviment_sistema);
nn=sum(contorn);
sistema=zeros(6*n,1);
j=1;
k=nn+1;
for i=1:6*n % Crea el vector sistema on s'emmagatzema en cada posició el nus corresponent ordenats per
construir la Matriu de Rigidesa.
    if moviment_sistema(i)
        if contorn(i)
            sistema(i)=j;
            j=j+1;
        else
            sistema(i)=k;
            k=k+1;
        end
    end
end
end
end

%% CÀLCUL DE LA MATRIU DE CANVI DE BASE D'EIXOS LOCALS A GLOBALS.

function matriu_canvi_base=calcul_canvi_base(vector1,varargin)
vector1=vector1/norm(vector1);
if nargin==2 && any(varargin{1}) && (vector1*varargin{1}'==0) % Si s'ha introduït un segon vector,
corresponent al vector de l'eix d'inèrcia Z i perpendicular a la barra...
    vector3=varargin{1};
    vector3=vector3/norm(vector3);
    vector2=cross(vector3,vector1);
    vector2=vector2/norm(vector2);
elseif abs(vector1(3))==1 % Si el primer vector és paral·lel a l'eix z fa servir els altres dos eixos
per crear la matriu de canvi de base.
    vector3=[vector1(3),0,0];
    vector2=[0,1,0];
else
    vector2=cross([0,0,1],vector1); % Calcula dos vectors perpendiculars al primer i entre ells per
crear la matriu de canvi de base.
    vector2=vector2/norm(vector2);
    vector3=cross(vector1,vector2);
    vector3=vector3/norm(vector3);
end
matriu_canvi_base=[vector1',vector2',vector3'];
end

%% SUBMATRIUS DE REGIDESA EN EIXOS LOCALS.

function KL=calcul_submatrius_rigidesa(Lmecp,extrems,L)
E=Lmecp(1); % Mòdul de Young
Iz=Lmecp(4); % Inèrcia eix z
if any(Lmecp(6:8)) % Si existeix vector direcció d'Iz

```

```

Iy=Lmecip(5); % Inèrcia eix y
else
Iy=Iz;
end
GITL=Lmecip(9)/L; % Inèrcia a torsió
A=Lmecip(3); % Àrea de la secció
EAL=E*A/L;
EIZL=E*Iz/L;
EIYL=E*Iy/L;
EIZL2=E*Iz/L^2;
EIYL2=E*Iy/L^2;
EIZL3=E*Iz/L^3;
EIYL3=E*Iy/L^3;
% Sub-matrius de rigidesa segons els extrems siguin rígids o articulats.
switch extrems
case 0 % rígid-rígid
KL(1:6,1:6)=[EAL,0,0,0,0,0;0,12*EIZL3,0,0,0,6*EIZL2;0,0,12*EIYL3,0,-6*EIYL2,0;0,0,0,GITL,0,0;
0,0,-6*EIYL2,0,4*EIYL,0;0,6*EIZL2,0,0,0,4*EIZL];
KL(1:6,7:12)=[-EAL,0,0,0,0,0;0,-12*EIZL3,0,0,0,6*EIZL2;0,0,-12*EIYL3,0,-6*EIYL2,0;0,0,0,-GITL,
0,0;0,0,6*EIYL2,0,2*EIYL,0;0,-6*EIZL2,0,0,0,2*EIZL];
KL(7:12,1:6)=KL(1:6,7:12)';
KL(7:12,7:12)=[EAL,0,0,0,0,0;0,12*EIZL3,0,0,0,-6*EIZL2;0,0,12*EIYL3,0,6*EIYL2,0;0,0,0,GITL,0,0;
0,0,6*EIYL2,0,4*EIYL,0;0,-6*EIZL2,0,0,0,4*EIZL];
%KL(1:6,1:6)=[EAL,0,0,0,0,0;0,12*EIZL3,0,0,0,6*EIZL2;0,0,12*EIYL3,0,6*EIYL2,0;0,0,0,GITL,0,0;
0,0,6*EIYL2,0,4*EIYL,0;0,6*EIZL2,0,0,0,4*EIZL];
%KL(1:6,7:12)=[-EAL,0,0,0,0,0;0,-12*EIZL3,0,0,0,6*EIZL2;0,0,-12*EIYL3,0,6*EIYL2,0;0,0,0,-GITL,
0,0;0,0,-6*EIYL2,0,2*EIYL,0;0,-6*EIZL2,0,0,0,2*EIZL];
%KL(7:12,1:6)=KL(1:6,7:12)';
%KL(7:12,7:12)=[EAL,0,0,0,0,0;0,12*EIZL3,0,0,0,-6*EIZL2;0,0,12*EIYL3,0,-6*EIYL2,0;0,0,0,GITL,
0,0;0,0,-6*EIYL2,0,4*EIYL,0;0,-6*EIZL2,0,0,0,4*EIZL];
case 1 % rígid-articulat
KL(1:6,1:6)=[EAL,0,0,0,0,0;0,3*EIZL3,0,0,0,3*EIZL2;0,0,3*EIYL3,0,-3*EIYL2,0;0,0,0,0,0,0;0,0,-3
*EIYL2,0,3*EIYL,0;0,3*EIZL2,0,0,0,3*EIZL];
KL(1:6,7:12)=[[-EAL,0,0,0,0,0,-3*EIZL3,0;0,0,-3*EIYL3;0,0,0,0,3*EIYL2;0,-3*EIZL2,0],zeros(6,3)];
KL(7:12,1:6)=KL(1:6,7:12)';
KL(7:12,7:12)=[EAL,0,0,0,0,0,3*EIZL3,0;0,0,3*EIYL3],zeros(3,3);zeros(3,6)];
case 2 % articulad-rígid
KL(1:6,1:6)=[EAL,0,0,0,0,0,3*EIZL3,0;0,0,3*EIYL3],zeros(3);zeros(3,6)];
KL(1:6,7:12)=[[-EAL,0,0,0,0,0,0,-3*EIZL3,0,0,0,3*EIZL2;0,0,-3*EIYL3,0,-3*EIYL2,0];zeros(3,6)];
KL(7:12,1:6)=KL(1:6,7:12)';
KL(7:12,7:12)=[EAL,0,0,0,0,0,0,3*EIZL3,0,0,0,-3*EIZL2;0,0,3*EIYL3,0,3*EIYL2,0;0,0,0,0,0,0;0,0,3
*EIYL2,0,3*EIYL,0,0,-3*EIZL2,0,0,0,3*EIZL];
case 3 % articulad-articulat
KL(1:6,1:6)=[EAL,0,0,0,0,0;zeros(5,6)];
KL(1:6,7:12)=[-EAL,0,0,0,0,0;zeros(5,6)];
KL(7:12,1:6)=KL(1:6,7:12);
KL(7:12,7:12)=KL(1:6,1:6);
end
end

%% APLICACIÓ CONDICIÓ DE CONTORN RODET INCLINAT A LES SUBMATRIUS DE RIGIDESA.

% Condició de contorn rodet inclinat.
% Si algú dels extrems connecta amb un rodet inclinat, multiplica les submatrius per la matriu de
canvi de base als eixos del rodet, els moviments i esforços del nus corresponent queden en eixos locals
del rodet.

function KG=contorn_rodet_inclinat(KG,n1,n2,Ncond)
if Ncond(n1,1)==1 % Si el nus inferior és un rodet inclinat
Tr=calcul_canvi_base(Ncond(n1,2:4)); % Calcula matriu canvi de base del rodet.
T=[Tr,zeros(3);zeros(3),Tr];
KG(1:6,1:6)=T'*KG(1:6,1:6)*T; % Sub-matriu en la base del rodet.
KG(1:6,7:12)=T'*KG(1:6,7:12); % Sub-matriu en la base del rodet.
KG(7:12,1:6)=KG(7:12,1:6)*T; % Sub-matriu en la base del rodet.
end
if Ncond(n2,1)==1 % Si el nus superior és un rodet inclinat
Tr=calcul_canvi_base(Ncond(n2,2:4)); % Calcula matriu canvi de base del rodet.
T=[Tr,zeros(3);zeros(3),Tr];
KG(1:6,7:12)=KG(1:6,7:12)*T; % Sub-matriu en la base del rodet.
KG(7:12,1:6)=T'*KG(7:12,1:6); % Sub-matriu en la base del rodet.
KG(7:12,7:12)=T'*KG(7:12,7:12)*T; % Sub-matriu en la base del rodet.
end
end

%% ENSAMBLATGE DE LA MATRIU DE RIGIDESA GLOBAL.

% Ensamblatge de la Matriu de Rigidesa Global.
% S'elimina directament tercera dimensió i girs de ròtules si és el cas.
% Primer comprova la fila i dins de la fila les col·lumnas que s'han d'afegir a la matriu.

```

```

function Matriu_Rigidesa_G=ensamblatge_matriu_rigidesa(KG,ns,sistema,n1,n2)
Matriu_Rigidesa_G=zeros(ns,ns);
for l=1:6
    if sistema(6*n1-6+l)
        j=sistema(6*n1-6+l); % Índex dins de la matriu
        for m=1:6
            if sistema(6*n1-6+m)
                k=sistema(6*n1-6+m); % Índex dins de la matriu
                Matriu_Rigidesa_G(j,k)=KG(l,m);
            end
            if sistema(6*n2-6+m)
                k=sistema(6*n2-6+m); % Índex dins de la matriu
                Matriu_Rigidesa_G(j,k)=KG(l,m+6);
            end
        end
    end
    if sistema(6*n2-6+l)
        j=sistema(6*n2-6+l); % Índex dins de la matriu
        for m=1:6
            if sistema(6*n1-6+m)
                k=sistema(6*n1-6+m); % Índex dins de la matriu
                Matriu_Rigidesa_G(j,k)=KG(l+6,m);
            end
            if sistema(6*n2-6+m)
                k=sistema(6*n2-6+m); % Índex dins de la matriu
                Matriu_Rigidesa_G(j,k)=KG(l+6,m+6);
            end
        end
    end
end
end

%% CÀLCUL DE LES SOL·LICITACIONS ALS NUSOS PER CÀRREGUES A LES BARRES.

% Per a cada tipus d'extrem de barra, rígid o articulad, en calcula els
% esforços d'empotrment perfecte, per a càrregues repartides projectades
% (longitud de la càrrega = longitud de la barra) o no sobre la barra,
% càrregues puntuals i moments aplicats a la bara. Per a càrregues
% repartides permet definir distàncies al nus inferior i superior per
% delimitar la zona de la barra que rep la càrrega. Igualment amb la
% distància al nus inferior es situa la càrrega puntual i el moment a la
% barra. Totes les càrregues s'introdueixen, i els resultats es retornen,
% en eixos globals, però es calculen en eixos locals (eix de la barra
% coincideix amb l'eix X local).

function reaccions=calcul_carregues_barra_local(carrega_rep,carrega_rep_proj,carrega_punt,carrega_mom,
TL2G,L,extrems)
reaccions=[0,0,0,0,0,0,0,0,0,0,0,0];
if carrega_rep_proj(1) % Càlcul de les forces equivalents als nusos a la càrrega repartida projectada
sobre la barra.
    vbarra=[L,0,0]; % Vector barra en eixos locals;
    carrega=TL2G'*carrega_rep_proj(2:4)'; % Vector direcció de la càrrega en eixos locals.
    carrega=carrega_rep_proj(1)*carrega/norm(carrega); % Vector càrrega en eixos locals.
    vperp=cross(vbarra,carrega);
    vperp=cross(vperp,carrega);
    vperp=vperp/norm(vperp);
    Lc=abs(dot(vperp,vbarra)); % Longitud de la càrrega.
    reaccions=-calcul_repartida(carrega,carrega_rep_proj(5),carrega_rep_proj(6),L,Lc,extrems);
end
if carrega_rep(1) % Càlcul de les forces equivalents als nusos a la càrrega repartida a la barra.
    carrega=TL2G'*carrega_rep(2:4)'; % Vector direcció de la càrrega repartida en eixos locals.
    carrega=carrega_rep(1)*carrega'/norm(carrega); % Vector càrrega
    reaccions=reaccions-calcul_repartida(carrega,carrega_rep(5),carrega_rep(6),L,L,extrems);
end
if carrega_punt(1) % Càlcul de les forces equivalents als nusos a la càrrega puntual a la barra.
    carrega=TL2G'*carrega_punt(2:4)'; % Vector direcció de la càrrega puntual
    carrega=carrega'*carrega_punt(1)/norm(carrega); % Vector càrrega.
    reaccions=reaccions-calcul_puntual(carrega,carrega_punt(5),L,extrems);
end
if carrega_mom(1) % Càlcul de les forces equivalents als nusos al moment aplicat a la barra.
    carrega=TL2G'*carrega_mom(2:4)'; % Vector direcció del moment.
    carrega=carrega'*carrega_mom(1)/norm(carrega); % Vector moment.
    reaccions=reaccions-calcul_moment(carrega,carrega_mom(5),L,extrems);
end
end

% FUNCIÓ DE CÀLCUL DE LES REACCIONS PER CÀRREGA REPARTIDA EN EIXOS LOCALS

function reaccions=calcul_repartida(carrega,dist1,dist2,L,Lc,extrems)
c=(L-dist1-dist2)*Lc/(2*L);

```



```

a=dist1*Lc/L+c;
b=dist2*Lc/L+c;
switch extrems
    case 0 % rígid-rígid
        reaccions(1:3)=-carrega*2*c*(1-(3*a^2+c^2)/Lc^2+2*a*((a^2+c^2)/Lc^3)); % Reaccions
d'empotrament als nus inferior.
        reaccions(4:6)=[0,[carrega(3),-carrega(2)]*L*2*c*(a*b^2/Lc^3-c^2*(3*b-Lc)/(3*Lc^3)); % Moments
d'empotrament al nus inferior.
        reaccions(7:9)=-carrega*2*c*((3*a^2+c^2)/Lc^2-2*a*(a^2+c^2)/Lc^3); % Reaccions d'empotrament
als nus superior.
        reaccions(10:12)=[0,[-carrega(3),carrega(2)]*L*2*c*(b*a^2/Lc^3-c^2*(3*a-Lc)/(3*Lc^3)); %
Moments d'empotrament al nus superior.
    case 1 % rígid-articulat
        reaccions(1:3)=-carrega*c*(2-(3*a^2+c^2)/Lc^2+a*(a^2+c^2)/Lc^3); % Reaccions d'empotrament als
nus inferior.
        reaccions(4:6)=[0,[carrega(3),-carrega(2)]*L*b*c*(a*(Lc+b)-c^2)/Lc^3]; % Moments d'empotrament
al nus inferior.
        reaccions(7:9)=-carrega*c*((3*a^2+c^2)/Lc^2-a*(a^2+c^2)/Lc^3); % Reaccions d'empotrament als
nus superior.
        reaccions(10:12)=[0,0,0]; % Moments d'empotrament al nus superior.
    case 2 % articulad-rígid
        reaccions(1:3)=-carrega*c*((3*b^2+c^2)/Lc^2-b*(b^2+c^2)/Lc^3); % Reaccions d'empotrament als
nus inferior.
        reaccions(4:6)=[0,0,0]; % Moments d'empotrament al nus inferior.
        reaccions(7:9)=-carrega*c*(2-(3*b^2+c^2)/Lc^2+b*(b^2+c^2)/Lc^3); % Reaccions d'empotrament als
nus superior.
        reaccions(10:12)=[0,[-carrega(3),carrega(2)]*L*a*c*(b*(Lc+a)-c^2)/Lc^3]; % Moments
d'empotrament al nus superior.
    case 3 % articulad_articulad
        reaccions(1:3)=-carrega*2*c*(Lc-a)/Lc; % Reaccions d'empotrament als nus inferior.
        reaccions(4:6)=[0,0,0]; % Moments d'empotrament al nus inferior.
        reaccions(7:9)=-carrega*2*c*(Lc-b)/Lc; % Reaccions d'empotrament als nus superior.
        reaccions(10:12)=[0,0,0]; % Moments d'empotrament al nus superior.
end
end

```

% FUNCIO DE CÀLCUL DE LES REACCIONS PER CÀRREGA PUNTUAL EN EIXOS LOCALS

```

function reaccions=calcul_puntual(carrega,dist1,L,extrems)
dist2=L-dist1;
switch extrems
    case 0 % rígid-rígid
        reaccions(1:3)=-carrega*dist2^2*(3-2*dist2/L)/L^2; % Reaccions d'empotrament als nus inferior.
        reaccions(4:6)=[0,[carrega(3),-carrega(2)]*dist1*dist2^2/L^2]; % Moments d'empotrament al nus
inferior.
        reaccions(7:9)=-carrega*dist1^2*(3-2*dist1/L)/L^2; % Reaccions d'empotrament als nus superior.
        reaccions(10:12)=[0,[-carrega(3),carrega(2)]*dist1^2*dist2/L^2]; % Moments d'empotrament al nus
superior.
    case 1 % rígid-articulad
        reaccions(1:3)=-carrega*dist2*(3-dist2^2/L^2)/(2*L); % Reaccions d'empotrament als nus
inferior.
        reaccions(4:6)=[0,[carrega(3),-carrega(2)]*dist1*dist2*(L+dist2)/(2*L^2)]; % Moments
d'empotrament al nus inferior.
        reaccions(7:9)=-carrega*dist1^2*(3-dist1/L)/(2*L^2); % Reaccions d'empotrament als nus
superior.
        reaccions(10:12)=[0,0,0]; % Moments d'empotrament al nus superior.
    case 2 % articulad-rígid
        reaccions(1:3)=-carrega*dist2^2*(3-dist2/L)/(2*L^2); % Reaccions d'empotrament als nus
inferior.
        reaccions(4:6)=[0,0,0]; % Moments d'empotrament al nus inferior.
        reaccions(7:9)=-carrega*dist1*(3-dist1^2/L^2)/(2*L); % Reaccions d'empotrament als nus
superior.
        reaccions(10:12)=[0,[-carrega(3),carrega(2)]*dist1*dist2*(L+dist1)/(2*L^2)]; % Moments
d'empotrament al nus superior.
    case 3 % articulad_articulad
        reaccions(1:3)=-carrega*dist2/L; % Reaccions d'empotrament als nus inferior.
        reaccions(4:6)=[0,0,0]; % Moments d'empotrament al nus inferior.
        reaccions(7:9)=-carrega*dist1/L; % Reaccions d'empotrament als nus superior.
        reaccions(10:12)=[0,0,0]; % Moments d'empotrament al nus superior.
end
end

```

% FUNCIO DE CÀLCUL DE LES REACCIONS PER MOMENT APLICAT A LA BARRA EN EIXOS LOCALS

