Lino Mediavilla, Carlos Cabezas, Paul Vidal
Proyecto de Mecánica Computacional – Módulo 1
Prof. Juan Baus
08/03/2017

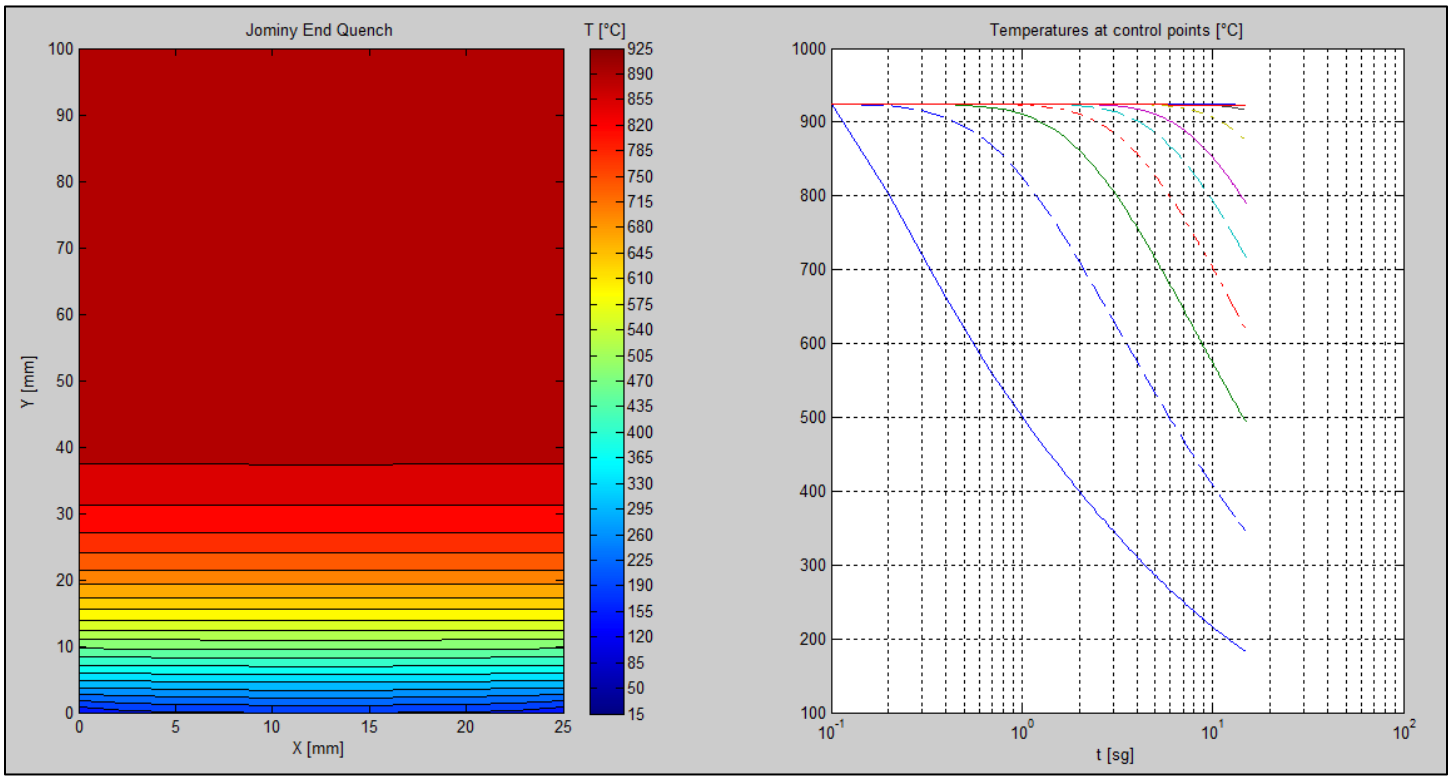
Anexo: Solución alternativa con método implícito

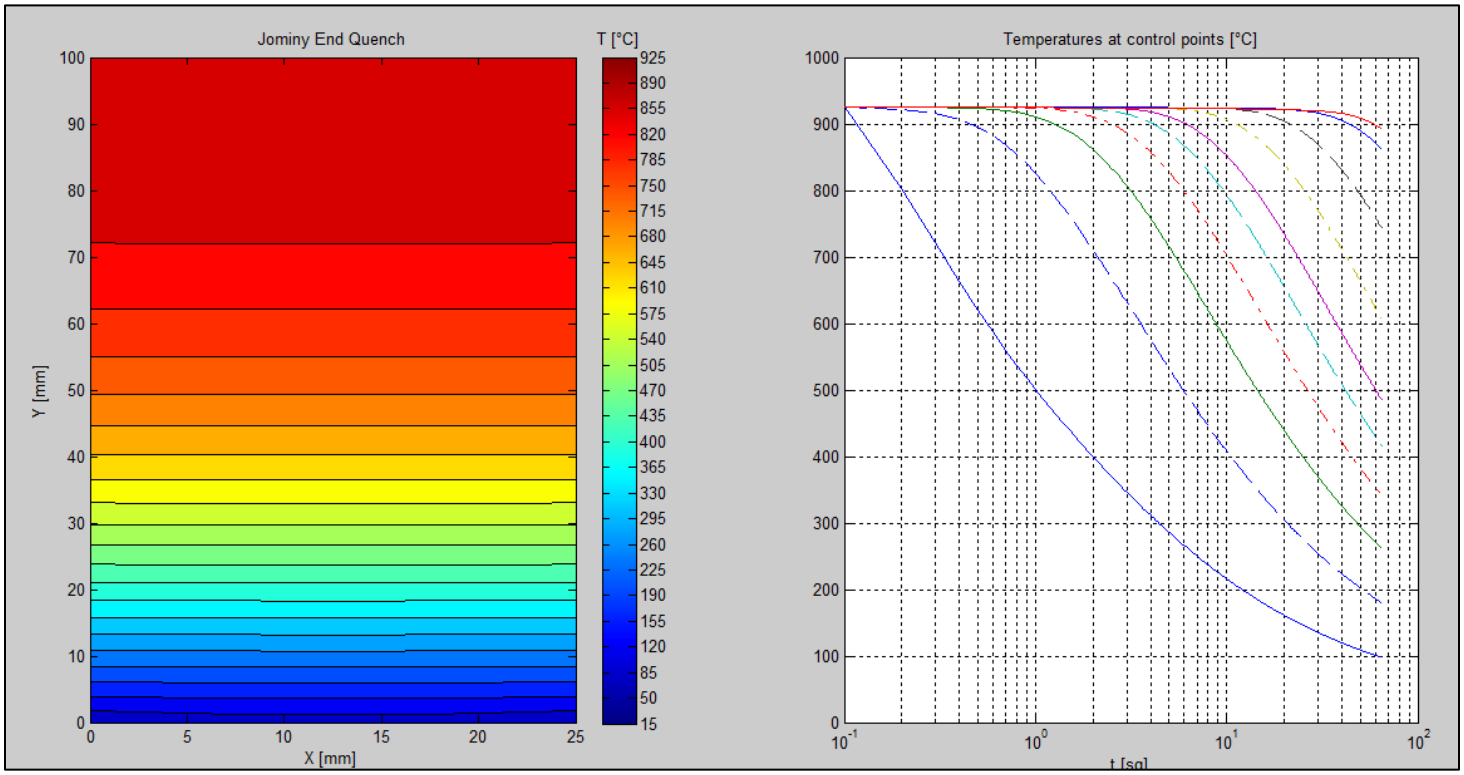
Se presenta una solución alternativa al problema de transferencia de calor en el ensayo de Jominy. Se optó por escribir un programa en Matlab que generara automáticamente el sistema lineal correspondiente a una malla estructurada. Anteriormente se trabajó con $\Delta x = \Delta y = 0.0050$ (126 ecuaciones), pero con el nuevo método es posible alterar esos valores sin los problemas de inestabilidad numérica característicos de un método explícito.

