

Treball de Fi de Grau

## **Grau en Enginyeria Química**

# **Control del perfil de curat durant el processat de materials termostables duals basats en mescles tiol-epoxi**

**ANNEX**

**Autor:** Maria Romero Casademunt  
**Director:** Xavier Fernàndez Francos; Xavier Ramis Juan  
**Convocatòria:** Juliol 2017



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona





# Índex

1. Esquema de funcionament.....	3
2. Codi .....	5
2.1. Cos del programa .....	5
2.2. Conductivitat tèrmica .....	13
2.3. Calor específica.....	13
2.4. Densitat .....	14
2.5. Temps d'inducció.....	15
2.6. Model Tg-x.....	15
2.7. Model cinètic.....	16
2.1. Programes tèrmics .....	17



# 1. Esquema de funcionament

En general, el programa rep una sèrie de paràmetres d'entrada, que extreu d'un document Excel estructurat. D'aquesta manera, editant el document i guardant-lo s'aconsegueix simular diferents situacions, sense necessitat d'editar el codi del programa en sí. El mateix Excel també conté les propietats necessàries pels materials, el programa tèrmic, els paràmetres dels models de propietats termofísiques i models cinètics, etc.

Amb les variables necessàries, es fan els primers càlculs, que permetin establir els condicionants de la integració. Amb aquests, s'inicialitzen els valors i índexs necessaris, i es procedeix a la primera integració.

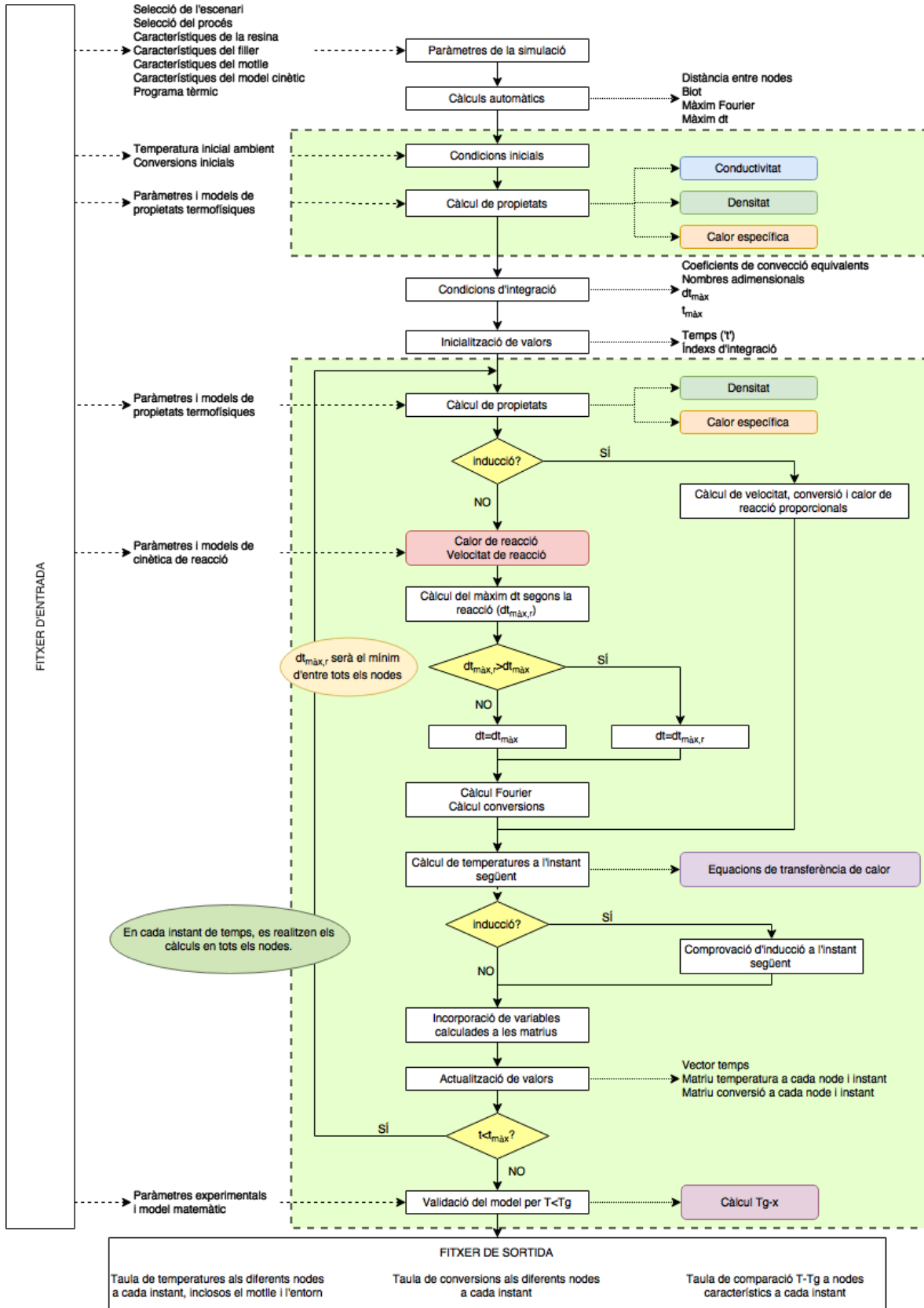
A cada instant de temps, es calculen les propietats termofísiques en aquell instant, es comprova en cadascun dels nodes del sistema si es troba o no en inducció i així es realitzen els càlculs de calor i velocitat de reacció pertinents. Per tots els nodes que ja no es trobin en període d'inducció, es calcula el màxim diferencial de temps i amb el conjunt d'informació s'estableix el diferencial de temps adequat que compleixi totes les restriccions.