```

function reaccions=calcul_moment(carrega,dist1,L,extrems)
dist2=L-dist1;
switch extrems
    case 0 % rígid-rígid
        reaccions(1:3)=[0,[carrega(3),-carrega(2)]*6*dist1*dist2/L^3]; % Reaccions d'empotrament als
nus inferior.

```



```

    reaccions(4:6)=[-carrega(1)*dist2/L,[carrega(2),carrega(3)]*dist2*(2-3*dist2/L)/L]; % Moments
d'empotrament al nus inferior.
    reaccions(7:9)=[0,[-carrega(3),carrega(2)]*6*dist1*dist2/L^3]; % Reaccions d'empotrament als
nus superior.
    reaccions(10:12)=[-carrega(1)*dist1/L,[carrega(2),carrega(3)]*dist1*(2-3*dist1/L)/L]; % Moments
d'empotrament al nus superior.
    case 1 % rígid-articulat
        reaccions(1:3)=[0,[carrega(3),-carrega(2)]*3*dist1*(2-dist1/L)/(2*L^2)]; % Reaccions
d'empotrament als nus inferior.
        reaccions(4:6)=[-carrega(1),[carrega(2),carrega(3)]*(1-3*dist2^2/L^2)/2]; % Moments
d'empotrament al nus inferior.
        reaccions(7:9)=[0,[-carrega(3),carrega(2)]*3*dist1*(2-dist1/L)/(2*L^2)]; % Reaccions
d'empotrament als nus superior.
        reaccions(10:12)=[0,0,0]; % Moments d'empotrament al nus superior.
    case 2 % articulad-rígid
        reaccions(1:3)=[0,[carrega(3),-carrega(2)]*3*dist2*(2-dist2/L)/(2*L^2)]; % Reaccions
d'empotrament als nus inferior.
        reaccions(4:6)=[0,0,0]; % Moments d'empotrament al nus inferior.
        reaccions(7:9)=[0,[-carrega(3),carrega(2)]*3*dist2*(2-dist2/L)/(2*L^2)]; % Reaccions
d'empotrament als nus superior.
        reaccions(10:12)=[-carrega(1),[carrega(2),carrega(3)]*(1-3*dist1^2/L^2)/2]; % Moments
d'empotrament al nus superior.
    case 3 % articulad_articulad
        reaccions(1:3)=[0,[carrega(3),-carrega(2)]/L]; % Reaccions d'empotrament als nus inferior.
        reaccions(4:6)=[0,0,0]; % Moments d'empotrament al nus inferior.
        reaccions(7:9)=[0,[-carrega(3),carrega(2)]/L]; % Reaccions d'empotrament als nus superior.
        reaccions(10:12)=[0,0,0]; % Moments d'empotrament al nus superior.
end
end

```

```

%% CREACIÓ DELS VECTORS D'ESFORÇOS I MOVIMENTS DE L'ESTRUCTURA INTRODUIENT MOVIMENTS I GIRS IMPOSATS I
RODET INCLINAT.

```

```

function [Vector_esforços_prescrits,Vector_esforços,Vector_moviments]=contorn_prescrits_i_rodet
(Matriu_Rigidesa_G,n,ns,sistema,Ncond,Ntotfor)
Vector_esforços=zeros(ns,1);
Vector_esforços_prescrits=Vector_esforços;
Vector_moviments=zeros(ns,1);
for i=1:n
    sumafor=Ntotfor(6*i-5:6*i);
    if Ncond(i,1)==1 % Si és rodet inclinat canvia esforços al sistema d'eixos locals del rodet
        Tr=calcul_canvi_base(Ncond(i,2:4)); % Calcula matriu canvi de base plànol del rodet.
        T=[Tr,zeros(3);zeros(3),Tr];
        sumafor=T'*sumafor;
    end
    for j=1:6
        k=sistema(6*i+j-6);
        if k
            Vector_esforços(k)=sumafor(j); % Col·loca els esforços al vector d'esforços
            Vector_esforços_prescrits(k)=sumafor(j);
        end
    end
    if Ncond(i,1)==7 || Ncond(i,1)==9 % Moviments imposats
        for j=1:3
            k=sistema(6*i+j-6);
            l=Ncond(i,j+1);
            if k && l
                Vector_esforços_prescrits=Vector_esforços_prescrits-l*Matriu_Rigidesa_G(1:ns,k);
                Vector_moviments(k,l)=1;
            end
        end
    end
    if Ncond(i,1)==8 || Ncond(i,1)==9 % Girs imposats
        for j=4:6
            k=sistema(6*i+j-6);
            l=Ncond(i,j+1);
            if k && l
                Vector_esforços_prescrits=Vector_esforços_prescrits-l*Matriu_Rigidesa_G(1:ns,k);
                Vector_moviments(k,l)=1;
            end
        end
    end
end
end
end
end

```

```

%% MODIFICACIÓ DE LA MATRIU DE RIGIDESA GLOBAL PER INTRODUIR CONTORNS ELÀSTICS.

```

```

function Matriu_Rigidesa_G=contorn_elastic(Matriu_Rigidesa_G,n,sistema,contorn,K)
for i=1:n
    if contorn(i)==4 || contorn(i)==6 % Moviments elàstics
        for j=1:3

```

```

        k=sistema(6*i+j-6);
        if k
            Matriu_Rigidesa_G(k,k)=Matriu_Rigidesa_G(k,k)+K(i,j); %Afegeix la constant elàstica a
la diagonal de la matriu.
        end
    end
end
if contorn(i)==5 || contorn(i)==6 % Girs elàstics
    for j=4:6
        k=sistema(6*i+j-6);
        if k
            Matriu_Rigidesa_G(k,k)=Matriu_Rigidesa_G(k,k)+K(i,j); %Afegeix la constant elàstica a
la diagonal de la matriu.
        end
    end
end
end
end
%% REORDENA ELS RESULTATS SEGONS L'ORDRE INICIAL I TORNA ELLS MOVIMENTS I REACCIONS DE RODET INCLINAT
ALS EIXOS GLOBAIS.

function [Return_reaccions,Return_moviments]=ordre_inicial_nusos(sistema,Vector_reaccions,
Vector_moviments,n,Ncond)
Return_moviments=zeros(6*n,1);
Return_reaccions=zeros(6*n,1);
for i=1:n
    for j=1:6
        k=sistema(6*i+j-6);
        if k
            Return_moviments(6*i+j-6)=Vector_moviments(k);
            if Ncond(i,1)>0
                Return_reaccions(6*i+j-6)=Vector_reaccions(k);
            end
        end
    end
end
if Ncond(i,1)==1 % Si és rodet inclinat torna esforços i desplaçaments del sistema d'eixos locals
del rodet al global
    Tr=calcul_canvi_base(Ncond(i,2:4)); % Calcula matriu canvi de base del rodet.
    T=[Tr,zeros(3);zeros(3),Tr];
    Return_moviments(6*i-5:6*i)=T*Return_moviments(6*i-5:6*i);
    Return_reaccions(6*i-5:6*i)=T*Return_reaccions(6*i-5:6*i);
end
end
end
end

```

```

% Funció dibuix_diagrames per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 1 de juny de 2.012.
%
% Darrera versió 13/10/2012
%
% Dibuixa els diagrames d'esforços i deformada de les barres.
% Crea un objecte diagrama per a cada barra, amb un objecte hgtransform per a cada tipus de diagrama.
% Tot en eixos locals per a ser traslladat i rotat al seu lloc a la gràfica.
%
% Calcula els trams i les expressions dels diferents diagrames.
% Calcula la deformada.
% Calcula màxims i mínims.
% Dibuixa els diagrames i la deformada.
% Retorna les expressions en forma de text.
%
% Entrada:
% - Long: longitud de la barra.
% - esforcos: esforços en extrems de barra, en format vector 1x12.
% - moviments: moviments dels nusos inferior i superior, en format vector 1x12.
% - repartida: càrrega repartida, vector 1x6. [mòdul,x,y,z,distància a nus inferior,distància a nus superior].
% - projectada: càrrega repartida projectada, mateix format que la repartida.
% - puntual: càrrega puntual, vector 1x5. [mòdul,x,y,z,distancia nus inferior].
% - moment: càrrega moment, vector 1x5, mateix format que puntual.
% - E,A,Iz,Iz: propietats mecàniques de la barra.
%
% Sortida:
% - diag: objecte classe diagrama, on:
% + hgN: handle de l'objecte hgtransform que conté la gràfica del diagrama d'axils, en eixos locals de la barra.
% + hgTy: handle de l'objecte hgtransform que conté la gràfica del diagrama de tallants en y, en eixos locals de la barra.
% + hgTz: handle de l'objecte hgtransform que conté la gràfica del diagrama de tallants en z, en eixos locals de la barra.
% + hgMx: handle de l'objecte hgtransform que conté la gràfica del diagrama de moments en x, en eixos locals de la barra.
% + hgMy: handle de l'objecte hgtransform que conté la gràfica del diagrama de moments en y, en eixos locals de la barra.
% + hgMz: handle de l'objecte hgtransform que conté la gràfica del diagrama de moments en z, en eixos locals de la barra.
% + hgdeformada: handle de l'objecte hgtransform que conté la grafica de la deformada, en eixos locals de la barra.
% + Tmax: Tallant màxim en valor absolut, per a escalar els diagrames d'axils i tallants a la gràfica.
% + Mmax: Tallant màxim en valor absolut, per a escalar els diagrames de moments a la grafica.
% + Dmax: Desplaçament màxim en valor absolut, per a escalar la deformada a la grafica.
% + Oini: Gir a l'extrem inicial de la barra.
% + Ofin: Gir a l'extrem final de la barra.
% + fletxay: [fletxa màxima en y,índex a la llista de coordenades de la deformada]
% + fletxaz: [fletxa màxima en z,índex a la llista de coordenades de la deformada]
% - varargout: Text amb les expressions analítiques de les lleis d'esforços, deformada i màxims i mínims. Opcional.
%
% Funcionament:
% Calcula les lleis d'esforços a partir de les carregues i els eforços en extrem de barra, en forma de coeficients de polinomi per a cada tram de barra delimitat per la localització de les carregues.
% Per obtenir la deformada es calculen les integrals de les lleis d'esforços a cada tram. S'utilitza com a condició de contorn els desplaçaments verticals calculats dels nusos extrems i la continuïtat de la deformada i la seva pendent.
%
%% FUNCIO PRINCIPAL:

function [diag,varargout]=dibuix_diagrames(long,esforcos,moviments,repartida,projectada,puntual,moment,E,A,Iz,Iy)

detall=nargout-1; % Si s'hi introdueix més d'una sortida retorna els diagrames amb marques i dades numèriques per a la presentació en detall. Amb una sola sortida dibuixa els diagrames simples per a ser presentats a la grafica de l'estructura.

sep=long/50; % Separació entre punts per al dibuix dels diagrames, 'resolució' del diagrama.
diag=diagrama; % Crea l'objecte de la classe diagrama.

[puntual,moment,repartida,projectada,dcx]=carregues(puntual,moment,repartida,projectada); % Normalitza les càrregues per al càlcul.

N=calcul_tallant(esforcos(1),[puntual(2),puntual(5)],[repartida(2),repartida(5),long-repartida(6)],[projectada(2),projectada(5),long-projectada(6)],long,dcx); % Expressió axils.
Ty=calcul_tallant(esforcos(2),[puntual(3),puntual(5)],[repartida(3),repartida(5),long-repartida(6)],[projectada(3),projectada(5),long-projectada(6)],long,dcx); % Expressió tallants en y.
Tz=calcul_tallant(esforcos(3),[puntual(4),puntual(5)],[repartida(4),repartida(5),long-repartida(6)],[projectada(4),projectada(5),long-projectada(6)],long,dcx); % Expressió tallants en z.

```

```

Mx=calcul_moment(0,esforcos(4),[moment(2),moment(5)],[0,0],[0,0,0],[0,0,0],long,dcx,1); % Expressió moment torsor.
My=calcul_moment(esforcos(3),esforcos(5),[moment(3),moment(5)],[puntual(4),puntual(5)],[repartida(4),repartida(5),long-repartida(6)],[projectada(4),projectada(5),long-projectada(6)],long,dcx,-1); % Expressió moment en y.
Mz=calcul_moment(esforcos(2),esforcos(6),[moment(4),moment(5)],[puntual(3),puntual(5)],[repartida(3),repartida(5),long-repartida(6)],[projectada(3),projectada(5),long-projectada(6)],long,dcx,1); % Expressió moment en z.

dx=calcul_dx(moviments(1),N,E*A,long); % Expressió deformació longitudinal.
[dy,Oz,Ozini,Ozfin]=calcul_d([moviments(2),moviments(8)],Mz,E*Iz,long,1); % Expressions deformació transversal en y i gir en z.
[dz,Oy,Oyini,Oyfin]=calcul_d([moviments(3),moviments(9)],My,E*Iy,long,-1); % Expressions deformació transversal en z i gir en y.
diag.Oini=[Oyini,Ozini];
diag.Ofin=[Oyfin,Ozfin];

N0=zero(N); % Punts on N=0.
Ty0=zero(Ty); % Punts on Ty=0.
Tz0=zero(Tz); % Punts on Tz=0.
Mx0=zero(Mx); % Punts on Mx=0.
My0=zero(My); % Punts on My=0.
Mz0=zero(Mz); % Punts on Mz=0.
Oy0=zero(Oy); % punts on Oy=0.
Oz0=zero(Oz); % punts on Oz=0.

[diag.hgN,Nmin,Nmax]=dibudiag('N',N,N0,[],sep,long,1,1); % Dibuixa el diagrama d'axils.
[diag.hgTy,Tymin,Tymax]=dibudiag('Ty',Ty,Ty0,[],sep,long,1,1); % Dibuixa el diagrama de tallants en y.
[diag.hgTz,Tzmin,Tzmax]=dibudiag('Tz',Tz,Tz0,[],sep,long,1,1); % Dibuixa el diagrama de tallants en z.
[diag.hgMx,Mxmin,Mxmax]=dibudiag('Mx',Mx,Mx0,[],sep,long,1,1); % Dibuixa el diagrama de moments en x.
[diag.hgMy,Mymin,Mymax]=dibudiag('My',My,My0,Tz0,sep,long,1,-1); % Dibuixa el diagrama de moments en y.
[diag.hgMz,Mzmin,Mzmax]=dibudiag('Mz',Mz,Mz0,Ty0,sep,long,1,1); % Dibuixa el diagrama de moments en z.
[diag.hgdeformada,dxmax,dymax,dzmax]=dibudef(dx,dy,dz,My0,Mz0,Oy0,Oz0,Oyini,Oyfin,Ozini,Ozfin,sep,long,detall); % Dibuixa la deformada.

diag.Tmax=max(abs([Nmin(2),Nmax(2),Tymin(2),Tymax(2),Tzmin(2),Tzmax(2)])); % Màxim tallant o axil en termes absoluts, per a determinar l'escala dels diagrames.
diag.Mmax=max(abs([Mxmin(2),Mxmax(2),Mymin(2),Mymax(2),Mzmin(2),Mzmax(2)])); % Màxim moment en termes absoluts, per a determinar l'escala dels diagrames.
diag.Dmax=max(abs([dxmax(2),dymax(2),dzmax(2)])); % Màxim desplaçament en termes absoluts, per a determinar l'escala de la deformada.

if detall
    txt=monta_texte('','N',equac(N),Nmin,Nmax,N0); % Crea les cadenes de text amb les expressions analítiques.
    txt=monta_texte(txt,'Ty',equac(Ty),Tymin,Tymax,Ty0);
    txt=monta_texte(txt,'Tz',equac(Tz),Tzmin,Tzmax,Tz0);
    txt=monta_texte(txt,'Mx',equac(Mx),Mxmin,Mxmax,Mx0);
    txt=monta_texte(txt,'My',equac(My),Mymin,Mymax,My0);
    txt=monta_texte(txt,'Mz',equac(Mz),Mzmin,Mzmax,Mz0);
    txt=monta_texte(txt,'Oy',equac(Oy));
    txt=monta_texte(txt,'Oz',equac(Oz));
    txt=monta_texte(txt,'dx',equac(dx),dxmax);
    txt=monta_texte(txt,'dy',equac(dy),dymax);
    txt=monta_texte(txt,'dz',equac(dz),dzmax);
    varargout{1}=char(txt);
end
end

%% NORMALITZACIÓ DE LES CÀRREGUES.
%
% Multiplica el vector direcció de cada càrrega pel seu mòdul.
% Obtè la rel·lació entre longitud de la càrrega projectada i de la barra.

function [puntual,moment,repartida,projectada,dcx]=carregues(puntual,moment,repartida,projectada)
    if puntual(1) && any(puntual(2:4))
        puntual(2:4)=puntual(1)*puntual(2:4)/norm(puntual(2:4));
    else
        puntual(2:4)=[0,0,0];
    end
    if moment(1) && any(moment(2:4))
        moment(2:4)=moment(1)*moment(2:4)/norm(moment(2:4));
    else
        moment(2:4)=[0,0,0];
    end
    if repartida(1) && any(repartida(2:4))
        repartida(2:4)=repartida(1)*repartida(2:4)/norm(repartida(2:4));
    else
        repartida(2:4)=[0,0,0];
    end
    if projectada(1) && any(projectada(2:4))

```

```

    projectada(2:4)=projectada(1)*projectada(2:4)/norm(projectada(2:4));
    vbarra=[1,0,0];
    vcarrega=projectada(2:4);
    vperp=cross(vbarra,vcarrega);
    vperp=cross(vperp,vcarrega);
    vperp=vperp/norm(vperp);
    dcx=abs(dot(vperp,vbarra));
else
    projectada(2:4)=[0,0,0];
    dcx=0;
end
end
end

%% CÀLCUL DELS POLINOMIS QUE DEFINEIXEN ELS DIAGRAMES D'AXILS I TALLANTS.

function T=calcul_tallant(forestrem,fpuntual,frepartida,fprojectada,long,dcx)
if forestrem || fpuntual(1) || frepartida(1) || fprojectada(1) % Si hi ha tallant en crea el diagrama.
    xT=[];
    % Defineix els trams del diagrama:
    if fpuntual(1)
        xT=fpuntual(2);
    end
    if frepartida(1)
        xT=[xT,frepartida(2),frepartida(3)];
    end
    if fprojectada(1)
        xT=[xT,fprojectada(2),fprojectada(3)];
    end
    xT=unique([0,xT,long]);
    T=zeros(numel(xT)-1,4);