A continuació es calcula el Fourier i la conversió corresponent a cada node, i amb tot plegat es calculen les temperatures a l'instant següent, utilitzant les equacions de l'escenari escollit. Cada 10 segons (es pot modificar, en ser un paràmetre d'entrada) es guarda tota la informació de temperatures i conversions en una matriu. Al final de cada integració s'actualitzen els valors, i es procedeix a la integració a l'instant següent de la mateixa manera.

El bucle finalitza quan s'assoleix el temps final indicat al programa tèrmic. Finalment, per validar el model es realitza el càlcul de la temperatura de transició vítria en els nodes més representatius de la mostra a cada instant, i es compara amb la corresponent al node al mateix instant. D'aquesta manera es comprova que en qualsevol moment, la temperatura de la mostra és superior a la  $T_g$ , és a dir, que el material no està vitrificat.

Finalment es genera un fitxer de sortida en forma de full de càlcul Excel amb diferents pestanyes, corresponent a les temperatures, les conversions i la validació del model indicada. Per altra banda, el programa també ofereix la possibilitat de graficar directament els resultats obtinguts.

A continuació s'esquematitza el programa mitjançant un diagrama de blocs. Les zones ombrejades en verd representen els càlculs que es realitzen per cadascun dels nodes de la mostra, en cas que aquesta estigui discretitzada. Els quadres de color indiquen que el càlcul corresponent es realitza a través d'una funció cridada des del programa principal.



## 2. Codi

A continuació es mostren les línies de codi del programa, així com el conjunt de funcions secundàries que aquest utilitza per fer diferents càlculs.

### 2.1. Cos del programa

```

%
%INPUTS
%
%CONDICIONS DE PARTIDA
CondPartida=xlsread('PARAM','CondPartida');
tipus=CondPartida(1); %Sistema: senzill o dual
programa=CondPartida(2); %Escenari
Tini=CondPartida(3); %Temperatura inicial
L_m=CondPartida(4); %Gruix de mostra
L_met=CondPartida(5); %Gruix de metall(*)
n=CondPartida(6); %Nombre de nodes(*)
fi_f=CondPartida(7); %Fracció de filler (en volum)
metode_k=CondPartida(8); %Model de càlcul de conductivitat
%(*)Només s'utilitzen en els escenaris que ho requereixin

%PROPIETATS DELS MATERIALS
Propietats=xlsread('PARAM','Propietats');
k_f=Propietats(1); %Conductivitats (W/mK)
k_met=Propietats(2);
ro_f=Propietats(4); %Densitats (kg/m3)
ro_met=Propietats(5);
cp_f=Propietats(7); %Calor específica (J/kgK)
cp_met=Propietats(8);
h=Propietats(10); %Convecció a l'entorn (W/m2K)

%PROGRAMA TÈRMIC
ProgTermic=xlsread('PARAM','ProgTermic');
T_ent=ProgTermic(3,1); %Temperatura inicial (°C)
beta_programa=ProgTermic(1,:); %Velocitat d'escalfament (°C/s)
temps_programa=ProgTermic(2,:); %Temps (s)

%CONDICIONANTS INTEGRACIÓ
Condicionants=xlsread('PARAM','Condicionants');
dx_max=Condicionants(1); %Màxim dx
fs=Condicionants(2); %Factor de seguretat pel Fourier
dt_g=Condicionants(3); %Interval de temps de guardar valors

%PARÀMETRES CINÈTICS (importació dels paràmetres pels respectius models)
if tipus==1
    PCin=xlsread('PARAM','PCin_senzill');
    Ptind=xlsread('PARAM','Ptind_senzill');
elseif tipus==2
    PCin=xlsread('PARAM','PCin_dual');
    Ptind=xlsread('PARAM','Ptind_dual');
end

%MODELS PROPIETATS
if tipus==1
    CpRes=xlsread('PARAM','CpRes_senzill');
    RoRes=xlsread('PARAM','RoRes_senzill');
    DilRes=xlsread('PARAM','DilRes_senzill');

```

```

    k_r=xlsread('PARAM','kRes_senzill');
elseif tipus==2
    CpRes=xlsread('PARAM','CpRes_dual');
    RoRes=xlsread('PARAM','RoRes_dual');
    DilRes=xlsread('PARAM','DilRes_dual');
    k_r=xlsread('PARAM','kRes_dual');
end

%MODEL Tg-x
if tipus==1
    Tg_x=xlsread('PARAM','Tg_x_senzill');
elseif tipus==2
    Tg_x=xlsread('PARAM','Tg_x_dual');
end

%
%CÀLCULS
%
%GRUIX D'ESTUDI
if programa==1 || programa==2 || programa==3
    L_m=0.5*L_m;                %Meitat del gruix (si hi ha simetria)
end

%PAS ENTRE NODES
if programa==1
    dz=L_m/(n-0.5);
elseif programa==2
    dz=L_m;
    n=1;
elseif programa==3
    dz=L_m/(n-0.5);
elseif programa==4
    dz=L_m/n;
end

%VECTOR POSICIÓ NODES
if programa==1 || programa==3
    z_m_vec=[0:dz:L_m];
    if programa==3
        z_m_vec(n+1)=L_m+L_met/2;
    end
elseif programa==2
    z_m_vec=' ';
elseif programa==4
    z_m_vec=[dz/2:dz:(L_m-dz/2)];
    z_m_vec(n+1)=L_m+L_met/2;
end

%FRACCIÓ EN VOLUM DE RESINA
fi_r=1-fi_f;
%
%CONDICIONS INICIALS
%
T_m(1:n)=Tini;                %Temperatures inicials del sistema
if programa==3 || programa==4
    T_met=Tini;
end
x_r(1:n)=0;                  %Conversió inicial del 1r procés
y_r(1:n)=0;                  %Conversió inicial del 2n procés
induccio(1:n)=1;            %Inici en temps d'inducció

```



```

%PROPIETATS DE LA MESCLA (Conductivitat, Densitat i Calor Específica)
if metode_k==1
    [k_m]=f_k_LewisNielsen(fi_f,k_r,k_f);
elseif metode_k==2
    [k_m] = f_k_fillerfibra(fi_f,k_r,k_f);
end
for p=1:n
    [ro_r(p),ro_m(p)]=f_ro_m(RoRes,DilRes,fi_f,ro_f,T_m,x_r,y_r,p,tipus);
    [cp_r(p),cp_m(p)]=f_cp_m(CpRes,cp_f,fi_f,ro_r,ro_f,T_m,x_r,y_r,p,tipus);
end

%
%CONDICIONS D'INTEGRACIÓ
%
%CONVECCIÓ EQUIVALENT
h_m_ent=1/(dz/(2*k_m)+1/h);           %h eq. mostra-entorn
if programa~1
    h_met_ent=1/(L_met/(2*k_met)+1/h);   %h eq. metall-entorn
    h_m_met=1/(dz/(2*k_m)+L_met/(2*k_met)); %h eq. mostra-metall
end

%NOMBRES ADIMENSIONALS
%Biot
if programa==1
    Bi=h_m_ent*dz/k_m;
elseif programa==3
    Bi=h_m_met*dz/k_m;
elseif programa==4
    Bil=h_m_ent*dz/k_m;
    Bin=h_m_met*dz/k_m;
    Bi=max(Bil,Bin);
end
%Fourier màxim --> màxim dt
if programa==1 || programa==3 || programa==4
    %Màxim Fourier que compleix la condició
    Fomax_conv=1/(1+Bi);           %Convecció
    Fomax_cond=0.5;               %Conducció
    %Selecció del mínim
    Fomax=min(Fomax_conv,Fomax_cond);
    %Màxim dt
    dtmax=min(Fomax*ro_m.*cp_m.*dz^2/k_m);
    %Si hi ha metall
    if programa==3 || programa==4
        dtmax_met=ro_met*cp_met*L_met/(h_met_ent+h_m_met);
        dtmax=min(dtmax,dtmax_met);
    end
elseif programa==2
    %Màxim dt
    dtmax=(ro_m*cp_m*L_m+ro_met*cp_met*L_met)/(2*h);
end

%INICIALITZACIÓ DE VALORS
i_pt=1;           %Índex del programa tèrmic
i_real=1;        %Índex posició vector temps
t_real=0;        %Temps real (s)
t_ind(1:n)=0;    %Temps d'inducció (s)
i_graf=1;        %Índex posició vector temps gràfic
i_comp_graf=0;   %Índex comparatiu pel gràfic
%Paràmetres per la integració
x_rant(1:n)=0;   %Conversió per la inducció
Int(1:n)=0;      %Integral per la inducció