    % Per a cada tram calcula els coeficients de la expressió analítica
    % del diagrama de tallants en y:
    for i=2:numel(xT)
        x0=xT(i-1);
        x1=xT(i);
        T(i-1,3:4)=[x0,x1];
        if fpuntual(1) && x1>fpuntual(2)
            T(i-1,1:2)=[0,-forestrem-fpuntual(1)];
        else
            T(i-1,1:2)=[0,-forestrem];
        end
        if frepartida(1) && x1>frepartida(2) && x1<=frepartida(3)
            T(i-1,1:2)=T(i-1,1:2)+[-frepartida(1),frepartida(1)*frepartida(2)];
        elseif frepartida(1) && x1>frepartida(3)
            T(i-1,1:2)=T(i-1,1:2)+[0,-frepartida(1)*(frepartida(3)-frepartida(2))];
        end
        if fprojectada(1) && x1>fprojectada(2) && x1<=fprojectada(3)
            T(i-1,1:2)=T(i-1,1:2)+[-fprojectada(1)*dcx,fprojectada(1)*fprojectada(2)*dcx];
        elseif fprojectada(1) && x1>fprojectada(3)
            T(i-1,1:2)=T(i-1,1:2)+[0,-fprojectada(1)*(fprojectada(3)-fprojectada(2))*dcx];
        end
    end
end
else
    T=[];
end
end

%% CÀLCUL DELS POLINOMIS QUE DEFINEIXEN ELS DIAGRAMES DE MOMENTS.

function M=calcul_moment(forestrem,momextrem,fmoment,fpuntual,frepartida,fprojectada,long,dcx,signe)
if forestrem || momextrem || fmoment(1) || fpuntual(1) || frepartida(1) || fprojectada(1) % Si hi ha moments en eix z en crea el diagrama.
    xM=[];
    if fmoment(1)
        xM=fmoment(2);
    end
    if fpuntual(1)
        xM=[xM,fpuntual(2)];
    end
    if frepartida(1)
        xM=[xM,frepartida(2),frepartida(3)];
    end
    if fprojectada(1)
        xM=[xM,fprojectada(2),fprojectada(3)];
    end
    xM=unique([0,xM,long]);
    M=zeros(numel(xM)-1,5);
    for i=2:numel(xM)
        x0=xM(i-1); % Valor X a l'inici del tram.
        x1=xM(i); % Valor X al final del tram.
        M(i-1,4:5)=[x0,x1]; % Afegeix les dades d'X inicial i final a cada expressió analítica.
    end
end
end

```

```

% Expressió analítica del diagrama de moments segons els esforços.
% En forma de vector amb els valors dels factors.
if fmoment(1) && xM(i)>fmoment(2)
    M(i-1,1:3)=[0,forextrem*signe,-momextrem-fmoment(1)];
else
    M(i-1,1:3)=[0,forextrem*signe,-momextrem];
end
if fpuntual(1) && x1>fpuntual(2)
    M(i-1,1:3)=M(i-1,1:3)+[0,fpuntual(1),-fpuntual(1)*fpuntual(2)]*signe;
end
if frepartida(1) && x1>frepartida(2) && x1<=frepartida(3)
    M(i-1,1:3)=M(i-1,1:3)+[frepartida(1)/2,-frepartida(1)*frepartida(2),frepartida(1)
*frepartida(2)^2/2]*signe;
elseif frepartida(1) && x1>frepartida(3)
    M(i-1,1:3)=M(i-1,1:3)+[0,frepartida(1)*(frepartida(3)-frepartida(2)),-frepartida(1)*
(frepartida(3)-frepartida(2))*(frepartida(3)+frepartida(2))/2]*signe;
end
if fprojectada(1) && x1>fprojectada(2) && x1<=fprojectada(3)
    M(i-1,1:3)=M(i-1,1:3)+[fprojectada(1)*dcx/2,-fprojectada(1)*fprojectada(2)*dcx,
fprojectada(1)*(fprojectada(2)*dcx)^2/2]*signe;
elseif fprojectada(1) && x1>fprojectada(3)
    M(i-1,1:3)=M(i-1,1:3)+[0,fprojectada(1)*(fprojectada(3)*dcx-fprojectada(2)*dcx,-
fprojectada(1)*(fprojectada(3)*dcx-fprojectada(2)*dcx)*(fprojectada(3)*dcx+fprojectada(2)*dcx)/2]
*signe;
end
end
else
    M=[];
end
end
end

%% Càlcul dels polinomis que defineixen les deformacions.
function dx=calcul_dx(d0,N,EA,long)
n=size(N,1); % Número de trams.
if n
    dx=zeros(n,5);
    dx(:,4:5)=N(:,3:4);
    for i=1:n
        dx(i,1:3)=polyint(N(i,1:2)/EA);
        dx(i,3)=polyval([-dx(i,1:2),d0],dx(i,4));
        d0=polyval(dx(i,1:3),dx(i,5));
    end
else
    dx=[0,0,d0,0,long];
end
end

function [d,O,Oini,Ofin]=calcul_d(moviment,M,EI,long,signe)
n=size(M,1); % Número de trams.
if n
    O=zeros(n,6);
    d=zeros(n,7);
    O(:,5:6)=M(:,4:5);
    d(:,6:7)=M(:,4:5);
    O0=(moviment(2)-moviment(1))/long; % Valors inicials de gir de sòlid rígid i desplaçament
vertical.
    d0=moviment(1);
    for i=1:n
        x0=M(i,4);
        x1=M(i,5);
        O(i,1:4)=polyint(M(i,1:3)*signe/EI); % Integració del diagrama de moments per obtenir
l'expressió del gir.
        O(i,4)=polyval([-O(i,1:3),O0],x0); % Terme independent, per continuïtat.
        O0=polyval(O(i,1:4),x1); % Gir al final del tram, serà l'inicial del tram següent.

        d(i,1:5)=polyint(O(i,1:4)); % Integració del gir per obtenir el desplaçament vertical.
        d(i,5)=polyval([-d(i,1:4),d0],x0); % Terme independent, per continuïtat.
        d0=polyval(d(i,1:5),x1); % Desplaçament vertical al final del tram, serà l'inicial al tram
següent.
    end
    O1=(moviment(2)-d0)/long; % Calcula el gir a l'extrem inferior imponent la condició de contorn
del desplaçament vertical conegut a l'extrem superior.
    O(:,4)=O(:,4)+ones(n,1)*O1; % Corregeix les expressions de gir i desplaçaments afegint el gir
inicial calculat.
    d(:,4)=d(:,4)+ones(n,1)*O1;
    Oini=polyval(O(i,1:4),0); % Gir inicial.
    Ofin=polyval(O(i,1:4),long); % Gir final.
elseif any(moviment)
    Oini=(moviment(2)-moviment(1))/long;
    d=[0,0,0,Oini,moviment(1),0,long];
end
end

```

```

        O=[0,0,0,Oini,0,long];
        Ofin=Oini;
    else
        d=[];
        O=[];
        Oini=0;
        Ofin=0;
    end
end

%% RECERCA DE ZEROS ALS POLINOMIS.

function x0=zero(P)
    [n,l]=size(P);
    x0=[];
    for i=1:n % Per a tots els trams comproba les arrels dels polinomis.
        x=roots(P(i,1:l-2)); % Arrels dels polinomis que defineixen els trams.
        for j=1:size(x,1); % Per a totes les arrels calculades...
            if isreal(x(j)) && x(j)>=P(i,l-1) && x(j)<=P(i,l) % Si l'arrel és un nombre real i està
compresa entre els límits del tram.
                x0=[x0,x(j)]; % Afegeix l'arrel a la llista.
            end
        end
    end

    for i=2:n % Comproba els límits entre trams.
        if polyval(P(i-1,1:l-2),P(i-1,l)) * polyval(P(i,1:l-2),P(i,l-1)) < 0 % Si el valor final del
tram anterior i el valor inicial del tram actual són de signe diferent...
            x0=[x0,P(i,l-1)]; % Afegeix el límit entre els trams a la llista.
        end
    end
    x0=unique(x0); % Ordena i elimina resultats repetits.
end

%% DIBUIXA EL DIAGRAMA.

function [hg,miny,maxy]=dibudiag(txt,P,P0,xm,sep,long,detall,signe)
    if ~isempty(P)
        hg=hgtransform;
        miny=[0,Inf];
        maxy=[0,-Inf];
        [n,l]=size(P);
        xd=0;
        yd=0;
        for i=1:n % En cada tram on es defineix el diagrama...
            if l>4 && any(P(i,1:l-4)) % Si el diagrama és una corba defineix una sèrie de punts
intermitjos per definirla.
                x=[P(i,l-1):sep:P(i,l)];
            else
                x=[P(i,l-1),P(i,l)]; % Si el diagrama és una recta en defineix els punts inicial i
final del tram.
            end
            for j=1:size(xm,1) % Si dins del tram hi ha un candidat a mínim o màxim l'afegeix als punts
perque sigui evaluat.
                if xm(j)>P(i,l-1) && xm(j)<P(i,l)
                    x=[x,xm(j)];
                end
            end
            x=unique(x); % Ordena els punts i en descarta els repetits.
            y=polyval(P(i,1:l-2),x); % Evalua els punts.
            xd=[xd,x]; % Afegeix els punts al diagrama.
            yd=[yd,y];
            [miny(2),k]=min([miny(2),y]); % Comproba si el mínim i màxim del tram són els absoluts.
            if k~=1
                miny(1)=x(k-1);
            end
            [maxy(2),k]=max([maxy(2),y]);
            if k~=1
                maxy(1)=x(k-1);
            end
        end
        xd=[xd,P(n,l)]; % Uneix el punt final del diagrama amb la barra.
        yd=[yd,0];
        line(xd,-yd*signe,'Linewidth',2,'Parent',hg,'HitTest','off','Clipping','on') % Dibuixa el
diagrama.

        if detall
            if miny(1)~=0 && miny(1)~=long
                scatter(miny(1),-miny(2)*signe,100,'+','k','Parent',hg) % Marca el mínim absolut.
                txtdiag(miny(1),miny(2),txt,long,hg,signe)
            end
            if maxy(1)~=0 && maxy(1)~=long && maxy(1)~=miny(1)

```



```

        scatter(maxy(1),-maxy(2)*signe,100,'+','k','Parent',hg) % Marca el màxim absolut.
        txtdiag(maxy(1),maxy(2),txt,long,hg,signe)
    end
    txtdiag(0,yd(2),txt,long,hg,signe) % Moment inicial.
    txtdiag(long,yd(length(yd)-1),txt,long,hg,signe) % Moment final.
    for i=1:size(P0,2) % Si dins del tram hi han ceros.
        txtdiag(P0(i),0,txt,long,hg,signe) % Marca els ceros.
    end
end
else
    hg=[];
    miny=[0,0];
    maxy=[0,0];
end
end
function txtdiag(x,y,txt,long,hg,signe)
    if x>long/2
        halin='right';
    else
        halin='left';
    end
    if y>0
        valin='top';
    else
        valin='bottom';
    end
    text(x,-y*signe,[txt,'(x=',num2str(x),')=',num2str(y)],'VerticalAlignment',
    valin,'HorizontalAlignment',halin,'Parent',hg);
end

%% DIBUIXA LA DEFORMADA.

function [hg,dxmax,dymax,dzmax]=dibudef(dx,dy,dz,My0,Mz0,Oy0,Oz0,Oyini,Oyfin,Ozini,Ozfin,sep,long,
detall)
    hg=hgtransform;

    [nx,lx]=size(dx);
    [ny,ly]=size(dy);
    [nz,lz]=size(dz);

    if ny || nz %si hi ha deformació transversal.
        t=[0,dx(:,5)']; % Defineix els trams.
        if ny
            t=[t,dy(:,7)'];
        end
        if nz
            t=[t,dz(:,7)'];
        end
        t=unique(t);

        ix=1;
        iy=1;
        iz=1;
        xt=[];
        e=[];
        s=[];
        y=[];
        z=[];

        for i=2:size(t,2)
            xinfly=[];
            xinflz=[];
            xmaxy=[];
            xmaxz=[];
            x=t(i-1):sep:t(i);
            for j=1:size(My0,2)
                if My0(j)>t(i-1) && My0(j)<t(i)
                    xinfly=[xinfly,My0(j)];
                end
            end
            for j=1:size(Mz0,2)
                if Mz0(j)>t(i-1) && Mz0(j)<t(i)
                    xinflz=[xinflz,Mz0(j)];
                end
            end
            for j=1:size(Oy0,2)
                if Oy0(j)>t(i-1) && Oy0(j)<t(i)
                    xmaxz=[xmaxz,Oy0(j)];
                end
            end
            for j=1:size(Oz0,2)

```

```

        if Oz0(j)>t(i-1) && Oz0(j)<t(i)
            xmaxy=[xmaxy,Oz0(j)];
        end
    end

    x=unique([x,xinflz,xinflz,xmaxy,xmaxz]);
    xt=[xt,x];
    e=[e,polyval(dx(ix,1:3),x)]; % Deformació unitària en x.
    s=[xt+e]; % Deformada en x.
    while t(i)>dx(ix,5)
        ix=ix+1;
    end
    if ny
        while t(i)>dy(iy,7)
            iy=iy+1;
        end
        y=[y,polyval(dy(iy,1:5),x)]; % Deformada en y.
    else
        y=[y,x*0];
    end
    if nz
        while t(i)>dz(iz,7)
            iz=iz+1;
        end
        z=[z,polyval(dz(iz,1:5),x)]; % Deformada en z.
    else
        z=[z,x*0];
    end

end
else
    xt=[0,dx(nx,5)];
    e=[polyval(dx(1,1:3),0),polyval(dx(nx,1:3),dx(nx,5))];
    s=xt+e;
    y=[0,0];
    z=[0,0];
end
defx.x=xt;
defx.e=e;
line(s,y,z,'Linewidth',2,'Parent',hg,'UserData',defx,'HitTest','off','Clipping','on') % Dibuixa la
deformada.

[xm,imx]=max(abs(s)); % Desplaçament en x màxim.
dxmax=[xt(imx),e(imx)];
[xm,imy]=max(abs(y)); % Desplaçament en y màxim.
dymax=[xt(imy),y(imy)];
[xm,imz]=max(abs(z)); % Desplaçament en z màxim.
dzmax=[xt(imz),z(imz)];

if detall % Si s'ha cridat la funció amb l'opció de mostrar detalls numerics i marques.
    % Faig servir un petit truc per mostrar correctament els scatter3.
    % Matlab no els col·loca correctament si s'encapsulen directament
    % en un hgtransform, el que faig és extreure els objectes que conté
    % l'scatter3 i ficar-los dins del hgtransform en comptes de ficar
    % el scatter3 dins del hgtransform. D'aquesta manera funciona.

    l=length(xt);
    set(get(scatter3(s(imy),y(imy),z(imy),100,'+', 'k'),'Children'),'Parent',hg,'UserData','dy') %
Marca dy màx.
    for i=1:length(Mz0)
        imoz=find(xt==Mz0(i)); % Punt d'inflexió.
        if imoz(1) ~ =1 && imoz(length(imoz)) ~ =1 % Si el punt d'inflexió no està a l'inici o al
final.
            set(get(scatter3(s(imoz),y(imoz),z(imoz),100,'o', 'k'),'Children'),'Parent',
hg,'UserData','dy') % Marca inflexió gir en z.
            end
        end
        if imy ~ =1 && imy ~ =length(imy) % Si el desplaçament màxim en y no està a l'inici ni al final...
            txt={['dx(x=',num2str(xt(imy)),')=',num2str(e(imy))];['dy(x=',num2str(xt(imy)),')=',num2str
(y(imy))];['Oz(x=',num2str(xt(imy)),')=0']}; % Texte desplaçament màxim en y.
            txtdef(s(imy),y(imy),z(imy),txt,'dy',long,hg)
        end
        txt={['dx(x=0)',num2str(e(1))];['dy(x=0)',num2str(y(1))];['Oz(x=0)',num2str(Ozini)]}; %
Texte desplaçament en y a l'inici.
        txtdef(s(1),y(1),z(1),txt,'dy',long,hg)
        txt={['dx(x=',num2str(long),')=',num2str(e(1))];['dy(x=',num2str(long),')=',num2str(y(1))];['Oz
(x=',num2str(long),')=',num2str(Ozfin)]}; % Texte desplaçament en y al final.
        txtdef(s(1),y(1),z(1),txt,'dy',long,hg)

        set(get(scatter3(s(imz),y(imz),z(imz),100,'+', 'k'),'Children'),'Parent',hg,'UserData','dz') %
Marca dz màx.
        for i=1:length(My0)

```

```

        imoy=find(xt==My0(i)); % Punt d'inflexió.
        if imoy(1) ~=1 && imoy(length(imoy)) ~=1 % Si el punt d'inflexió no està a l'inici o al
final.
            set(get(scatter3(s(imoy),y(imoy),z(imoy),100,'o','k'),'Children'),'Parent',
hg,'UserData','dz') % Marca inflexió gir en y.
        end
    end
    if imz ~=1 && imz~=1 % Si el desplaçament maxím en z no està a l'inici ni al final...
        txt=[['dx(x=',num2str(xt(imz)),')='],num2str(e(imz))];['dz(x=',num2str(xt(imz)),')='],num2str
(z(imz))];['Oy(x=',num2str(xt(imz)),')=0']; % Texte desplaçament màxim en z.
        txtdef(s(imz),y(imz),z(imz),txt,'dz',long,hg)
    end
    txt=[['dx(x=0)='],num2str(e(1))];['dz(x=0)='],num2str(z(1))];['Oy(x=0)='],num2str(Oyini)]; %
Texte desplaçament en z a l'inici.
    txtdef(s(1),y(1),z(1),txt,'dz',long,hg)
    txt=[['dx(x=',num2str(long),')='],num2str(e(1))];['dz(x=',num2str(long),')='],num2str(z(1))];['Oy
(x=',num2str(long),')='],num2str(Oyfin)]]; % Texte desplaçament en z al final.
    txtdef(s(1),y(1),z(1),txt,'dz',long,hg)
end
end

function txtdef(x,y,z,txt,marca,long,hg)
    if x>long/2
        halin='right';
    else
        halin='left';
    end
    if y>0
        valin='bottom';
    else
        valin='top';
    end
    text(x,y,z,txt,'VerticalAlignment',valin,'HorizontalAlignment',halin,'Parent',hg,'UserData',marca);
end

%% CONVERTEIX ELS POLINOMIS EN TEXTE.