```

```

%TEMPS MÀXIM (del programa tèrmic) (s)
t_max=temps_programa(length(temps_programa));

%
%INTEGRACIÓ
%

%BUCLE
while t_real<=t_max
|   %pas de temps
|   dt=dtmax/fs;

|   %PROPIETATS DE LA MESCLA (Densitat i Calor Específica)
|   for p=1:n
|       [ro_r(p),ro_m(p)]=f_ro_m(RoRes,DilRes,fi_f,ro_f,T_m,x_r,y_r,p,tipus);
|       [cp_r(p),cp_m(p)]=f_cp_m(CpRes,cp_f,fi_f,ro_r,ro_f,T_m,x_r,y_r,p,tipus);
|   end

|   %VELOCITAT, CALOR I dt SEGONS LA REACCIÓ
|   for p=1:n
|       %Si encara està en període d'inducció
|       if induccio(p)
|           dxdt(p)=(x_r(p)-x_rant(p))/dt;
|           dydt(p)=0;
|           q_r(p)=dxdt(p)*PCin(1,5);
|           %Aproximació de la conversió en període d'inducció
|           x_rant(p)=x_r(p);
|           x_r_nova(p)=Int(p)/exp(Ptind(1))*0.01;
|           y_r_nova(p)=0;

|           %si ha sobrepassat el període d'inducció
|           else
|               [q_r(p),dxdt(p),dydt(p)]=f_model_cin(PCin,x_r,y_r,T_m,p,tipus);
|               dtmax_r=dx_max/max(dxdt(p),dydt(p));
|               %Corregir dt tenint en compte la reacció (si cal)
|               if dtmax_r<dt
|                   dt=dtmax_r;
|               end
|               x_r_nova(p)=x_r(p)+dt*dxdt(p);
|               y_r_nova(p)=y_r(p)+dt*dydt(p);
|           end
|       end

|       %FOURIER
|       Fo=dt.*k_m./(dz^2.*ro_m.*cp_m);

|       %TEMPERATURES A L'INSTANT SEGÜENT
|       %Entorn (programa tèrmic)
|       if t_real-temps_programa(i_pt)>=0 && i_pt<length(temps_programa)
|           i_pt=i_pt+1;
|       end
|       T_ent=T_ent+beta_programa(i_pt)*dt;
|       %Mostra i metall (si n'hi ha)
|       if programa==1
|           %Simetria mostra convecció ; n nodes
|           [T_m_nova] = f_programa1(n,Bi,Fo,T_ent,T_m,dt,q_r,fi_r,ro_r,ro_m,cp_m);

|       elseif programa==2
|           %Simetria conjunt(mostra+metall) convecció ; 1 node ; "capa fina"
|           [T_m_nova] = f_programa2(T_ent,T_m,L_m,L_met,dt,q_r,fi_r,ro_r,ro_m,
|               ro_met,cp_m,cp_met,h);

```

```

| elseif programa==3
|     %Simetria mostra metall convecció ; n nodes ; "motlle tancat"
|     [T_m_nova,T_met_nova] = f_programa3 (n,Bi,Fo,T_ent,T_m,T_met,L_met,dt,
|     q_r,h_m_met,h_met_ent,fi_r,ro_r,ro_m,ro_met,cp_m,cp_met);
|
|     elseif programa==4
|     %Convecció metall mostra convecció ; n nodes; "motlle obert"
|     [T_m_nova,T_met_nova] = f_programa4(n,Bi1,Bin,Fo,T_ent,T_m,T_met,L_met,
|     dt,q_r,h_m_met,h_met_ent,fi_r,ro_r,ro_m,ro_met,cp_m,cp_met);
|
| end
|
| %Comprova (node a node) si encara hi haurà inducció l'instant següent
| %Només en cas que en l'instant actual n'hi hagi
| for p=1:n
|     if induccio(p)
|         [induccio(p),Int(p)]=f_tempsinduccio(Ptind,T_m,dt,Int,p);
|         t_ind(p)=t_real;
|     end
| end
|
| %VECTORS I MATRIUS PELS GRÀFICS
| if (i_real-i_comp_graf)*dt>dt_g;
|     t_vec(i_graf)=t_real/60;           %Vector t
|     x_mat(i_graf,:)=x_r';             %Matriu x ; f(temps, node)
|     y_mat(i_graf,:)=y_r';             %Matriu y ; f(temps, node)
|     T_m_mat(i_graf,:)=T_m';           %Matriu T ; f(temps, node)
|     if programa==3 || programa==4
|         T_met_vec(i_graf)=T_met;      %Vector T_met ; f(temps)
|     end
|     T_ent_vec(i_graf)=T_ent;           %Vector T_ent ; f(temps)
|     %Actualització de valors pels gràfics
|     i_graf=i_graf+1;
|     i_comp_graf=i_real;
| end
|
| %ACTUALITZACIÓ DE VALORS PER A L'INSTANT SEGÜENT
| T_m=T_m_nova_m;
| if programa==3 || programa==4
|     T_met=T_met_nova;
| end
| x_r=x_r_nova;
| y_r=y_r_nova;
| i_real=i_real+1;
| t_real=t_real+dt;
end

%
%VALIDACIÓ MODEL SEGONS Tg-x
%

%Selecció dels nodes d'estudi
if programa==1 || programa==3
    T_Tg=[T_m_mat(:,1),T_m_mat(:,n)];
    x_Tg=[x_mat(:,1),x_mat(:,n)];
    y_Tg=[y_mat(:,1),y_mat(:,n)];
elseif programa==2
    T_Tg=T_m_mat;
    x_Tg=x_mat;
    y_Tg=y_mat;
elseif programa==4
    centr=round((n+1)/2);
    T_Tg=[T_m_mat(:,1),T_m_mat(:,centr),T_m_mat(:,n)];
    x_Tg=[x_mat(:,1),x_mat(:,centr),x_mat(:,n)];

```

```

    y_Tg=[y_mat(:,1),y_mat(:,centr),y_mat(:,n)];
end

%Dimensionalització i inicialització de valors
[fil,col]=size(x_Tg);
c=1;
cmat=1;

%Comparació de valors T i Tg a cada instant en tots els punts d'estudi
while c<=col
    f=1;
    while f<=fil
        Tg=f_Tg_x(x_Tg(f,c),y_Tg(f,c),Tg_x,tipus);
        if T_Tg(f,c)<Tg
            simb={'<'};
        else
            simb={'>'};
        end
        T_Tg_mat(f,cmat:cmat+2)=[T_Tg(f,c),simb,Tg];
        f=f+1;
    end
    c=c+1;
    cmat=cmat+3;
end

%
%FITXER DE SORTIDA
%

fitxer='Fitxer.xls';
%TEMPERATURA
%Títol de les variables que es representen
if programa~=2
    xlswrite(fitxer,{'posició(m)->'},'Temperatura','A1');
    xlswrite(fitxer,{'t(min)/node->'},'Temperatura','A2');
    xlswrite(fitxer,z_m_vec,'Temperatura','B1');
elseif programa==2
    xlswrite(fitxer,{'t(min) '},'Temperatura','A2');
end
%Temps
xlswrite(fitxer,t_vec,'Temperatura','A3');
%Nodes
if programa~=2
    znodesmostra_vec=[1:n];
    for node=1:n
        znodes_vec(node)={znodesmostra_vec(node)};
    end
elseif programa==2
    znodes_vec(1)={'MOSTRA'};
end
if programa==1 || programa ==2
    znodes_vec(n+1)={'ENTORN'};
elseif programa==3 || programa==4
    znodes_vec(n+1)={'METALL'};
    znodes_vec(n+2)={'ENTORN'};
end
xlswrite(fitxer,znodes_vec,'Temperatura','B2');
%Temperatures
if programa==1 || programa==2
    xlswrite(fitxer,[T_m_mat,T_ent_vec'],'Temperatura','B3');
elseif programa==3 || programa==4
    xlswrite(fitxer,[T_m_mat,T_met_vec',T_ent_vec'],'Temperatura','B3');
end

```

```

%CONVERSIÓ
%Títol de les variables que es representen
%Procés senzill o Primer procés dual
if programa~=2
    xlswrite(fitxer,{'posició(m)->'},'Conversió1','A1');
    xlswrite(fitxer,z_m_vec(1:n),'Conversió1','B1');
    xlswrite(fitxer,{'t(min)/node->'},'Conversió1','A2');
    xlswrite(fitxer,[1:n],'Conversió1','B2');
elseif programa==2
    xlswrite(fitxer,{'t(min)'},'Conversió1','A2');
end
%Temps
xlswrite(fitxer,t_vec,'Conversió1','A3');
xlswrite(fitxer,x_mat,'Conversió1','B3');
%Segon procés dual (si cal)
if tipus==2
    if programa~=2
        %Títol de les variables que es representen
        xlswrite(fitxer,{'posició(m)->'},'Conversió2','A1');
        xlswrite(fitxer,z_m_vec(1:n),'Conversió2','B1');
        xlswrite(fitxer,{'t(min)/node->'},'Conversió2','A2');
        xlswrite(fitxer,[1:n],'Conversió2','B2');
    elseif programa==2
        xlswrite(fitxer,{'t(min)'},'Conversió2','A2');
    end
    %Temps
    xlswrite(fitxer,t_vec,'Conversió2','A3');
    xlswrite(fitxer,y_mat,'Conversió2','B3');
end
%Conversió mitjana (si cal)
if tipus==2
    if programa~=2
        %Títol de les variables que es representen
        xlswrite(fitxer,{'posició(m)->'},'Conversió mitjana','A1');
        xlswrite(fitxer,z_m_vec(1:n),'Conversió mitjana','B1');
        xlswrite(fitxer,{'t(min)/node->'},'Conversió mitjana','A2');
        xlswrite(fitxer,[1:n],'Conversió mitjana','B2');
    elseif programa==2
        xlswrite(fitxer,{'t(min)'},'Conversió mitjana','A2');
    end
    %Temps
    xlswrite(fitxer,t_vec,'Conversió mitjana','A3');
    x_mitjana_mat=0.5.*x_mat+0.5.*y_mat;
    xlswrite(fitxer,x_mitjana_mat,'Conversió mitjana','B3');
end

%COMPARACIÓ T-Tg EN ALGUNS NODES
%Títol de les variables que es representen
if programa~=2
    xlswrite(fitxer,{'t(min)/node->'},'T-Tg','A2');
elseif programa==2
    xlswrite(fitxer,{'t(min)'},'T-Tg','A2');
end
%Temps
xlswrite(fitxer,t_vec,'T-Tg','A3');
%Nodes
if programa==1
    znodes_vec={'->','CENTRAL','<-','->','EXTERIOR','<-'};
elseif programa==2
    znodes_vec={'->','MOSTRA','<-'};
elseif programa==3
    znodes_vec={'->','CENTRAL','<-','->','METALL','<-'};
elseif programa==4
    znodes_vec={'->','ENTORN','<-','->','CENTRAL','<-','->','METALL','<-'};
end