function txt=equac(polinomi)
    txt=[];
    [n,l]=size(polinomi);

    if n~=1 || any(polinomi(1,1:l-2))
        t=[];
        for i=1:n
            t=strcat('(x=',num2str(polinomi(i,l-1)),',... ',num2str(polinomi(i,l)),') =');
            if ~any(polinomi(i,1:l-2))
                t=strcat(t,' 0');
            else
                for j=1:l-2
                    t2=[];
                    if polinomi(i,j)
                        if j==l-2
                            t2=txtcoef(polinomi(i,j));
                        elseif polinomi(i,j)==1
                            t2=' + x';
                        elseif polinomi(i,j)==-1
                            t2=' - x';
                        else
                            t2=strcat(txtcoef(polinomi(i,j)), '*x');
                        end
                        if j<l-3
                            t2=strcat(t2, '^', num2str(l-j-2));
                        end
                    end
                    t=strcat(t,t2);
                end
            end
            ttxt=strcat(t,t);
        end
        txt=txtcoef(t);
    end
    txt={txt};
end

function txt=txtcoef(coef)
    if coef>0
        txt=[' + ', num2str(coef)];
    elseif coef<0
        txt=[' - ', num2str(-coef)];
    else
        txt='';
    end
end

```

```

end

function txt=monta_texte(txt,txt1,txt2,varargin)
    if ~isempty(txt2{1})
        switch nargin
            case 3
                txt=[txt;strcat(txt1,txt2)];
            case 4
                max=varargin{1};
                txtmax=strcat(txt1,'max = ',txt1,'(x= ',num2str(max(1)),') = ',num2str(max(2)));
                txt=[txt;strcat(txt1,txt2);txtmax];
            otherwise
                min=varargin{1};
                max=varargin{2};
                cero=varargin{3};
                txtmin=strcat(txt1,'min = ',txt1,'(x= ',num2str(min(1)),') = ',num2str(min(2)));
                txtmax=strcat(txt1,'max = ',txt1,'(x= ',num2str(max(1)),') = ',num2str(max(2)));
                txt=[txt;strcat(txt1,txt2);txtmin;txtmax];
                if size(cero,2)
                    txtkk=strcat(txt1,' = 0 en x = ');
                    txtcero='';
                    for i=1:size(cero,2)
                        txtcero=strcat(txtcero,txtkk,num2str(cero(i)));
                        txtkk=' ; x = ';
                    end
                    txt=[txt;txtcero];
                end
            end
        end
    end
end
end
end

```

```

% Funció dlgnus per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 24 de maig de 2.012.
%
% Crea una finestra per introduir-hi o modificar les dades dels nusos.
%
% Entrada:
% - n: objecte tipus nus on s'emmagatzemen les dades del nus als següents camps:
% + num: Número de nus.
% + posicio: Posició a l'espai.
% + contorn: Tipus de condició de contorn.
% + rodet: Vector normal al plànol de moviment del rodet, si n'hi ha.
% + desp: desplaçaments forçats si n'hi han.
% + rig: rigideses de contorn elàstic si n'hi ha.
%
% Sortida:
% - n: Igual que l'entrada però convenientment modificades les dades del nus editat.

function n = dlgnus(n)
%% CREA LA FINESTRA, ELS PANELLS I ELS BOTONS.

tamanycme2=get(gcf, 'Position');
tm=[255,205];
sc=(tamanycme2(3:4)-tm(1:2))/2+tamanycme2(1:2);
bg=get(0, 'defaultUicontrolBackgroundColor');
f = figure('Units','pixels','Position',[sc,tm], 'NumberTitle', 'off', 'Name', ['Nus n°: ', num2str(n.
num)], 'Toolbar', 'none', 'Menubar', 'none', 'Resize', 'off', 'WindowStyle', 'modal', 'color', bg, 'KeyPressFcn',
@tecla);
panelpos=uipanel(f, 'Units', 'pixels', 'Position', [5 155 245 45], 'BorderType', 'etchedin', 'BorderWidth',
1, 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'Title', 'Posició:');
panelcontorn=uipanel(f, 'Units', 'pixels', 'Position', [5 45 245
105], 'BorderType', 'etchedin', 'BorderWidth', 1, 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'Title', 'Condició de contorn:');
uicontrol(f, 'Units', 'pixels', 'Position',
[5,10,110,30], 'Style', 'pushbutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'String', 'Acceptar', 'Callback', @acceptar);
uicontrol(f, 'Units', 'pixels', 'Position',
[140,10,110,30], 'Style', 'pushbutton', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'String', 'Cancelar', 'Callback', @cancelar);

%% CREA ELS CONTROLS AL PANELL SUPERIOR AMB INFORMACIÓ DE LA POSICIÓ.

uicontrol(panelpos, 'Units', 'pixels', 'Position',
[5,5,30,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'left', 'String', 'X:'); % Etiqueta.
uicontrol(panelpos, 'Units', 'pixels', 'Position',
[85,5,30,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'left', 'String', 'Y:'); % Etiqueta.
uicontrol(panelpos, 'Units', 'pixels', 'Position',
[165,5,30,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'left', 'String', 'Z:'); % Etiqueta.
editpos(1)=uicontrol(panelpos, 'Units', 'pixels', 'Position',
[35,5,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.posicio
(1)), 'KeyPressFcn', @tecla); % Caixa de text coordenada x.
editpos(2)=uicontrol(panelpos, 'Units', 'pixels', 'Position',
[115,5,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.posicio
(2)), 'KeyPressFcn', @tecla); % Caixa de text coordenada y.
editpos(3)=uicontrol(panelpos, 'Units', 'pixels', 'Position',
[195,5,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.posicio
(3)), 'KeyPressFcn', @tecla); % Caixa de text coordenada z.

%% CREA ELS CONTROLS AL PANELL INFERIOR AMB INFORMACIÓ DE LES CONDICIONS DE CONTORN.

popupcontorn=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[5,65,235,20], 'Style', 'popupmenu', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'BackgroundColor', 'white', 'Callback', @select_contorn, 'Value', n.contorn+1, 'String', {'Nus
lliure', 'Rodet inclinat', 'Recolzament simple', 'Encastament', 'Desplaçaments elàstics i girs
lliures', 'Girs elàstics i desplaçaments nuls', 'Desplaçaments i girs elàstics', 'Desplaçaments imposats i
girs lliures', 'Girs imposats i desplaçaments lliures', 'Desplaçaments i girs imposats'}); % Llista
selecció condició de contorn.

labelrodet(1)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[5,5,30,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'left', 'String', 'x:', 'Visible', 'off');
labelrodet(2)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[85,5,30,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'left', 'String', 'y:', 'Visible', 'off');
labelrodet(3)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[165,5,30,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',

```



```

14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.desp
(1)), 'Visible', 'off', 'KeyPressFcn', @tecla);
editdesp(2)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[115,30,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.desp
(2)), 'Visible', 'off', 'KeyPressFcn', @tecla);
editdesp(3)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[195,30,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.desp
(3)), 'Visible', 'off', 'KeyPressFcn', @tecla);
editgirs(1)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[35,5,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.desp
(4)), 'Visible', 'off', 'KeyPressFcn', @tecla);
editgirs(2)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[115,5,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.desp
(5)), 'Visible', 'off', 'KeyPressFcn', @tecla);
editgirs(3)=uicontrol(panelcontorn, 'Units', 'pixels', 'Position',
[195,5,45,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.desp
(6)), 'Visible', 'off', 'KeyPressFcn', @tecla);

select_contorn

uiwait(f) % Atura l'execució de Matlab fins que es tanqui la finestra i es retornin les dades. Necessari
per tal de que es retornin les dades corrèctament.

%% FUNCIO QUE S'EXECUTARÀ QUAN ES SELECCIONI EL TIPUS DE CONTORN
% la condició de contorn. Fa visibles o invisibles les etiquetes i les
% caixes de text segons la condició triada.

function select_contorn(hObject, eventdata, handles)
switch (get(popupcontorn, 'Value')-1)
case 1
set([editrodet, labelrodet], 'Visible', 'on')
set([editdesp, editgirs, editrigdesp, editriggirs, labeldesp, labelgirs, labelrigdesp,
labelriggirs], 'Visible', 'off')
case 4
set([editrodet, editdesp, editgirs, labelrodet, labeldesp, labelgirs], 'Visible', 'off')
set([labelrigdesp, labelriggirs, editrigdesp, editriggirs], 'Visible', 'on')
set(editrigdesp, 'Enable', 'on')
set(editriggirs, 'Enable', 'off')
case 5
set([editrodet, editdesp, editgirs, labelrodet, labeldesp, labelgirs], 'Visible', 'off')
set([labelrigdesp, labelriggirs, editrigdesp, editriggirs], 'Visible', 'on')
set(editrigdesp, 'Enable', 'off')
set(editriggirs, 'Enable', 'on')
case 6
set([editrodet, editdesp, editgirs, labelrodet, labeldesp, labelgirs], 'Visible', 'off')
set([labelrigdesp, labelriggirs, editrigdesp, editriggirs], 'Visible', 'on')
set([editrigdesp, editriggirs], 'Enable', 'on')
case 7
set([editrodet, editrigdesp, editriggirs, labelrodet, labelrigdesp,
labelriggirs], 'Visible', 'off')
set([labeldesp, labelgirs, editdesp, editgirs], 'Visible', 'on')
set(editdesp, 'Enable', 'on')
set(editgirs, 'Enable', 'off')
case 8
set([editrodet, editrigdesp, editriggirs, labelrodet, labelrigdesp,
labelriggirs], 'Visible', 'off')
set([labeldesp, labelgirs, editdesp, editgirs], 'Visible', 'on')
set(editdesp, 'Enable', 'off')
set(editgirs, 'Enable', 'on')
case 9
set([editrodet, editrigdesp, editriggirs, labelrodet, labelrigdesp,
labelriggirs], 'Visible', 'off')
set([labeldesp, labelgirs, editdesp, editgirs], 'Visible', 'on')
set([editdesp, editgirs], 'Enable', 'on')
otherwise
set([editrodet, editrigdesp, editriggirs, labelrodet, labelrigdesp, labelriggirs, labeldesp,
labelgirs, editdesp, editgirs], 'Visible', 'off')
end
end

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BOTÓ ACCEPTAR.
% Llegeix les dades de les caixes de text per retornarles i tanca la finestra.

function acceptar(hObject, eventdata, handles)
for j=1:3
n.posicio(j)=str2double(get(editpos(j), 'String'));

```



```

        n.rodet(j)=str2double(get(editrodet(j), 'String'));
        n.desp(j)=str2double(get(editdesp(j), 'String'));
        n.desp(j+3)=str2double(get(editgirs(j), 'String'));
        n.rig(j)=str2double(get(editrigdesp(j), 'String'));
        n.rig(j+3)=str2double(get(edittriggirs(j), 'String'));
    end
    n.contorn=get(popupcontorn, 'Value')-1;
    close(f)
end

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BOTÓ CANCELAR.
% Tanca la finestra sense actualitzar les dades.

function cancelar(hObject, eventdata, handles)
    close(f)
end

%% FUNCIO QUE S'EXECUTA AL PREMIER ALGUNA TECLA.
% Si es prem el retorn surt actualitzant les dades, si es prem escapament surt sense actualitzar.

function tecla(hObject,eventdata,handles)
    switch eventdata.Key
        case 'escape'
            cancelar
        case 'return'
            uicontrol(hObject)
            acceptar
    end
end
end
end

```

```

% Funció dlgbarra per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 24 de maig de 2.012.
%
% Crea una finestra per introduir-hi o modificar les dades de les barres.
%
% Entrada:
% - n: número identificador de la barra.
% - numnus: número de nusos.
% - barres: llista d'objectes tipus barra on s'emmagatzemen les propietats de la barra als següents
camps:
% + num: Número de barra.
% + nusinf: Número de nus inferior.
% + nussup: Número de nus superior.
% + artinf: Nus inferior articulat: 0=rígid, 1=articulat.
% + artsup: Nus superior articulat: 0=rígid, 1=articulat.
% + color: Color de la barra.
% + E: Mòdul de Young.
% + A: Àrea de la secció.
% + Iz: Inèrcia en eix z.
% + Iy: Inèrcia en eix y.
% + vectorIz: Vector direcció de l'eix z de la secció.
% + GI: Inèrcia a torsió.
%
% Sortida:
% - barres: Igual que la entrada però convenientment modificades les dades de la barra editada.

function barres = dlgbarra(numnus,barres,n)
%% CREA LA FINESTRA, ELS PANELLS I ELS BOTONS.

tamanycme2=get(gcf,'Position'); % Obtè el tamany i la posició de la finestra del programa principal.
tm=[255,430]; % Tamany de la finestra.
sc=(tamanycme2(3:4)-tm(1:2))/2+tamanycme2(1:2); % Posició de la finestra al centre de la finestra del
programa principal.
bg=get(0,'defaultUiControlBackgroundColor'); % Color per defecte del fons.

f = figure('Units','pixels','Position',[sc,tm],'NumberTitle','off','Name',['Barra n°: ',num2str(barres
(n).num),' (' ,num2str(barres(n).nusinf),'-',num2str(barres(n).
nussup),')'], 'ToolBar','none','MenuBar','none','Resize','off','WindowStyle','modal','color',
bg,'KeyPressFcn',@tecla); % Crea la finestra.
panelnus=uipanel(f,'Units','pixels','Position',[5 345 245 80],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Extrems de barra:'); % Crea el
panell superior amb informació dels extrems.
panelfis=uipanel(f,'Units','pixels','Position',[5 45 245 295],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Propietats:'); % Crea el panell
inferior amb les propietats.
uicontrol(f,'Units','pixels','Position',
[5,10,110,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Acceptar','Callback',@acceptar); % Crea el botó acceptar.
uicontrol(f,'Units','pixels','Position',
[140,10,110,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Cancelar','Callback',@cancelar); % Crea el botó cancelar.

%% CREA ELS CONTROLS AL PANELL SUPERIOR AMB INFORMACIÓ DELS EXTREMS.

uicontrol(panelnus,'Units','pixels','Position',
[5,40,60,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Inferior:'); % Etiqueta.
uicontrol(panelnus,'Units','pixels','Position',
[5,10,60,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Superior:'); % Etiqueta.

txtnusos=cellstr(num2str((1:numnus)')); % Llistat en text dels nusos.
popupnusinf=uicontrol(panelnus,'Units','pixels','Position',
[65,40,90,20],'Style','popupmenu','FontName','default','FontUnits','pixels','FontSize',
14,'BackgroundColor','white','String',txtnusos,'Value',barres(n).nusinf,'KeyPressFcn',@tecla); %
Control que fa apareixer llistat de nusos per seleccionar.
popupnussup=uicontrol(panelnus,'Units','pixels','Position',
[65,10,90,20],'Style','popupmenu','FontName','default','FontUnits','pixels','FontSize',
14,'BackgroundColor','white','String',txtnusos,'Value',barres(n).nussup,'KeyPressFcn',@tecla); %
Control que fa apareixer llistat de nusos per seleccionar.
checkartinf=uicontrol(panelnus,'Units','pixels','Position',
[160,40,80,20],'Style','checkbox','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Articulat','Value',barres(n).artinf,'KeyPressFcn',@tecla); %
Control que permet marcar si l'extrem inferior és articulat.
checkartsup=uicontrol(panelnus,'Units','pixels','Position',
[160,10,80,20],'Style','checkbox','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Articulat','Value',barres(n).artsup,'KeyPressFcn',@tecla); %
Control que permet marcar si l'extrem superior és articulat.

%% CREA ELS CONTROLS AL PANELL INFERIOR AMB LES PROPIETATS DE LA BARRA.

```

```

uicontrol(panelfis,'Units','pixels','Position',␣
[5,244,150,26],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',␣
14,'String','Importar de barra','Callback',@importar,'KeyPressFcn',@tecla); % Botó per importar dades␣
d'altres barres.
ns=zeros(length(barres),2);
for j=1:length(barres) % Crea una llista de les barres.
    ns(j,:)=[barres(j).nusinf,barres(j).nussup];
end
txtbarres=strcat(cellstr(num2str((1:length(barres))),' '),cellstr(num2str(ns(:,1))),'-',cellstr␣
(num2str(ns(:,2))),' ');
popupperfil=uicontrol(panelfis,'Units','pixels','Position',␣
[160,250,80,20],'Style','popupmenu','FontName','default','FontUnits','pixels','FontSize',␣
14,'BackgroundColor','white','String',txtbarres,'KeyPressFcn',@tecla); % Control que permet seleccionar␣
una barra per importar-hi les propietats.

uicontrol(panelfis,'Units','pixels','Position',␣
[5,210,150,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','Color:'); % Etiqueta.
buttoncolor=uicontrol(panelfis,'Units','pixels','Position',␣
[130,210,110,26],'Style','pushbutton','BackgroundColor',barres(n).color,'Callback',␣
@color,'KeyPressFcn',@tecla); % Botó que presenta el color i permet canviar-lo.

%Etiquetes:
uicontrol(panelfis,'Units','pixels','Position',␣
[5,180,150,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','Mòdul de Young E:');
uicontrol(panelfis,'Units','pixels','Position',␣
[5,150,150,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','Àrea de la secció:');
uicontrol(panelfis,'Units','pixels','Position',␣
[5,120,150,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','Inèrcia Iz:');
uicontrol(panelfis,'Units','pixels','Position',␣
[5,90,150,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','Inèrcia Iy:');
uicontrol(panelfis,'Units','pixels','Position',␣
[5,60,150,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','Rigidesa torsió GI:');
uicontrol(panelfis,'Units','pixels','Position',␣
[5,30,235,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','Vector direcció Iz en eixos globals:');
uicontrol(panelfis,'Units','pixels','Position',␣
[5,5,25,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','x=');
uicontrol(panelfis,'Units','pixels','Position',␣
[85,5,25,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','y=');
uicontrol(panelfis,'Units','pixels','Position',␣
[165,5,25,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','left','String','z=');

% Caixes de text:
editprop(1)=uicontrol(panelfis,'Units','pixels','Position',␣
[130,180,110,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(barres(n).E),'KeyPressFcn',␣
@tecla);
editprop(2)=uicontrol(panelfis,'Units','pixels','Position',␣
[130,150,110,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(barres(n).A),'KeyPressFcn',␣
@tecla);
editprop(3)=uicontrol(panelfis,'Units','pixels','Position',␣
[130,120,110,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(barres(n).␣
Iz),'KeyPressFcn',@tecla);
editprop(4)=uicontrol(panelfis,'Units','pixels','Position',␣
[130,90,110,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(barres(n).␣
Iy),'KeyPressFcn',@tecla);
editprop(5)=uicontrol(panelfis,'Units','pixels','Position',␣
[130,60,110,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(barres(n).␣
GI),'KeyPressFcn',@tecla);
editprop(6)=uicontrol(panelfis,'Units','pixels','Position',␣
[30,5,50,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(barres(n).vectorIz␣
(1)),'KeyPressFcn',@tecla);
editprop(7)=uicontrol(panelfis,'Units','pixels','Position',␣
[110,5,50,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(barres(n).vectorIz␣
(2)),'KeyPressFcn',@tecla);
editprop(8)=uicontrol(panelfis,'Units','pixels','Position',␣
[190,5,50,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',␣

```

```

14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(barres(n).vectorIz
(3)), 'KeyPressFcn', @tecla);

uiwait(f) % Atura l'execució de Matlab fins que es tanqui la finestra i es retornin les dades. Necessari
per tal de que es retornin les dades correctament.