```

```
end
xlswrite(fitxer,znodes_vec,'T-Tg','B2');
%Matriu comparació T-Tg
xlswrite(fitxer,T_Tg_mat,'T-Tg','B3');

%
%GRÀFICS
%

%Temperatures en funció del temps
figure()
hold on
for i=1:n
    plot(t_vec,T_m_mat(:,i),'Color',[rand,rand,rand]);
    plot(t_vec,T_ent_vec,'Color',[0,0,0]);
    if programa==3 || programa==4
        plot(t_vec,T_met_vec,'--')
    end
end
xlabel('t (min)');
ylabel('T (°C)');
hold off

%Conversions en funció del temps
figure()
hold on
for i=1:n
    plot(t_vec,x_mat(:,i),'Color',[rand,rand,rand]);
    if tipus==2
        plot(t_vec,y_mat(:,i),'Color',[rand,rand,rand]);
    end
end
xlabel('t (min)');
ylabel('conversió');
hold off

%Conversions mitjanes (sistema dual)
if tipus==2
    figure()
    hold on
    for i=1:n
        plot(t_vec,x_mitjana_mat(:,i),'Color',[rand,rand,rand]);
    end
    xlabel('t (min)');
    ylabel('conversió mitjana');
    hold off
end
```

## 2.2. Conductivitat tèrmica

```
function [k_m] = f_k_LewisNielsen(fi_f,k_r,k_f)
%Paràmetres
fi_f_max=0.637;
A=1.5;

%Aplicació del model
B=(k_f/k_r-1)/(k_f/k_r+A);
psi=1+(1-fi_f_max)/fi_f_max^2*fi_f;
k_m=k_r*(1+A*B*fi_f)/(1-B*psi*fi_f);
end
```

```
function [k_m] = f_k_fillerfibra(fi_f,k_r,k_f)
%Aplicació del model
k_m=k_r*((1+fi_f)*k_f+(1-fi_f)*k_r)/((1-fi_f)*k_f+(1+fi_f)*k_r);
end
```

## 2.3. Calor específica

```
function [cp_r,cp_m] = f_cp_m(CpRes,cp_f,fi_f,ro_r,ro_f,T,x_r,y_r,p,tipus)
%Paràmetres
a=CpRes(:,1);
b=CpRes(:,2);
c=CpRes(:,3);
[fil,col]=size(CpRes);
if col>3
    Ttrans=CpRes(:,4);
    for f=1:fil
        if isnan(Ttrans(f))~=0 && T(p)>Ttrans(f)
            a(fil)=CpRes(f,5);
            b(fil)=CpRes(f,6);
            c(fil)=CpRes(f,7);
        end
    end
end
end

%Aplicació del model per la resina
cp0_r=a(1)*T(p)^2+b(1)*T(p)+c(1);
cpinf_r=a(2)*T(p)^2+b(2)*T(p)+c(2);
if tipus==2
    cpint_r=a(3)*T(p)^2+b(3)*T(p)+c(3);
end

%Càlcul segons el grau de reacció
%SISTEMA SENZILL
if tipus==1
    cp_r=cp0_r*(1-x_r(p))+cpinf_r*x_r(p);
%SISTEMA DUAL
elseif tipus==2
    cp_r=cp0_r*(1-x_r(p))+cpint_r*x_r(p)*(1-y_r(p))+cpinf_r*y_r(p);
end

%Fracció en massa de resina
w_r=((1-fi_f)*ro_r(p))/((1-fi_f)*ro_r(p)+fi_f*ro_f);

%Cp de la mostra (resina+filler)
cp_m=w_r*cp_r+(1-w_r)*cp_f;
end
```

## 2.4. Densitat

```

function [ro_r,ro_m] = f_ro_m(RoRes,DilRes,fi_f,ro_f,T,x_r,y_r,p,tipus)
%Paràmetres
a=DilRes(:,1);
b=DilRes(:,2);
c=DilRes(:,3);
[fil,col]=size(DilRes);
if col>3
    Ttrans=DilRes(:,4);
    for f=1:fil
        if isnan(Ttrans(f))~=0 && T(p)>Ttrans(f)
            a(f)=DilRes(f,5);
            b(f)=DilRes(f,6);
            c(f)=DilRes(f,7);
        end
    end
end
end

%Aplicació del model
dilatacio0_r=a(1)*T(p)^2+b(1)*T(p)+c(1);
ro0_r_0=RoRes(1);
ro0_r=ro0_r_0/dilatacio0_r;

dilatacioinf_r=a(2)*T(p)^2+b(2)*T(p)+c(2);
roinf_r_0=RoRes(2);
roinf_r=roinf_r_0/dilatacioinf_r;

if tipus==2
    dilatacioint_r=a(3)*T(p)^2+b(3)*T(p)+c(3);
    roint_r_0=RoRes(3);
    roint_r=roint_r_0/dilatacioint_r;
end

%Càlcul segons el grau de reacció
%SISTEMA SENZILL
if tipus==1
    ro_r=ro0_r*(1-x_r(p))+roinf_r*x_r(p);
%SISTEMA DUAL
elseif tipus==2
    ro_r=ro0_r*(1-x_r(p))+roint_r*x_r(p)*(1-y_r(p))+roinf_r*y_r(p);
end

%Densitat de la mostra (resina+filler)
ro_m=(1-fi_f)*ro_r+fi_f*ro_f;
end

```



## 2.5. Temps d'inducció

```
function [ind,Integral] = f_tempsinduccio(Ptind,T,dt,Int,p)
%Paràmetres
    lngA=Ptind(1);      %min
    E=Ptind(2);        %J/mol
    gA=exp(lngA);      %min
    R=8.314;           %J/(mol·K)

%INTEGRAL --> TIND?
Integral=Int(p)+exp(-E/(R*(T(p)+273.15)))*dt/60;      %min
if Integral>=gA
    ind=0;
else
    ind=1;
end
end
```

## 2.6. Model Tg-x

```
function Tg = f_Tg_x(x,y,Tg_x,tipus)
%Definició de paràmetres
if tipus==1
    Tg0=Tg_x(1)+273.15;
    Tginf=Tg_x(2)+273.15;
    lambda=Tg_x(3);
elseif tipus==2
    if y==0
        Tg0=Tg_x(1,1)+273.15;
        Tginf=Tg_x(2,1)+273.15;
        lambda=Tg_x(3,1);
    else
        Tg0=Tg_x(1,2)+273.15;
        Tginf=Tg_x(2,2)+273.15;
        lambda=Tg_x(3,2);
        x=y;
    end
end
end

%Model
Tg=exp(((1-x)*log(Tg0)+lambda*x*log(Tginf))/((1-x)+lambda*x))-273.15; %°C
end
```

## 2.7. Model cinètic

```

function [q_r,dxdt,dydt] = f_model_cin(PCin,x_r,y_r,T,p,tipus)
R=8.314; %J/(mol·K)

%Unitats:      E      lnA      deltah_r      dxdt      q_r
%              J/mol   s-1      J/kg        s-1        W/kg

%
%-----
%SISTEMA SENZILL
%
%-----
if tipus==1
    %Paràmetres
    E1=PCin(1,1);
    E2=PCin(2,1);
    lnA1=PCin(1,2);
    lnA2=PCin(2,2);
    pm=PCin(1,3);
    pn=PCin(1,4);
    dydt=0;
    deltah_r=PCin(1,5);
    %Constants de reacció
    k1=exp(lnA1-E1/(R*(T(p)+273.15)));
    k11=exp(lnA2-E2/(R*(T(p)+273.15)));
    %Velocitat de reacció
    dxdt=(k1+k11*x_r(p)^pm)*(1-x_r(p))^pn;
    %Calor de reacció
    q_r=dxdt*deltah_r;
end

%
%-----
%SISTEMA DUAL
%
%-----
if tipus==2

    %1r PROCÉS
    deltah_r1=PCin(1,5);
    if x_r(p)<1
        %Paràmetres
        E1=PCin(1,1);
        E11=PCin(2,1);
        E2=PCin(3,1);
        lnA1=PCin(1,2);
        lnA11=PCin(2,2);
        lnA2=PCin(3,2);
        pm1=PCin(1,3);
        pm2=PCin(3,3);
        pn1=PCin(1,4);
        pn2=PCin(3,4);
        %Constants de reacció
        k1=exp(lnA1-E1/(R*(T(p)+273.15)));
        k11=exp(lnA11-E11/(R*(T(p)+273.15)));
        k2=exp(lnA2-E2/(R*(T(p)+273.15)));
        %Velocitat de reacció
        dxdt=(k1+k11*x_r(p)^pm1)*(1-x_r(p))^pn1+k2*x_r(p)^pm2*(1-x_r(p))^pn2;
    %s-1
    else
        dxdt=0;
    end
end

```

```

%Calor de reacció 1a etapa    q_r1=dxdt*deltah_r1;

%2n PROCÉS
deltah_r2=PCin(4,5);
%Si no ha començat el segon procés
if y_r(p)<1
    if x_r(p)<=0.995
        dydt=0;
    else
        %Paràmetres
        E1_2=PCin(4,1);
        E11_2=PCin(5,1);
        E2_2=PCin(6,1);
        E3_2=PCin(7,1);
        lnA1_2=PCin(4,2);
        lnA11_2=PCin(5,2);
        lnA2_2=PCin(6,2);
        lnA3_2=PCin(7,2);
        pm1_2=PCin(4,3);
        pm2_2=PCin(6,3);
        pm3_2=PCin(7,3);
        pn1_2=PCin(4,4);
        pn2_2=PCin(6,4);
        pn3_2=PCin(7,4);
        %Constants de reacció
        k1_2=exp(lnA1_2-E1_2/(R*(T(p)+273.15)));
        k11_2=exp(lnA11_2-E11_2/(R*(T(p)+273.15)));
        k2_2=exp(lnA2_2-E2_2/(R*(T(p)+273.15)));
        k3_2=exp(lnA3_2-E3_2/(R*(T(p)+273.15)));
        %Velocitat de reacció
        dydt=(k1_2+k11_2*y_r(p)^pm1_2)*(1-y_r(p))^pn1_2+k2_2*y_r(p)^pm2_2*(1-
y_r(p))^pn2_2+k3_2*y_r(p)^pm3_2*(1-y_r(p))^pn3_2;
    end
else
    dydt=0;
end
%Calor de reacció 2a etapa
q_r2=dydt*deltah_r2;
%Calor de reacció global
q_r=q_r1+q_r2;
end
end