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BOTÓ ACCEPTAR.
% Llegeix les dades de les caixes de text per retornarles i tanca la finestra.

function acceptar(hObject, eventdata, handles)
    barres(n).nusunf=get(popupnusunf, 'Value');
    barres(n).nussup=get(popupnussup, 'Value');
    barres(n).artinf=get(checkartinf, 'Value');
    barres(n).artsup=get(checkartsup, 'Value');
    barres(n).E=str2double(get(editprop(1), 'String'));
    barres(n).A=str2double(get(editprop(2), 'String'));
    barres(n).Iz=str2double(get(editprop(3), 'String'));
    barres(n).Iy=str2double(get(editprop(4), 'String'));
    barres(n).GIT=str2double(get(editprop(5), 'String'));
    barres(n).vectorIz=[str2double(get(editprop(6), 'String')),str2double(get(editprop
(7), 'String')),str2double(get(editprop(8), 'String'))];
    barres(n).color=get(buttoncolor, 'BackgroundColor');

    close(f)
end

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BOTÓ CANCELAR.
% Tanca la finestra sense actualitzar les dades.

function cancelar(hObject, eventdata, handles)
    close(f)
end
function importar(hObject, eventdata, handles)
    m=get(popupperfil, 'Value');
    set(editprop(1), 'String', num2str(barres(m).E));
    set(editprop(2), 'String', num2str(barres(m).A));
    set(editprop(3), 'String', num2str(barres(m).Iz));
    set(editprop(4), 'String', num2str(barres(m).Iy));
    set(editprop(5), 'String', num2str(barres(m).GIT));
    set(buttoncolor, 'BackgroundColor', barres(m).color);
end

%% FUNCIO QUE S'EXECUTA AL CLICAR EL BOTÓ DE CANVI DE COLOR.

function color(hObject, eventdata, handles)
    c=uisetcolor(barres(n).color); % Obre la finestra de Matlab per seleccionar color.
    set(buttoncolor, 'BackgroundColor', c); % Li dona el color seleccionat al botó.
end

%% FUNCIO QUE S'EXECUTA AL PREMIER ALGUNA TECLA.
% Si es prem el retorn surt actualitzant les dades, si es prem escapament surt sense actualitzar.

function tecla(hObject, eventdata, handles)
    switch eventdata.Key
        case 'escape'
            cancelar
        case 'return'
            uicontrol(hObject)
            acceptar
    end
end
end

```

```

% Funció dlgcarreganus per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 24 de maig de 2.012.
%
% Crea una finestra per introduir-hi o modificar les càrregues sobre les barres.
%
% Entrada:
% - n: objecte tipus nus on s'emmagatzemen les propietats i les càrregues del nus als següents camps:
% + num: Número de barra.
% + puntual: Càrrega puntual aplicada al nus.
% + moment: Moments aplicats al nus.
%
% Sortida:
% - n: Igual que la entrada però convenientment modificades les càrregues del nus editat.

function n = dlgcarreganus(n)
%% CREA LA FINESTRA, ELS PANELLS I ELS BOTONS.

tamanycme2=get(gcf,'Position');
tm=[255,265];
sc=(tamanycme2(3:4)-tm(1:2))/2+tamanycme2(1:2);
bg=get(0,'defaultUicontrolBackgroundColor');
f = figure('Units','pixels','Position',[sc,tm],'NumberTitle','off','Name',['Càrrega al nus nº: ',
num2str(n.num)],'ToolBar','none','MenuBar','none','Resize','off','WindowStyle','modal','color',
bg,'KeyPressFcn',@tecla);
panelpunt=uipanel(f,'Units','pixels','Position',[5 155 245 105],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Càrregues puntuals:'); % Panell per
introduir forces.
panelmom=uipanel(f,'Units','pixels','Position',[5 45 245 105],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Moments:'); % Panell per introduir
moments.
uicontrol(f,'Units','pixels','Position',
[5,10,110,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Acceptar','Callback',@acceptar); % Botó acceptar.
uicontrol(f,'Units','pixels','Position',
[140,10,110,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Cancelar','Callback',@cancelar); % Botó cancelar.

%% ETIQUETES I CAIXES DE TEXTE DEL PANELL PER INTRODUIR MOMENTS

uicontrol(panelmom,'Units','pixels','Position',
[5,65,35,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Mx=');
uicontrol(panelmom,'Units','pixels','Position',
[5,35,35,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','My=');
uicontrol(panelmom,'Units','pixels','Position',
[5,5,35,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Mz=');
editmom(1)=uicontrol(panelmom,'Units','pixels','Position',
[40,65,200,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(n.moment(1)),
'KeyPressFcn',
@tecla);
editmom(2)=uicontrol(panelmom,'Units','pixels','Position',
[40,35,200,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(n.moment(2)),
'KeyPressFcn',
@tecla);
editmom(3)=uicontrol(panelmom,'Units','pixels','Position',
[40,5,200,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(n.moment(3)),
'KeyPressFcn',
@tecla);

%% ETIQUETES I CAIXES DE TEXTE DEL PANELL PER INTRODUIR FORCES

uicontrol(panelpunt,'Units','pixels','Position',
[5,65,35,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Fx=');
uicontrol(panelpunt,'Units','pixels','Position',
[5,35,35,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Fy=');
uicontrol(panelpunt,'Units','pixels','Position',
[5,5,35,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Fz=');
editpunt(1)=uicontrol(panelpunt,'Units','pixels','Position',
[40,65,200,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(n.puntual
(1)),
'KeyPressFcn',
@tecla);
editpunt(2)=uicontrol(panelpunt,'Units','pixels','Position',
[40,35,200,20],'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(n.puntual
(2)),
'KeyPressFcn',
@tecla);
editpunt(3)=uicontrol(panelpunt,'Units','pixels','Position',

```

```

[40,5,200,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize', 14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'String', num2str(n.puntual(3)), 'KeyPressFcn', @tecla);

uiwait(f) % Atura l'execució de Matlab fins que es tanqui la finestra i es retornin les dades. Necessari per tal de que es retornin les dades correctament.

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BUTÓ ACCEPTAR.
% Llegeix les dades de les caixes de text per retornarles i tanca la finestra.

function acceptar(hObject, eventdata, handles)
    for j=1:3
        n.puntual(j)=str2double(get(editpunt(j), 'String'));
        n.moment(j)=str2double(get(editmom(j), 'String'));
    end

    close(f)
end

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BUTÓ CANCELAR.
% Tanca la finestra sense actualitzar les dades.

function cancelar(hObject, eventdata, handles)
    close(f)
end

%% FUNCIO QUE S'EXECUTA AL PREMIER ALGUNA TECLA.
% Si es prem el retorn surt actualitzant les dades, si es prem escapament surt sense actualitzar.

function tecla(hObject, eventdata, handles)
    switch eventdata.Key
        case 'escape'
            cancelar
        case 'return'
            uicontrol(hObject)
            acceptar
    end
end
end

```

```

% Funció dlgcarregabarra per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 24 de maig de 2.012.
%
% Crea una finestra per introduir-hi o modificar les càrregues sobre les barres.
%
% Entrada:
% - bar: objecte tipus barra on s'emmagatzemen les propietats i les càrregues de la barra als següents
camps:
% + num: Número de barra.
% + nusinf: Número de nus inferior.
% + nussup: Número de nus superior.
% + puntual: Càrrega puntual: mòdul, vector direcció x,y,z i distància al nus inferior.
% + moment: Moment aplicat a la barra: mòdul, vector direcció x,y,z i distància al nus inferior.
% + repartida: Càrrega repartida: mòdul, vector direcció x,y,z i distàncies al nus inferior i superior.
% + projectada: Càrrega repartida peojectada: mòdul, vector direcció x,y,z i distàncies al nus inferior
i superior.
%
% Sortida:
% - bar: Igual que la entrada però convenientment modificades les càrregues de la barra editada.

function bar = dlgcarregabarra(bar)
%% CREA LA FINESTRA, ELS PANELLS I ELS BOTONS.

tamanycme2=get(gcf, 'Position');
tm=[330,545];
sc=(tamanycme2(3:4)-tm(1:2))/2+tamanycme2(1:2);
bg=get(0,'defaultUicontrolBackgroundColor');
f = figure('Units','pixels','Position',[sc,tm],'NumberTitle','off','Name',['Càrrega sobre la barra n°:
',num2str(bar.num)],'Toolbar','none','MenuBar','none','Resize','off','WindowStyle','modal','color',
bg,'KeyPressFcn',@tecla);
panelpunt=uipanel(f,'Units','pixels','Position',[5 435 320 105],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Càrrega puntual:'); % Panell
càrregues puntuals
panelmom=uipanel(f,'Units','pixels','Position',[5 325 320 105],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Moment:'); % Panell moments
panelrep=uipanel(f,'Units','pixels','Position',[5 185 320 135],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Càrregues repartides sobre la
barra:'); % Panell càrregues repartides
panelproj=uipanel(f,'Units','pixels','Position',[5 45 320 135],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Càrregues repartides projectades:');
% Panell càrregues projectades
uicontrol(f,'Units','pixels','Position',
[5,10,110,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Acceptar','Callback',@acceptar); % Botó acceptar
uicontrol(f,'Units','pixels','Position',
[215,10,110,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Cancelar','Callback',@cancelar); % Botó cancelar

%% CREA ETIQUETES I CAIXES DE TEXTE ALS PANELLS

txti=['Distància al nus n° ',num2str(bar.nusinf),':'];
txts=['Distància al nus n° ',num2str(bar.nussup),':'];
strlabel={'Mòdul:', 'x:', 'y:', 'z:', txti, txts};
poslabel=[5,35,70,20;115,5,15,20;185,5,15,20;255,5,15,20;5,65,150,20;5,95,150,20];
posedit=[200,35,115,20;130,5,45,20;200,5,45,20;270,5,45,20;200,65,115,20;200,95,115,20];

editpunt=zeros(1,5);
editmom=editpunt;
for i=1:5
    uicontrol(panelpunt,'Units','pixels','Position',poslabel(i,1:
4),'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String',strlabel(i));
    uicontrol(panelmom,'Units','pixels','Position',poslabel(i,1:
4),'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String',strlabel(i));
    editpunt(i)=uicontrol(panelpunt,'Units','pixels','Position',posedit(i,1:
4),'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(bar.puntual
(i)), 'KeyPressFcn',@tecla);
    editmom(i)=uicontrol(panelmom,'Units','pixels','Position',posedit(i,1:
4),'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(bar.moment
(i)), 'KeyPressFcn',@tecla);
end
uicontrol(panelpunt,'Units','pixels','Position',
[5,5,110,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Vector direcció:');
uicontrol(panelmom,'Units','pixels','Position',
[5,5,110,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Vector direcció:');

```



```

editrep=zeros(1,6);
editproj=editrep;
for i=1:6
    uicontrol(panelrep,'Units','pixels','Position',poslabel(i,1:
4),'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String',strlabel(i));
    uicontrol(panelproj,'Units','pixels','Position',poslabel(i,1:
4),'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String',strlabel(i));
    editrep(i)=uicontrol(panelrep,'Units','pixels','Position',posedit(i,1:
4),'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(bar.repartida
(i)), 'KeyPressFcn',@tecla);
    editproj(i)=uicontrol(panelproj,'Units','pixels','Position',posedit(i,1:
4),'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','right','BackgroundColor','white','String',num2str(bar.projectada
(i)), 'KeyPressFcn',@tecla);
end
uicontrol(panelrep,'Units','pixels','Position',
[5,5,110,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Vector direcció:');
uicontrol(panelproj,'Units','pixels','Position',
[5,5,110,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Vector direcció:');

uiwait(f) % Atura l'execució de Matlab fins que es tanqui la finestra i es retornin les dades. Necessari
per tal de que es retornin les dades corrèctament.

%% FUNCIÓ QUE S'EXECUTARÀ QUAN ES PETJI EL BOTÓ ACCEPTAR.
% Llegeix les dades de les caixes de text per retornarles i tanca la finestra.

function acceptar(hObject, eventdata, handles)
    for j=1:5
        bar.puntual(j)=str2double(get(editpunt(j), 'String'));
        bar.moment(j)=str2double(get(editmom(j), 'String'));
        bar.repartida(j)=str2double(get(editrep(j), 'String'));
        bar.projectada(j)=str2double(get(editproj(j), 'String'));
    end
    bar.repartida(6)=str2double(get(editrep(6), 'String'));
    bar.projectada(6)=str2double(get(editproj(6), 'String'));

    close(f)
end

%% FUNCIÓ QUE S'EXECUTARÀ QUAN ES PETJI EL BOTÓ CANCELAR.
% Tanca la finestra sense actualitzar les dades.

function cancelar(hObject, eventdata, handles)
    close(f)
end

%% FUNCIÓ QUE S'EXECUTA AL PREMIER ALGUNA TECLA.
% Si es prem el retorn surt actualitzant les dades, si es prem escapament surt sense actualitzar.

function tecla(hObject, eventdata, handles)
    switch eventdata.Key
        case 'escape'
            cancelar
        case 'return'
            uicontrol(hObject)
            acceptar
    end
end
end
end

```

```

% Funció dlgoptions per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 24 de maig de 2.012.
%
% Crea una finestra per introduir-hi o modificar les opcions del programa.
%
% Carrega les opcions de l'arxiu opcions.mat, si no existeix fa servir per defecte.
% Al sortit guarda les opcions al mateix arxiu, no permet cancel·lar si l'arxiu no existeix.

function dlgoptions
%% CREA LA FINESTRA, ELS PANELLS I ELS BOTONS.
set(0,'Units','pixels') % Per obtenir el tamany de la pantalla en pixels i no en cap altre mesura
extraña.
sc=get(0,'ScreenSize'); % Obté el tamany de la pantalla.
tm=[500,565]; % Tamany de la finestra.
sc=(sc(3:4)-tm(1:2))/2+sc(1:2); % Posició de la finestra al centre de la pantalla.
bg=get(0,'defaultUicontrolBackgroundColor'); % Color de fons per defecte.

f=figure('Units','pixels','Position',[sc,
tm],'NumberTitle','off','Name','Opcions','WindowStyle','modal','color',bg,'CloseRequestFcn',
@tancar,'KeyPressFcn',@tecla); % Crea la finestra principal del programa.
panelopcions=uipanel(f,'Units','pixels','Position',[5 485 490
75],'BorderType','etchedin','BorderWidth',1,'FontName','default','FontUnits','pixels','FontSize',
14,'Title','Opcions:');
paneldir=uipanel(f,'Units','pixels','Position',[5 430 490 50],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Carpeta on guardar i carregar:');
panelfonts=uipanel(f,'Units','pixels','Position',[5 350 490 75],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Fonts:');
panelcolors=uipanel(f,'Units','pixels','Position',[5 45 240 300],'BorderType','etchedin','BorderWidth',
1,'FontName','default','FontUnits','pixels','FontSize',14,'Title','Colors:');
panelescala=uipanel(f,'Units','pixels','Position',[255 45 240
300],'BorderType','etchedin','BorderWidth',1,'FontName','default','FontUnits','pixels','FontSize',
14,'Title','Escala:');
uicontrol(f,'Units','pixels','Position',
[65,5,120,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Acceptar','Callback',@acceptar); % Botó acceptar.
uicontrol(f,'Units','pixels','Position',
[315,5,120,30],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Cancelar','Callback',@cancelar); % Botó cancelar.