```

## 2.1. Programes tèrmics

```

function [T_m_nova] = f_programa1(n,Bi,Fo,T_ent,T_m,dt,q_r,fi_r,ro_r,ro_m,cp_m)

%Node adiabàtic
T_m_nova(1)=2*Fo(1)*T_m(2)+(1-
2*Fo(1))*T_m(1)+q_r(1)*dt*ro_r(1)*fi_r/(ro_m(1)*cp_m(1));
%Nodes interns
for p=2:n-1
    T_m_nova(p)=Fo(p)*(T_m(p+1)+T_m(p-1))+(1-
2*Fo(p))*T_m(p)+q_r(p)*dt*ro_r(p)*fi_r/(ro_m(p)*cp_m(p));
end
%Node resina-->entorn
T_m_nova(n)=Fo(n)*T_m(n-1)+(1-Fo(n)-
Bi*Fo(n))*T_m(n)+Bi*Fo(n)*T_ent+q_r(n)*dt*ro_r(n)*fi_r/(ro_m(n)*cp_m(n));
end

```

```
function [T_m_nova] =
f_programa2(T_ent,T_m,L_m,L_met,dt,q_r,fi_r,ro_r,ro_m,ro_met,cp_m,cp_met,h)

%Mostra
T_m_nova=h*dt/(ro_m*cp_m*L_m+ro_met*cp_met*L_met)*T_ent+(-
h*dt/(ro_m*cp_m*L_m+ro_met*cp_met*L_met)+1)*T_m+q_r*dt*ro_r*fi_r*L_m/(ro_m*cp_m*L
_m+ro_met*cp_met*L_met);
end
```

```
function [T_m_nova,T_met_nova] =
f_programa3(n,Bi,Fo,T_ent,T_m,T_met,L_met,dt,q_r,h_m_met,h_met_ent,fi_r,ro_r,ro_m
,ro_met,cp_m,cp_met)

%Node adiabàtic
T_m_nova(1)=2*Fo(1)*T_m(2)+(1-
2*Fo(1))*T_m(1)+q_r(1)*dt*ro_r(1)*fi_r/(ro_m(1)*cp_m(1));
%Nodes interns
for p=2:n-1
    T_m_nova(p)=Fo(p)*(T_m(p+1)+T_m(p-1))+(1-
2*Fo(p))*T_m(p)+q_r(p)*dt*ro_r(p)*fi_r/(ro_m(p)*cp_m(p));
end
%Node resina-->metall
T_m_nova(n)=Fo(n)*T_m(n-1)+(1-Fo(n)-
Bi*Fo(n))*T_m(n)+Bi*Fo(n)*T_met+q_r(n)*dt*ro_r(n)*fi_r/(ro_m(n)*cp_m(n));
%Node metall
T_met_nova=((-h_m_met-
h_met_ent)/(L_met*ro_met*cp_met)*dt+1)*T_met+h_m_met*dt/(L_met*ro_met*cp_met)*T_m
(n)+h_met_ent*dt/(L_met*ro_met*cp_met)*T_ent;
end
```

```
function [T_m_nova,T_met_nova] =
f_programa4(n,Bi1,Bin,Fo,T_ent,T_m,T_met,L_met,dt,q_r,h_m_met,h_met_ent,fi_r,ro_r
,ro_m,ro_met,cp_m,cp_met)

%Node resina-->entorn
T_m_nova(1)=Fo(1)*T_m(2)+(1-Bi1*Fo(1)-
Fo(1))*T_m(1)+Bi1*Fo(1)*T_ent+q_r(1)*dt*ro_r(1)*fi_r/(ro_m(1)*cp_m(1));
%Nodes interns
for p=2:n-1
    T_m_nova(p)=Fo(p)*(T_m(p+1)+T_m(p-1))+(1-
2*Fo(p))*T_m(p)+q_r(p)*dt*ro_r(p)*fi_r/(ro_m(p)*cp_m(p));
end
%Node resina-->metall
T_m_nova(n)=Fo(n)*T_m(n-1)+(1-Fo(n)-
Bin*Fo(n))*T_m(n)+Bin*Fo(n)*T_met+q_r(n)*dt*ro_r(n)*fi_r/(ro_m(n)*cp_m(n));
%Node metall-->entorn
T_met_nova=((-h_m_met-
h_met_ent)/(L_met*ro_met*cp_met)*dt+1)*T_met+h_m_met*dt/(L_met*ro_met*cp_met)*T_m
(n)+h_met_ent*dt/(L_met*ro_met*cp_met)*T_ent;
end
```