%% PANEL·L OPCIONS
checkedit=uicontrol(panelopcions,'Units','pixels','Position',
[5,30,480,20],'Style','checkbox','FontName','default','FontUnits','pixels','FontSize',
14,'String','Editar nusos i barres després de crear-los','KeyPressFcn',@tecla);
checkautobar=uicontrol(panelopcions,'Units','pixels','Position',
[5,5,480,20],'Style','checkbox','FontName','default','FontUnits','pixels','FontSize',14,'String','Al·l
crear un nus unir-lo automaticament amb l'anterior amb una barra','KeyPressFcn',@tecla);

%% PANEL·L CARPETA
editdir=uicontrol(paneldir,'Units','pixels','Position',
[5,5,400,25],'Style','edit','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','BackgroundColor','white','KeyPressFcn',@tecla);
uicontrol(paneldir,'Units','pixels','Position',
[405,5,80,25],'Style','pushbutton','FontName','default','FontUnits','pixels','FontSize',
14,'String','Explorar','Callback',@dir,'KeyPressFcn',@tecla);

%% PANEL·L FONTS
uicontrol(panelfonts,'Units','pixels','Position',
[5,30,100,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Font nusos:');
uicontrol(panelfonts,'Units','pixels','Position',
[5,5,100,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',
14,'HorizontalAlignment','left','String','Font forces:');
buttonfont(1)=uicontrol(panelfonts,'Units','pixels','Position',
[105,30,380,25],'Style','pushbutton','HorizontalAlignment','right','Callback',@font,'KeyPressFcn',
@tecla);
buttonfont(2)=uicontrol(panelfonts,'Units','pixels','Position',
[105,5,380,25],'Style','pushbutton','HorizontalAlignment','right','Callback',@font,'KeyPressFcn',
@tecla);

%% PANEL·L COLORS
txtcolor={'Càrregues:', 'Reaccions:', 'Deformada:', 'Diagrama N:', 'Diagrama Ty:', 'Diagrama Tz:', 'Diagrama
Mx:', 'Diagrama My:', 'Diagrama Mz:', 'Modificar:', 'Esborrar:'};
buttoncolor=zeros(1,11);
for i=1:11
    uicontrol(panelcolors,'Units','pixels','Position',[5,5+25*(11-i),
100,20],'Style','text','FontName','default','FontUnits','pixels','FontSize',

```

```

14, 'HorizontalAlignment', 'left', 'String', txtcolor(i));
    buttoncolor(i)=uicontrol(panelcolors, 'Units', 'pixels', 'Position', [105,5+25*(11-i),
130,20], 'Style', 'pushbutton', 'Callback', @color, 'KeyPressFcn', @tecla);
end

%% PANELL ESCALA

txtescala={'Factor de zoom:', 'Escala nusos i forces:', 'Separaci6 n° nus:', 'Precisi6 inicial:', 'X mìnima
inicial:', 'X màxima inicial:', 'Y mìnima inicial:', 'Y màxima inicial:', 'Z mìnima inicial:', 'Z màxima
inicial:'};
editescala=zeros(1,10);
for i=1:10
    uicontrol(panelescala, 'Units', 'pixels', 'Position', [5,5+25*(11-i),
150,20], 'Style', 'text', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'left', 'String', txtescala(i));
    editescala(i)=uicontrol(panelescala, 'Units', 'pixels', 'Position', [155,5+25*(11-i),
80,20], 'Style', 'edit', 'FontName', 'default', 'FontUnits', 'pixels', 'FontSize',
14, 'HorizontalAlignment', 'right', 'BackgroundColor', 'white', 'KeyPressFcn', @tecla);
end

%% CÀRREGA DE DADES

if exist('opcions.mat', 'file')==2 % Si existeix l'arxiu opcions.mat el carrega.
    load('opcions.mat')
else % Valors per defecte si no troba l'arxiu amb les opcions guardades.
    editalcrear=0;
    barraalcrearnus=0;
    fontnusos='Times New Roman';
    tamanyfontnusos=20;
    fontcarregues='SansSerif';
    tamanyfontcarregues=16;
    factorzoom=10;
    tamanynus=100;
    distnumnus=0.35;
    precini=0.5;
    XLimini=[-5,5];
    YLimini=[-2.5,2.5];
    ZLimini=[-5,5];
    colormodificar=[0,1,1]; % Cian
    coloresborrar=[1,0,0]; % Vermell
    colorcarregues=[.5,0,0];
    colorreaccions=[0,.5,0];
    colordeformada=[0.078431372549020,0.168627450980392,0.549019607843137];
    colorN=[0.043137254901961,0.517647058823530,0.780392156862745];
    colorTy=[.5,0,1];
    colorTz=[1,.25,.5];
    colorMx=[1,1,0];
    colorMy=[.875,.625,.25];
    colorMz=[1,0,0];
    direct=matlabroot;
end

if exist(direct, 'dir')==7 % Comproba si existeix la carpeta.
    set(editdir, 'String', direct);
else % Si no existeix el directori, el canvia pel directori per defecte de Matlab.
    direct=matlabroot;
    set(editdir, 'String', direct);
end
colors=[colorcarregues;colorreaccions;colordeformada;colorN;colorTy;colorTz;colorMx;colorMy;colorMz;
colormodificar;coloresborrar]; % Carrega els colors als botons.
for i=1:11
    set(buttoncolor(i), 'BackgroundColor', colors(i,:));
end

% Carrega les dades a les caixes i botons corresponents.
set(checkedit, 'Value', editalcrear);
set(checkautobar, 'Value', barraalcrearnus);
set(buttonfont(1), 'String', [fontnusos, ' ', num2str(tamanyfontnusos)], 'UserData',
tamanyfontnusos, 'FontName', fontnusos, 'FontUnits', 'pixels', 'FontSize', 14);
set(buttonfont(2), 'String', [fontcarregues, ' ', num2str(tamanyfontcarregues)], 'UserData',
tamanyfontcarregues, 'FontName', fontcarregues, 'FontUnits', 'pixels', 'FontSize', 14);
set(editescala(1), 'String', factorzoom);
set(editescala(2), 'String', tamanynus);
set(editescala(3), 'String', distnumnus);
set(editescala(4), 'String', precini);
set(editescala(5), 'String', XLimini(1));
set(editescala(6), 'String', XLimini(2));
set(editescala(7), 'String', YLimini(1));
set(editescala(8), 'String', YLimini(2));
set(editescala(9), 'String', ZLimini(1));
set(editescala(10), 'String', ZLimini(2));

```

uiwait(f) % Atura l'execució de Matlab fins que es tanqui la finestra i es retornin les dades. Necessari
per tal de que es retornin les dades correctament.

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BUTÓ ACCEPTAR.

% Llegeix les dades de les caixes de text, les guarda a l'arxiu opcions.mat i tanca la finestra.

```
function acceptar(hObject, eventdata, handles)
    editalcrear=get(checkedit, 'Value');
    barraalcrearnus=get(checkautobar, 'Value');
    direct=get(editdir, 'String');
    colorcarregues=get(buttoncolor(1), 'BackgroundColor');
    colorreaccions=get(buttoncolor(2), 'BackgroundColor');
    colordeformada=get(buttoncolor(3), 'BackgroundColor');
    colorN=get(buttoncolor(4), 'BackgroundColor');
    colorTy=get(buttoncolor(5), 'BackgroundColor');
    colorTz=get(buttoncolor(6), 'BackgroundColor');
    colorMx=get(buttoncolor(7), 'BackgroundColor');
    colorMy=get(buttoncolor(8), 'BackgroundColor');
    colorMz=get(buttoncolor(9), 'BackgroundColor');
    colormodificar=get(buttoncolor(10), 'BackgroundColor');
    coloresborrar=get(buttoncolor(11), 'BackgroundColor');

    fontnusus=get(buttonfont(1), 'FontName');
    tamanyfontnusus=get(buttonfont(1), 'UserData');
    fontcarregues=get(buttonfont(2), 'FontName');
    tamanyfontcarregues=get(buttonfont(2), 'UserData');

    factorzoom=str2double(get(editescala(1), 'String'));
    tamanynus=str2double(get(editescala(2), 'String'));
    distnumnus=str2double(get(editescala(3), 'String'));
    precini=str2double(get(editescala(4), 'String'));

    XLimini=[str2double(get(editescala(5), 'String')),str2double(get(editescala(6), 'String'))];
    YLimini=[str2double(get(editescala(7), 'String')),str2double(get(editescala(8), 'String'))];
    ZLimini=[str2double(get(editescala(8), 'String')),str2double(get(editescala(10), 'String'))];

    save
    ('opcions', 'editalcrear', 'barraalcrearnus', 'fontnusus', 'tamanyfontnusus', 'fontcarregues', 'tamanyfontcarregues', 'factorzoom', 'tamanynus', 'distnumnus', 'precini', 'XLimini', 'YLimini', 'ZLimini', 'colormodificar', 'coloresborrar', 'colorcarregues', 'colorreaccions', 'colordeformada', 'colorN', 'colorTy', 'colorTz', 'colorMx', 'colorMy', 'colorMz', 'direct')

    close(f)
end

%% FUNCIO QUE S'EXECUTARÀ QUAN ES PETJI EL BUTÓ CANCELAR.
% Tanca la finestra sense actualitzar les dades.
% No es permet si no existeix l'arxiu opcions.mat.
```

```
function cancelar(hObject, eventdata, handles)
    close(f)
end
function tancar(hObject, eventdata, handles)
    if exist('opcions.mat', 'file')==2
        delete(f)
    else
        return
    end
end
function dir(hObject, eventdata, handles)
    d=uigetdir(get(editdir, 'String'));
    set(editdir, 'String', d);
end
function font(hObject, eventdata, handles)
    fo=uisetfont(hObject);
    set(hObject, 'String', [fo.FontName, ' ', num2str(fo.FontSize)], 'UserData', fo.FontSize, 'FontName', fo.FontName, 'FontUnits', 'pixels', 'FontSize', 14);
end
function color(hObject, eventdata, handles)
    set(hObject, 'Backgroundcolor', uisetcolor(get(hObject, 'BackgroundColor')))
end
```

%% FUNCIO QUE S'EXECUTA AL PREMIER ALGUNA TECLA.

% Si es prem el retorn surt actualitzant les dades, si es prem escapament surt sense actualitzar.

```
function tecla(hObject, eventdata, handles)
    switch eventdata.Key
        case 'escape'
            cancelar
        case 'return'
            uicontrol(hObject)
            acceptar
```

end
end
end

```

% Crea objectes gràfics del tipus 'hgtransform' amb el dibuix del rodet, encastament, etc..
% Retorna el handle, es poden rotar i moure a la gràfica amb matriu creada amb la funció
% makehgtform.

```

```

function [contorn] = dibuix_contorn(tipus,distnum,fontnusus,tamanyfontnusus)
contorn=hgtransform;
text(-distnum,distnum,'0','FontName',fontnusus,'FontUnits','pixels','FontSize',
tamanyfontnusus,'HorizontalAlignment','right','Parent',contorn,'Clipping','on');
switch tipus
case {'nus',0}
line([-0.20,.20],[0,0],[0,0],'color','k','Parent',contorn,'Clipping','on','HitTest','off');
line([0,0],[-0.20,.20],[0,0],'color','k','Parent',contorn,'Clipping','on','HitTest','off');
line([0,0],[0,0],[-0.20,.20],'color','k','Parent',contorn,'Clipping','on','HitTest','off');
case {'rodet',1}
cercle(contorn,-0.04,-0.18,0.04); % Cercle esquerra.
cercle(contorn,0.04,-0.18,0.04); % Cercle dreta.
line([-0.08,0,0.08,-0.08],[-0.14,0,-0.14,-0.14],'color','k','Parent',
contorn,'Clipping','on','HitTest','off'); % Triangle.
massis(contorn,-0.14,0.14,-0.22);
case {'simple',2}
cercle(contorn,0,0,0.030); % Cercle.
line([-0.08,0,0.08],[-0.170,-0.030,-0.170],'color','k','Parent',
contorn,'Clipping','on','HitTest','off'); % Triangle.
massis(contorn,-0.16,0.16,-0.170);
case {'encastament',3}
massis(contorn,-0.16,0.16,0);
case {'despelastic',4}
line([0,-0.08,0.08,-0.08,0.08,-0.08,0.08,0],[-0.46,-0.44,-0.40,-0.36,-0.32,-0.28,-0.24,-0.20,-0.16,-0.14],
'color','k','Linewidth',2,'Parent',contorn,'Clipping','on','HitTest','off') % Molla
line([-0.08,0,0.08,-0.08],[-0.14,0,-0.14,-0.14],'color','k','Parent',
contorn,'Clipping','on','HitTest','off'); % Triangle.
massis(contorn,-0.16,0.16,-0.46);
case {'girelastic',5}
cercle(contorn,0,0,0.030); % Cercle.
line([-0.08,0,0.08],[-0.170,-0.030,-0.170],'color','k','Parent',
contorn,'Clipping','on','HitTest','off'); % Triangle.
massis(contorn,-0.16,0.32,-0.170);
line([0.24,.242,0.24,.2350,.2250,.2124,.20,.18,.16,.14,.12,.110,.10,.098,.1050,.12,.130,.1384,.14,.1350,.12,.10,.08,.04,0],
[-0.170,-0.12,-0.070,-0.0350,0,.0250,.04,.054,.06,.058,.050,.0432,.030,.0150,0,-0.0050,0,0.0150,.030,.0550,.08,.094,.10,.10,.0950],
'color','k','Parent',
contorn,'Clipping','on','HitTest','off'); % Molla.
case {'despigirelastic',6}
line([0,-0.08,0.08,-0.08,0.08,-0.08,0.08,0],[-0.46,-0.44,-0.40,-0.36,-0.32,-0.28,-0.24,-0.20,-0.16,-0.14],
'color','k','Linewidth',2,'Parent',contorn,'Clipping','on','HitTest','off') % Molla
line([-0.08,0,0.08,-0.08],[-0.14,0,-0.14,-0.14],'color','k','Parent',
contorn,'Clipping','on','HitTest','off'); % Triangle.
massis(contorn,-0.16,0.32,-0.46);
line([0.24,.242,0.24,.2350,.2250,.2124,.20,.18,.16,.14,.12,.110,.10,.098,.1050,.12,.130,.1384,.14,.1350,.12,.10,.08,.04,0],
[-0.46,-0.36,-0.26,-0.190,-0.12,-0.070,-0.04,-0.012,0,-0.004,-0.02,-0.0336,-0.06,-0.090,-0.12,-0.130,-0.12,-0.090,-0.06,-0.010,.04,.068,.08,.08,.070],
'color','k','Parent',
contorn,'Clipping','on','HitTest','off'); % Molla.12
case {'despimposat',7}
cercle(contorn,0,0,0.030)
line([0.20,0,0],[0,0,.20],'color','k','LineStyle','--','Parent',
contorn,'Clipping','on','HitTest','off')
line([0.20,.20,0],[0,.20,.20],'color','k','Parent',contorn,'Clipping','on','HitTest','off')
patch([0.20,.22,.18,.20,.16,.20],[0.20,.16,.16,.20,.18,.22,.20],'k','Parent',
contorn,'Clipping','on','HitTest','off')
case {'girimposat',8}
cercle(contorn,0,0,0.030)
line([-0.40,0,-0.3464],[0,0,-0.20],'color','k','LineStyle','--','Parent',
contorn,'Clipping','on','HitTest','off')
a=0:pi/30:pi/6;
line(-0.30*cos(a),-0.30*sin(a),'color','k','Parent',contorn,'Clipping','on','HitTest','off')
patch([-0.30*cos(pi/6),-0.30*cos(pi/6)-sin(0.987)*0.0448,-0.30*cos(pi/6)-sin(0.06)*0.0448],[-0.30*sin(pi/6),-0.30*sin(pi/6)+cos(0.987)*0.0448,-0.30*sin(pi/6)+cos(0.06)*0.0448],
'k','Parent',
contorn,'Clipping','on','HitTest','off')
case {'despigirimposats',9}
cercle(contorn,0,0,0.030)
line([0.20,0,0],[0,0,.20],'color','k','LineStyle','--','Parent',
contorn,'Clipping','on','HitTest','off')
line([0.20,.20,0],[0,.20,.20],'color','k','Parent',contorn,'Clipping','on','HitTest','off')
line([-0.40,0,-0.3464],[0,0,-0.20],'color','k','LineStyle','--','Parent',
contorn,'Clipping','on','HitTest','off')
a=0:pi/30:pi/6;
line(-0.30*cos(a),-0.30*sin(a),'color','k','Parent',contorn,'Clipping','on','HitTest','off')
patch([0.20,.22,.18,.20,.16,.16,.20],[0.20,.16,.16,.20,.18,.22,.20],'k','Parent',
contorn,'Clipping','on','HitTest','off')
patch([-0.30*cos(pi/6),-0.30*cos(pi/6)-sin(0.987)*0.0448,-0.30*cos(pi/6)-sin(0.06)*0.0448],[-0.30*sin(pi/6),-0.30*sin(pi/6)+cos(0.987)*0.0448,-0.30*sin(pi/6)+cos(0.06)*0.0448],
'k','Parent',
contorn,'Clipping','on','HitTest','off')
end

```

```

set(contorn, 'Clipping', 'on', 'HitTestArea', 'on', 'HitTest', 'off');

end

function [] = massis(h,xini,xfin,y)
line([xini,xfin],[y,y], 'color', 'k', 'LineWidth', 2, 'Parent', h, 'Clipping', 'on', 'HitTest', 'off'); % Línia
secció horitzontal.
line([xini,xini+.04],[y-.04,y], 'color', 'k', 'Parent', h, 'Clipping', 'on', 'HitTest', 'off');
line([xini,xini+.08],[y-.08,y], 'color', 'k', 'Parent', h, 'Clipping', 'on', 'HitTest', 'off');
for x=xini:.04:xfin-.12
    line([x,x+.12],[y-.12,y], 'color', 'k', 'Parent', h, 'Clipping', 'on', 'HitTest', 'off');
end
line([xfin-.08,xfin],[y-.12,y-.04], 'color', 'k', 'Parent', h, 'Clipping', 'on', 'HitTest', 'off');
line([xfin-.04,xfin],[y-.12,y-.08], 'color', 'k', 'Parent', h, 'Clipping', 'on', 'HitTest', 'off');
end

function [] = cercle(h,x,y,r)
c=0:pi/12:2*pi;
line(r*sin(c)-x,r*cos(c)+y, 'color', 'k', 'Parent', h, 'Clipping', 'on', 'HitTest', 'off');
end

```



```

% Funció dibuix_carreganus per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 13 d'octubre de 2.012.
%
% Crea objectes gràfics del tipus hgtransform amb el dibuix de les
% càrregues puntuals i moments aplicats als nusos o les reaccions.
% Retorna el handle, per a col·locar i escalar a la gràfica.
% Càrregues independents per a cada eix.
%
% Entrada:
% - n: número del nus.
% - posició: posició del nus a l'espai.
% - centre: mitja de la posició de tots els nusos.
% - forces: vector de forces aplicades al nus.
% - moments: vector de moments aplicats al nus.
% - vista: 'xy','xz','yz' o '3d' per a la seva correcta representació.
% - fontcarregues: font dels textos de les càrregues.
% - tamanyfontcarregues: tamany de la font;
%
% Sortida:
% - carrega: handle de l'objecte hgtransform amb la representació de les càrregues.
%
% Funcionament:
% Es parteix de les coordenades dels punts per dibuixar una fletxa al plànol xy amb direcció i sentit +
x.
% Es treballa al plànol xy com si fòs el de vista.
% Es roten els punts al plànol de la vista corresponent i es dibuixa la fletxa.
% De manera idèntica per als moments.
% No dibuixa les forces al plànol perpendicular al de la vista, sí els moments.

function hgcarrega = dibuix_carreganus(n,posicio,centre,forces,moments,vista,fontcarregues,
tamanyfontcarregues,fmax)

    tol=1e-8; % Valor mínim, per evitar mostrar errors d'arrodoniment a 0.

    if any(forces) || any(moments)
        hgcarrega=hgtransform('HitTestArea','on','Clipping','on','UserData',n); % Crea l'objecte
hgtransform.

        switch vista
            case 'xy'
                dibuixa_forces(hgcarrega,tol,forces(1:2),posicio(1:2)-centre(1:2),fontcarregues,
tamanyfontcarregues,[1,0,0;0,1,0;0,0,1],{'Fx','Fy'},fmax) % Dibuixa les forces al plànol xy.
                dibuixa_moments(hgcarrega,tol,moments(1:2),moments(3),posicio(1:2)-centre(1:2),
fontcarregues,tamanyfontcarregues,[1,0,0;0,1,0;0,0,1],{'Mx','My','Mz'}) % Dibuixa els moments al plànol
xy.
            case 'xz'
                dibuixa_forces(hgcarrega,tol,[forces(1),forces(3)],[posicio(1)-centre(1),posicio(3)-
centre(3)],fontcarregues,tamanyfontcarregues,[1,0,0;0,0,-1;0,1,0],{'Fx','Fz'},fmax) % Dibuixa les
forces al plànol xz.
                dibuixa_moments(hgcarrega,tol,[moments(1),moments(3)],moments(2),[posicio(1)-centre(1),
posicio(3)-centre(3)],fontcarregues,tamanyfontcarregues,[1,0,0;0,0,-1;0,1,0],{'Mx','Mz','My'}) %
Dibuixa els moments al plànol xz.
            case 'yz'
                dibuixa_forces(hgcarrega,tol,forces(2:3),posicio(2:3)-centre(2:3),fontcarregues,
tamanyfontcarregues,[0,0,1;1,0,0;0,0,1],{'Fy','Fz'},fmax) % Dibuixa les forces al plànol yz.
                dibuixa_moments(hgcarrega,tol,moments(2:3),moments(1),posicio(2:3)-centre(2:3),
fontcarregues,tamanyfontcarregues,[0,0,1;1,0,0;0,0,1],{'My','Mz','Mx'}) % Dibuixa els moments al plànol
yz.
            case '3d'
                dibuixa_forces(hgcarrega,tol,forces(1:3),posicio(1:3)-centre(1:3),fontcarregues,
tamanyfontcarregues,[1,0,0;0,1,0;0,0,1],{'Fx','Fy','Fz'},fmax) % Dibuixa les forces a l'espai.
                dibuixa_moments(hgcarrega,tol,moments(1:3),0,0,fontcarregues,tamanyfontcarregues,
[1,0,0;0,1,0;0,0,1],{'Mx','My','Mz'}) % Dibuixa els moments a l'espai.
        end
    else
        hgcarrega=[];
    end
end

function dibuixa_forces(handle,tol,forcesxy,posxy,fontcarregues,tamanyfontcarregues,matriu_canvi_vista,
txt,fmax)
    % Dibuixa les forces al plànol xy i després aplica una matriu de rotació per adaptar-les al plànol
de visualització.
    if abs(forcesxy(1))>tol % Forces en x.
        [postxt,halin]=dibuixa_forca(handle,forcesxy(1),posxy(1),matriu_canvi_vista,fmax);
        text(postxt(1),postxt(2),postxt(3),strcat(txt(1),'=',num2str(forcesxy(1))),'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'Parent',
handle,'HorizontalAlignment',halin)
    end
    if abs(forcesxy(2))>tol % Forces en y.
        [postxt,halin]=dibuixa_forca(handle,forcesxy(2),posxy(2),matriu_canvi_vista*[0,1,0;1,0,0;

```

```

0,0,1],fmax);
    text(postxt(1),postxt(2),postxt(3),strcat(txt(2),'=',num2str(forcesxy(2))),'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'Parent',
handle,'HorizontalAlignment',halin)
    end
    if numel(forcesxy)==3 && abs(forcesxy(3))>tol % Forces en z.
        [postxt,halin]=dibuixa_forca(handle,forcesxy(3),posxy(3),matriu_canvi_vista*[0,1,0;0,0,1;
1,0,0],fmax);
        text(postxt(1),postxt(2),postxt(3),strcat(txt(3),'=',num2str(forcesxy(3))),'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'Parent',
handle,'HorizontalAlignment',halin)
    end
end

function [postxt,halin]=dibuixa_forca(handle,forca,pos,mat,fmax)
    % Fletxa definida al plànol xy seguint la direcció de l'eix x, s'aplica una matriu de rotació per
    situar-la a l'eix corresponent.
    prop=abs(forca/fmax); % Longitud de la fletxa, proporcional a la força.
    signe=forca/abs(forca); % Sentit de la força.
    pos=pos/abs(pos); % Posició rel.lativa de la fletxa, >0 dreta, <0 esquerra

    if pos*signe>0 % estirant
        XYlf=[prop+0.33,0.1;0,0]*signe; % Coordenades de la línia de la fletxa que representa la força.
        XYpf=[0.1+prop,0.33+prop,0.1+prop;.0375,0,-.0375]*signe; % Coordenades de la punta de fletxa
        que representa la força.
        if signe>0
            halin='left';
        else
            halin='right';
        end
    else % empenyent
        XYlf=-[prop+0.33,0.33;0,0]*signe; % Coordenades de la línia de la fletxa que representa la
        força.
        XYpf=[-0.33,-0.1,-0.33;.0375,0,-.0375]*signe; % Coordenades de la punta de fletxa que
        representa la força.
        if signe>0
            halin='right';
        else
            halin='left';
        end
    end
    XYlf=mat*[XYlf;0,0];
    XYpf=mat*[XYpf;0,0,0];
    line(XYlf(1,:),XYlf(2,:),XYlf(3,),'color','k','LineWidth',2,'Parent',handle,'HitTest','off');
    patch(XYpf(1,:),XYpf(2,:),XYpf(3,),'k','Parent',handle,'HitTest','off');
    postxt=XYlf(:,1)*1.1;
end

function dibuixa_moments(handle,tol,momentsxy,momentz,posxy,fontcarregues,tamanyfontcarregues,
matriu_canvi_vista,txt)
    % Dibuixa els moments al plànol xy i després aplica una matriu de rotació per adaptar-los al plànol
    de visualització.
    if abs(momentsxy(1))>tol % Moment en x. Fletxa de dos puntes al plànol xy.
        [postxt,halin,valor]=dibuixa_moment_fletxa(handle,momentsxy(1),matriu_canvi_vista);
        text(postxt(1),postxt(2),postxt(3),strcat(txt(1),'=',num2str(momentsxy(1))),'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'Parent',
handle,'HorizontalAlignment',halin,'VerticalAlignment',valor)
    end
    if abs(momentsxy(2))>tol % Moment en y. Fletxa de dos puntes al plànol xy.
        [postxt,halin,valor]=dibuixa_moment_fletxa(handle,momentsxy(2),matriu_canvi_vista*[0,1,0;1,0,0;
0,0,1]);
        text(postxt(1),postxt(2),postxt(3),strcat(txt(2),'=',num2str(momentsxy(2))),'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'Parent',
handle,'HorizontalAlignment',halin,'VerticalAlignment',valor)
    end
    if numel(momentsxy)==3 && abs(momentsxy(3))>tol % Moment en z. Fletxa de dos puntes si es
    representa a l'espai.
        [postxt,halin,valor]=dibuixa_moment_fletxa(handle,momentsxy(3),matriu_canvi_vista*[0,1,0;0,0,1;
1,0,0]);
        text(postxt(1),postxt(2),postxt(3),strcat(txt(3),'=',num2str(momentsxy(3))),'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'Parent',
handle,'HorizontalAlignment',halin,'VerticalAlignment',valor)
    elseif abs(momentz)>tol % Moment en z. Arc si es representa al plànol xy.
        [postxt,halin,valor]=dibuixa_moment_arc(handle,momentz,posxy,matriu_canvi_vista);
        text(postxt(1),postxt(2),postxt(3),strcat(txt(3),'=',num2str(momentz)), 'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'Parent',
handle,'HorizontalAlignment',halin,'VerticalAlignment',valor)
    end
end

function [postxt,halin,valor]=dibuixa_moment_fletxa(handle,moment,mat)

```

```

% Fletxa definida al plànol xy seguint la direcció de l'eix x, s'aplica una matriu de rotació per
situar-la a l'eix corresponent.
signe=moment/abs(moment); % Signe del moment.
XYZlmf=mat* [.5*signe,0;0,0;0,0]; % Coordenades de la línia de la fletxa que representa el moment.
XYZpmf=mat* [.425,.5,.425,.425,.35,.35,.425]*signe;.0375,0,-.0375,0,-.0375,.0375,0;0,0,0,0,0,0];
% Coordenades de la punta de la fletxa que representa el moment.

line(XYZlmf(1,:),XYZlmf(2,:),XYZlmf(3,:), 'color','k','LineWidth',2,'Parent',
handle,'HitTest','off'); % Dibuixa la línia.
patch(XYZpmf(1,:),XYZpmf(2,:),XYZpmf(3,:), 'k','Parent',handle,'HitTest','off'); % Dibuixa la punta
de fletxa.
postxt=XYZlmf(:,1)*1.1; % Posiciona el text.
if signe >0 % Defineix l'alineació del text.
    halin='left';
    valin='bottom';
else
    halin='right';
    valin='top';
end
end

function [postxt,halin,valin]=dibuixa_moment_arc(handle,momentz,posxy,matriu_canvi_vista)
signe=momentz/abs(momentz); % Signe del moment.
if matriu_canvi_vista(2,3)==-1 % Correcció direcció moments en xz, en aquest plànol el moment
positiu és horari.
    signe=-signe;
end
% Gira l'arc segons la seva posició al plànol aplicant una matriu de rotació sobre l'eix z.
if posxy(1)>0 % dreta
    if posxy(2)>0 %dalt
        mat=matriu_canvi_vista;
        valin='bottom';
    else % baix
        mat=matriu_canvi_vista*[0,1,0;-1,0,0;0,0,1];
        valin='top';
    end
    halin='left';
else % esquerra
    if posxy(2)>0 % dalt
        mat=matriu_canvi_vista*[0,-1,0;1,0,0;0,0,1];
        valin='bottom';
    else %baix
        mat=matriu_canvi_vista*[-1,0,0;0,-1,0;0,0,1];
        valin='top';
    end
    halin='right';
end

kk=0.1*pi:pi/20:0.65*pi; % Punts de l'arc.
XYZlm=mat*[cos(kk)*0.3*signe;sin(kk)*0.3+0.15;kk*0]; % Defineix la línia de l'arc.
XYZpm=mat*[[sin(1.15*pi)*.3+sin(5*pi/18)*.0375,sin(1.4*pi)*.3,sin(1.15*pi)*.3-sin(5*pi/18)*.0375]
*signe;-cos(1.15*pi)*.3-cos(5*pi/18)*.0375+.15,-cos(1.4*pi)*.3+.15,-cos(1.15*pi)*.3+cos(5*pi/18)*.
0375+.15;0,0,0]; % Punta de l'arc.
line(XYZlm(1,:),XYZlm(2,:),XYZlm(3,:), 'color','k','LineWidth',2,'Parent',handle,'HitTest','off'); %
Dibuixa la línia.
patch(XYZpm(1,:),XYZpm(2,:),XYZpm(3,:), 'k','Parent',handle,'HitTest','off'); % Dibuixa la punta.
postxt=mat* [.4;.4;0]; % Posiciona el text a l'extrem superior dret de l'arc original.
end

```

```

% Funció dibuix_carregabarra per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 13 d'octubre de 2.012.
%
% Crea objectes gràfics del tipus hgtransform amb el dibuix de les
% càrregues puntuals i moments aplicats a les barres, segons la vista.
% Retorna els handles ja rotats i orientats segons la vista, per a
% col·locar i escalar a la grafica.
%
% Entrada:
% bar: objecte tipus barra, estructura de dades amb les propietats i les càrregues de la barra.
% vectorbarra: vector 1x3 amb la direcció de la barra.
% vista: tipus de vista en format cadena de text, 'xy','yz','xz' o '3d'.
% fontcarregues: tipus de font del text.
% tamanyfontcarregues: tamany de la font.
%
% Sortida:
% puntual: handle de l'objecte gràfic hgtransform creat que representa la càrrega puntual a la barra.
% moment: handle de l'objecte gràfic hgtransform creat que representa el moment a la barra.
%
% Funcionament:
% Es parteix de les coordenades dels punts per dibuixar una fletxa al plànol xy amb direcció i sentit +
x.
% Es rota la projecció de la direcció de la càrrega sobre el plànol de la vista al plànol xy, per
treballar-hi al plànol xy.
% Es roten els punts segons la orientació de la projecció de la càrrega.
% Es roten els punts de nou al plànol de la vista corresponent i es dibuixa la fletxa.
% De manera idèntica per als moments.
% No dibuixa les forces al plànol perpendicular al de la vista, sí els moments.

%% FUNCIO PRINCIPAL:

function [puntual,moment] = dibuix_carregabarra(bar,vectorbarra,vista,fontcarregues,
tamanyfontcarregues,fmax)
    bar.puntual(2:4)=bar.puntual(1)*bar.puntual(2:4)/norm(bar.puntual(2:4)); % Càrrega puntual en forma
de vector.
    bar.moment(2:4)=bar.moment(1)*bar.moment(2:4)/norm(bar.moment(2:4)); % Moment en forma de vector.
    if any(bar.puntual(2:4)) % Si hi ha càrrega puntual.
        puntual=hgtransform('HitTestArea','on','Clipping','on','UserData',bar.num);
        switch vista
            case 'xy'
                dibuixa_forca(puntual,bar.puntual(1),[bar.puntual(2:3),0],vista,fmax); % Dibuixa la
fletxa representant la força al plànol xy.
                posttxt=[-0.7*bar.puntual(2:3)/norm(bar.puntual(2:3)),0];
            case 'xz'
                dibuixa_forca(puntual,bar.puntual(1),[bar.puntual(2),0,bar.puntual(4)],vista,fmax); %
Dibuixa la fletxa representant la força al plànol xz.
                posttxt=-0.7*[bar.puntual(2),0,bar.puntual(4)]/norm([bar.puntual(2),0,bar.puntual(4)]);
            case 'yz'
                dibuixa_forca(puntual,bar.puntual(1),[0,bar.puntual(3:4)],vista,fmax); % Dibuixa la
fletxa representant la força al plànol yz.
                posttxt=[0,-0.7*bar.puntual(3:4)/norm(bar.puntual(3:4))];
            case '3d'
                dibuixa_forca(puntual,bar.puntual(1),bar.puntual(2:4),vista,fmax); % Dibuixa la fletxa
representant la força a l'espai.
                posttxt=-0.7*bar.puntual(2:4)/norm(bar.puntual(2:4));
        end
        texte_carrega(puntual,posttxt, strcat('P=',num2str(bar.puntual(1))),fontcarregues,
tamanyfontcarregues) % Text amb la magnitud de la força.
    else
        puntual=[];
    end

    if any(bar.moment(2:4)) % Si hi ha moment.
        moment=hgtransform('HitTestArea','on','Clipping','on','UserData',bar.num);
        switch vista
            case 'xy'
                posttxt=dibuixa_moment(moment,[bar.moment(2:3),0],bar.moment(4),[vectorbarra(1:2),0],
vista); % Dibuixa la representació del moment al plànol xy.
            case 'xz'
                posttxt=dibuixa_moment(moment,[bar.moment(2),0,bar.moment(4)],bar.moment(3),[vectorbarra
(1),0,vectorbarra(3)],vista); % Dibuixa la representació del moment al plànol xz.
            case 'yz'
                posttxt=dibuixa_moment(moment,[0,bar.moment(3:4)],bar.moment(2),[0,vectorbarra(2:3)],
vista); % Dibuixa la representació del moment al plànol yz.
            case '3d'
                posttxt=dibuixa_moment_fletxa(moment,bar.moment(2:4),vista); % Dibuixa la representació
del moment a l'espai.
        end
        texte_carrega(moment,posttxt, strcat('M=',num2str(bar.moment(1))),fontcarregues,
tamanyfontcarregues) % Text amb la magnitud del moment.
    else

```

```

        moment=[];
    end
end

%% FUNCIONS PER DIBUIXAR FORCES I MOMENTS:

function dibuixa_forca(handle,p,direccio,vista,fmax) % Dibuixa la fletxa que representa la força aplicada a la barra.
    prop=p/fmax;
    XYZlf=[-prop-0.23,0;0,0;0,0]; % Coordenades x,y,z de la recta de la fletxa en horitzontal i cap a la dreta.
    XYZpf=[0,-0.23,-0.23;0,-0.0375,0.0375;0,0,0]; % Coordenades x,y,z de la punta de fletxa en horitzontal i cap a la dreta.
    dibuixa_fletxa3d(XYZlf,XYZpf,handle,direccio,vista); % Dibuixa la fletxa orientant-la en la direccio de la força.
end

function postxt=dibuixa_moment(handle,momentpla,momenttransversal,projbarraalpla,vista) % Dibuixa el moment, arc per al moment transversal i fletxa per als moments al pla.
    if momenttransversal % Si hi ha moment en z.
        postxt=dibuixa_moment_arc(handle,momenttransversal,projbarraalpla,vista); % Dibuixa l'arc del moment.
    end
    if any(momentpla) % Si hi ha moment en x o y.
        postxt=dibuixa_moment_fletxa(handle,momentpla,vista); % Representa el moment amb una fletxa de dos puntes.
    end
end

function postxt=dibuixa_moment_fletxa(pare,direccio,vista) % Representa el moment amb una fletxa de dos puntes.
    XYZlmf=[0,0.5;0,0;0,0]; % Coordenades x,y,z de la recta de la fletxa en horitzontal i cap a la dreta.
    XYZpmf=[0.5,0.425,0.425,0.35,0.35,0.425,0.425;0,-0.0375,0,-0.0375,0.0375,0,0.0375;0,0,0,0,0,0]; % Coordenades x,y,z de la punta de fletxa doble en horitzontal i cap a la dreta.
    postxt=dibuixa_fletxa3d(XYZlmf,XYZpmf,pare,direccio,vista); % Dibuixa el moment orientant-lo en la direccio.
end

function postxt=dibuixa_moment_arc(handle,modul,direccio,vista)
    if ~any(direccio)
        direccio=[1,0,0];
    end
    signe=abs(modul)/modul; % Signe del moment.
    kk=0.3*pi:0.05*pi:1.15*pi; % Arc de moment.
    XYZlm=[0,cos(kk)*0.3;0,sin(kk)*0.3*signe;0,0]; % Coordenades x,y,z de l'arc.
    XYZpm=[cos(1.4*pi)*0.3,cos(1.15*pi)*0.3-cos(5*pi/18)*0.0375,cos(1.15*pi)*0.3+cos(5*pi/18)*0.0375; sin(1.4*pi)*0.3*signe,sin(1.15*pi)*0.3*signe-sin(5*pi/18)*0.0375*signe,sin(1.15*pi)*0.3*signe+sin(5*pi/18)*0.0375*signe;0,0,0]; % Coordenades x,y,z de la punta de fletxa.
    postxt=dibuixa_fletxa3d(XYZlm,XYZpm,handle,direccio,vista); % Dibuixa el moment orientant-lo en la direccio.
end

function postxt=dibuixa_fletxa3d(XYZl,XYZp,handle,direccio,vista)
    % Dibuixa una fletxa, formada per una línia i una punta de fletxa
    % definides per les coordenades introduïdes i orientades segons el vector direccio.
    % Estableix la posició del text a la punta de la fletxa.

    switch vista
        case 'xz'
            m=[1,0,0;0,0,1;0,-1,0]; % Matriu canvi de base de (x,y,z) a (x,z,-y).
        case 'yz'
            m=[0,0,1;1,0,0;0,1,0]; % Matriu canvi de base de (x,y,z) a (x,z,-y).
        otherwise
            m=[1,0,0;0,1,0;0,0,1]; % Matriu identitat.
    end
    direccio=(m*direccio)'; % Rota la direccio al plànol xy.
    mat=calcul_matriu(direccio); % Matriu de gir al plànol xy.
    XYZl=m*mat*XYZl; % Gira la fletxa o l'arc del moment seguint la inclinació de la barra i torna al plànol de la vista.
    XYZp=m*mat*XYZp;

    line(XYZl(1,:),XYZl(2,:),XYZl(3,),'color','k','LineWidth',2,'Parent',handle,'HitTest','off');
    patch(XYZp(1,:),XYZp(2,:),XYZp(3,),'k','Parent',handle,'HitTest','off');

    postxt=XYZp(:,1)*1.25;
end

function texte_carrega(handle,coords,txt,fontcarregues,tamanyfontcarregues) % Texte amb la magnitud de la força.
    text(coords(1),coords(2),coords(3),txt,'VerticalAlignment','middle','Parent',handle,'FontName',fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'HitTest','off')
end

```

```

end

%% CÀLCUL DE LA MATRIU DE CANVI DE BASE:

function matriu_rotacio=calcul_matriu(vector)
    vector=vector/norm(vector); % Normaliza el vector.
    if abs(vector(3))==1 % Si el vector és paral·lel a l'eix z...
        vector3=[vector(3),0,0];
        vector2=[0,1,0];
    else
        vector2=cross([0,0,1],vector); % Calcula dos vectors perpendiculars al primer i entre ells per
    crear la matriu de canvi de base.
        vector2=vector2/norm(vector2);
        vector3=cross(vector,vector2);
        vector3=vector3/norm(vector3);
    end
    matriu_rotacio=[vector',vector2',vector3'];
end

```

```

% Funció dibuix_projectada per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 13 d'octubre de 2.012.
%
% Crea objectes gràfics del tipus hgtransform amb la representació de les
% càrregues projectades sobre la barra.
% Retorna el handle, per a col·locar i escalar a la gràfica del programa.
% Les càrregues projectades sempre es representen per sobre de la barra.
%
% Entrada:
% - n: número de barra.
% - cordinf: coordenada del punt de la barra on comença la càrrega.
% - cordsup: coordenada del punt de la barra on acaba la càrrega .
% - projectada: vector de la càrrega projectada en format [mòdul,component x,component y,component z].
% - s: factor d'escala.
% - fontcarregues: tipus de font per als textos de les càrregues.
% - tamanyfontcarregues: tamany dels textos.
%
% Sortida:
% - hgproj: handle de l'objecte hgtransform que representa la càrrega projectada.

%% FUNCIÓ PRINCIPAL:

function hgproj = dibuix_projectada(n,cordinf,cordsup,projectada,s,fontcarregues,tamanyfontcarregues, qmax)

if projectada(1) && any(projectada(2:4)) % Si hi ha definida correctament la càrrega projectada.

    projectada=projectada*projectada(1)/abs(projectada(1));
    vbarra=cordsup-cordinf; % Vector direcció de la barra.
    vcarrega=projectada(2:4); % Vector direcció de la càrrega.
    prodesc=dot(vbarra/norm(vbarra),vcarrega/norm(vcarrega)); % Producte escalar dels vectors direcció
de la barra i de la càrrega.

    if any(vbarra) && ~(abs(prodesc)==1) % Si les direccions de la barra i la càrrega no son
paral·leles...
        if prodesc>0 % Si el dibuix de la càrrega es crea amb la barra intercanvia els extrems per
evitar-ho.
            cordsup=cordinf;
            vbarra=-vbarra;
        end
        vperp=cross(vbarra,vcarrega); % Calcula un vector perpendicular a la barra i a la direcció de
la càrrega.
        vperp=cross(vperp,vcarrega); % Calcula un vector perpendicular a l'anterior i a la direcció de
la càrrega, forma un triangle al mateix pla amb la càrrega i la barra, és el vector projecció de la
càrrega sobre la barra.
        vperp=vperp/norm(vperp);
        lcarrega=dot(vperp,vbarra); % Longitud de la càrrega.
        cordinf=cordsup-vperp*lcarrega; % Coordenada inferior de l'extrem de la projecció de la
càrrega.

        hgproj=hgtransform('HitTestArea','on','HitTest','off','Clipping','on','UserData',n); % Crea
l'objecte hgtransform.
        sep=(-0.1-0.5*projectada(1)/qmax)*(s*projectada(2:4)/norm(projectada(2:4))); % Gruix de la
càrrega en forma de vector.
        postxt=dibuixa_linies(hgproj,sep,cordinf,cordsup); % Dibuixa les línies que uneixen les
fletxes.
        dibuixa_fletxes(lcarrega,cordinf,cordsup,vcarrega,hgproj,s,norm(sep)/s) % Dibuixa les
fletxes.
        dibuixa_texte(postxt,projectada,fontcarregues,tamanyfontcarregues,hgproj) % Col·locació del
texte.
    else
        hgproj=[]; % Si la barra i la direcció de la càrrega són paral·leles no hi ha càrrega i no es
dibuixa.
    end
else
    hgproj=[]; % Si la càrrega no està definida correctament no es dibuixa.
end
end

%% DIBUIX DE LÍNIES, FLETXES I TEXTE:

function lininf=dibuixa_linies(handle,separacio,cordinf,cordsup) % Dibuixa dos línies paral·leles
separades per un vector.
    lininf=cordinf+separacio; % Coordenades de l'extrem inferior de la línia paral·lela.
    linsup=cordsup+separacio; % Coordenades de l'extrem superior de la línia paral·lela.
    line([lininf(1),linsup(1)],[lininf(2),linsup(2)],[lininf(3),linsup(3)], 'color','k','LineWidth',
1,'Parent',handle); % Línia paral·lela a la projecció de la càrrega.
    line([cordinf(1),cordsup(1)],[cordinf(2),cordsup(2)],[cordinf(3),cordsup
(3)], 'color','k','LineWidth',1,'Parent',handle); % Línia de la projecció de la càrrega.
end
function dibuixa_fletxes(long,cordinf,cordsup,vector,handle,s,separacio) % Dibuixa les fletxes

```



```

entre dos punts.
    numfletxes=abs(round(long/(0.2*s))); % Calcula el número de fletxes, a més de la inicial, que hi
cabem segons l'escala.
    if numfletxes < 1
        numfletxes=1; % Mínim 1 fletxa més la inicial.
    end
    interval=(cordsup-cordinf)/numfletxes; % Separació entre fletxes.
    m=calcul_matriu(vector); % Matriu orientació de les fletxetes.
    for i=0:numfletxes % dibuixa les fletxetes.
        dibuixa_fletxa(handle,s,separacio,(cordinf+interval*i)',m)
    end
end
function dibuixa_fletxa(handle,s,separacio,pos,m)
    XYZl=m*[-separacio,-0.1;0,0;0,0]*s+[pos,pos]; % Línia de la fletxa rotada, escalada i col·locada.
    XYZp=m*[-0.1,0,-0.1;0.018,0,-0.018;0,0,0]*s+[pos,pos,pos]; % Punta de la fletxa rotada, escalada i
col·locada.
    line(XYZl(1,:),XYZl(2,:),XYZl(3,:), 'color','k','LineWidth',1,'Parent',handle,'HitTest','off');
    patch(XYZp(1,:),XYZp(2,:),XYZp(3,:), 'k','Parent',handle,'HitTest','off');
end
function dibuixa_texte(postxt,carrega,fontcarregues,tamanyfontcarregues,handle)
    if carrega(2)>0
        halin='right';
    else
        halin='left';
    end
    if carrega(3)>0;
        valin='top';
    else
        valin='bottom';
    end
    text(postxt(1),postxt(2),postxt(3),strcat('Qp=',num2str(carrega(1))), 'FontName',
fontcarregues,'FontUnits','pixels','FontSize',tamanyfontcarregues,'VerticalAlignment',
valin,'HorizontalAlignment',halin,'Parent',handle);
end

%% CÀLCUL DE LA MATRIU DE CANVI DE BASE:

function matriu_rotacio=calcul_matriu(vector)
    vector=vector/norm(vector); % Normaliza el vector.
    if abs(vector(3))==1 % Si el vector és paral·lel a l'eix z...
        vector3=[vector(3),0,0];
        vector2=[0,1,0];
    else
        vector2=cross([0,0,1],vector); % Calcula dos vectors perpendiculars al primer i entre ells per
crear la matriu de canvi de base.
        vector2=vector2/norm(vector2);
        vector3=cross(vector,vector2);
        vector3=vector3/norm(vector3);
    end
    matriu_rotacio=[vector',vector2',vector3'];
end

```

```

% Funció dibuix_repartida per a Càlcul Matricial d'Estructures 2 (CME2).
%
% Per Sergio Guerrero Miralles, darrera versió 13 d'octubre de 2.012.
%
% Crea objectes gràfics del tipus hgtransform amb la representació de les
% càrregues repartides sobre la barra.
% Retorna el handle, per a col·locar i escalar a la gràfica del programa.
% Les càrregues repartides sempre penjen de la barra.
%
% Entrada:
% - n: número de barra.
% - cordinf: coordenada del punt de la barra on comença la càrrega.
% - cordsup: coordenada del punt de la barra on acaba la càrrega .
% - repartida: vector de la càrrega repartida en format [mòdul,component x,component y,component z].
% - s: factor d'escala.
% - fontcarregues: tipus de font per als textos de les càrregues.
% - tamanyfontcarregues: tamany dels textos.
%
% Sortida:
% - hgrep: handle de l'objecte hgtransform que representa la càrrega repartida.

%% FUNCIO PRINCIPAL:

function hgrep = dibuix_repartida(n,cordinf,cordsup,repertida,s,fontcarregues,tamanyfontcarregues,qmax)

if repertida(1) && any(repertida(2:4)) % Si hi ha definida correctament la càrrega repartida.
    repertida=repertida*repertida(1)/abs(repertida(1));
    hgrep=hgtransform('HitTestArea','on','HitTest','off','Clipping','on','UserData',n); % Crea
l'objecte hgtransform.
    sep=(0.1+0.5*repertida(1)/qmax)*(s*repertida(2:4)/norm(repertida(2:4))); % Gruix de la càrrega en
forma de vector.
    postxt=dibuixa_linies(hgrep,sep,cordinf,cordsup); % Dibuixa les línies que uneixen les fletxetes.
    dibuixa_fletxetes(norm(cordsup-cordinf),cordinf,cordsup,repertida(2:4),hgrep,s,norm(sep)/s) %
Dibuixa les fletxetes.
    dibuixa_texte(postxt,repertida,fontcarregues,tamanyfontcarregues,hgrep) % Col·locació del text.
else
    hgrep=[]; % Si la càrrega no està definida correctament no es dibuixa.
end
end

%% DIBUIX DE LÍNIES, FLETXES I TEXTE:

function lininf=dibuixa_linies(handle,separacio,cordinf,cordsup) % Dibuixa dos línies paral·leles
separades per un vector.
    lininf=cordinf+separacio; % Coordenades de l'extrem inferior de la línia paral·lela.
    linsup=cordsup+separacio; % Coordenades de l'extrem superior de la línia paral·lela.
    line([lininf(1),linsup(1)],[lininf(2),linsup(2)],[lininf(3),linsup(3)],'color','k','LineWidth',
1,'Parent',handle); % Línia paral·lela a la projecció de la càrrega.
    line([cordinf(1),cordsup(1)],[cordinf(2),cordsup(2)],[cordinf(3),cordsup
(3)],'color','k','LineWidth',1,'Parent',handle); % Línia de la projecció de la càrrega.
end
function dibuixa_fletxetes(long,cordinf,cordsup,vector,handle,s,separacio) % Dibuixa les fletxetes
entre dos punts.
    numfletxes=abs(round(long/(0.2*s))); % Calcula el número de fletxes, a més de la inicial, que hi
cabem segons l'escala.
    if numfletxes < 1
        numfletxes=1; % Mínim 1 fletxa més la inicial.
    end
    interval=(cordsup-cordinf)/numfletxes; % Separació entre fletxes.
    m=calcul_matriu(vector); % Matriu orientació de les fletxetes.
    for i=0:numfletxes % dibuixa les fletxetes.
        dibuixa_fletxa(handle,s,(cordinf+interval*i)',m,separacio)
    end
end
function dibuixa_fletxa(handle,s,pos,m,separacio)
    XYZl=m*[separacio,0;0,0;0,0]*s+[pos,pos]; % Línia de la fletxa rotada, escalada i col·locada.
    XYZp=m*[separacio-0.1,separacio,separacio-0.1;0.018,0,-0.018;0,0,0]*s+[pos,pos,pos]; % Punta de la
fletxa rotada, escalada i col·locada.
    line(XYZl(1,:),XYZl(2,:),XYZl(3,:), 'color','k','LineWidth',1,'Parent',handle,'HitTest','off');
    patch(XYZp(1,:),XYZp(2,:),XYZp(3,:), 'k','Parent',handle,'HitTest','off');
end
function dibuixa_texte(postxt,carrega,fontcarregues,tamanyfontcarregues,handle)
    if carrega(2)>0
        halin='right';
    else
        halin='left';
    end
    if carrega(3)>0;
        valin='bottom';
    else
        valin='top';
    end
end

```

```

text(postxt(1),postxt(2),postxt(3),strcat('Qr=',num2str(carrega(1))), 'FontName',
fontcarregues, 'FontUnits', 'pixels', 'FontSize', tamanyfontcarregues, 'VerticalAlignment',
valin, 'HorizontalAlignment', halin, 'Parent', handle);
end

```

```

%% CÀLCUL DE LA MATRIU DE CANVI DE BASE:

```

```

function matriu_rotacio=calcul_matriu(vector)
vector=vector/norm(vector); % Normaliza el vector.
if abs(vector(3))=1 % Si el vector és paral·lel a l'eix z...
vector3=[vector(3),0,0];
vector2=[0,1,0];
else
vector2=cross([0,0,1],vector); % Calcula dos vectors perpendiculars al primer i entre ells per
crear la matriu de canvi de base.
vector2=vector2/norm(vector2);
vector3=cross(vector,vector2);
vector3=vector3/norm(vector3);
end
matriu_rotacio=[vector',vector2',vector3'];
end

```

```

classdef nus
    %UNTITLED Summary of this class goes here
    % Detailed explanation goes here

    properties
        num=0; % Número de nus.
        posicio=[0,0,0]; % Posició a l'espai.
        contorn=0; % Tipus de contorn.
        rodet=[0,1,0]; % Vector normal al plànol de moviment del rodet, si n'hi ha.
        desp=[0,0,0,0,0,0]; % desplaçaments forçats si n'hi han.
        rig=[0,0,0,0,0,0]; % rigideses de contorn elàstic si n'hi ha.
        gir=0; % rotació del dibuix del nus.
        hg=[]; % handle de l'objecte hgtransform que conté el dibuix del nus.
        hgc=[]; % handle de l'objecte hgtransform que conté el dibuix de la càrrega aplicada al nus.
        puntual=[0,0,0]; % Càrrega puntual aplicada al nus.
        moment=[0,0,0]; % Moments aplicats al nus.
    end

    methods

    end

end
end

```

```

classdef barra
    %UNTITLED Summary of this class goes here
    % Detailed explanation goes here

    properties
        num=0; % Número de barra.
        nusinf=0; % Número de nus inferior.
        nussup=0; % Número de nus superior.
        artinf=0; % Nus inferior articulat: 0=rígid, 1=articulat.
        artsup=0; % Nus superior articulat: 0=rígid, 1=articulat.
        color='k'; % Color de la barra.
        E=1; % Mòdul de Young.
        A=1; % Àrea de la secció.
        Iz=1; % Inèrcia en eix z.
        Iy=1; % Inèrcia en eix y.
        vectorIz=[0,0,1]; % Vector direcció de l'eix z de la secció.
        GI=1; % Inèrcia a torsió.
        hg % handle a l'objecte gràfic de la barra.
        hgpuntual % handle a l'objecte gràfic de la càrrega puntual a la barra.
        hgmoment % handle a l'objecte gràfic del moment a la barra.
        hgrepartida % handle a l'objecte gràfic de la càrrega repartida.
        hgprojectada % handle a l'objecte gràfic de la càrrega repartida projectada.
        puntual=[0,0,-1,0,0]; % Càrrega puntual: mòdul, vector direcció x,y,z i distància al nus
    inferior.
        moment=[0,0,0,1,0]; % Moment aplicat a la barra: mòdul, vector direcció x,y,z i
    distància al nus inferior.
        repartida=[0,0,-1,0,0,0]; % Càrrega repartida: mòdul, vector direcció x,y,z i distància al
    nus inferior i superior.
        projectada=[0,0,-1,0,0,0]; % Càrrega repartida peojectada: mòdul, vector direcció x,y,z i
    distància al nus inferior i superior.
    end

    methods

    end
end
end

```

```

classdef diagrama
    %UNTITLED Summary of this class goes here
    % Detailed explanation goes here

    properties
        num=0; % Número de la barra a la qual pertanyen els diagrames.

        hgN=[]; % Handle de l'objecte hgtransform amb el diagrama d'axils.
        hgTy=[]; % Handle de l'objecte hgtransform amb el diagrama de tallants en y.
        hgTz=[]; % Handle de l'objecte hgtransform amb el diagrama de tallants en z.
        hgMx=[]; % Handle de l'objecte hgtransform amb el diagrama de moments en x.
        hgMy=[]; % Handle de l'objecte hgtransform amb el diagrama de momentss en y.
        hgMz=[]; % Handle de l'objecte hgtransform amb el diagrama de moments en z.
        hgdeformada=[]; % Handle de l'objecte hgtransform amb la deformada.

        Mmax=0; % Moment màxim absolut.
        Tmax=0; % Axil o tallant màxim absolut.
        Dmax=0; % Desplaçament màxim absolut.

        Oini=[0,0]; % Girs y i z a l'extrem inicial de la barra.
        Ofin=[0,0]; % Gir y i z a l'extrem final de la barra.
    end

    methods
    end
end
end

```