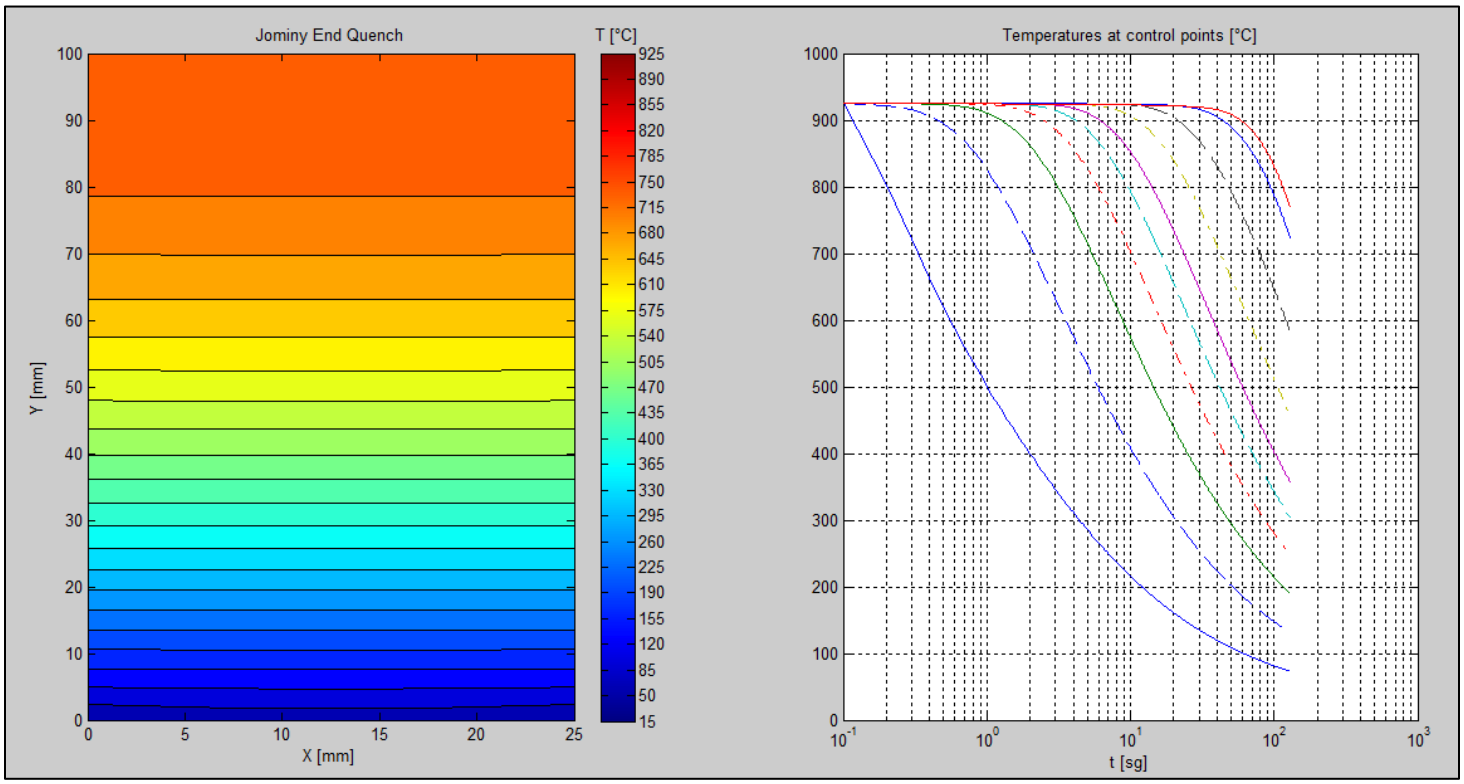
Trabajando con $\Delta x = \Delta y = 0.0025$, se generó una malla de 41 x 11 nodos (451 ecuaciones). Se estableció un tiempo de simulación máximo de 10^4 sg., y se escogió un $\Delta t = 0.1$ sg. para fines prácticos. (Si se buscara mayor precisión, se podría trabajar tranquilamente con un valor mucho más pequeño). Las ecuaciones utilizadas provienen del libro de Transferencia de Calor y Masa de Frank Incropera (2013, p. 334).

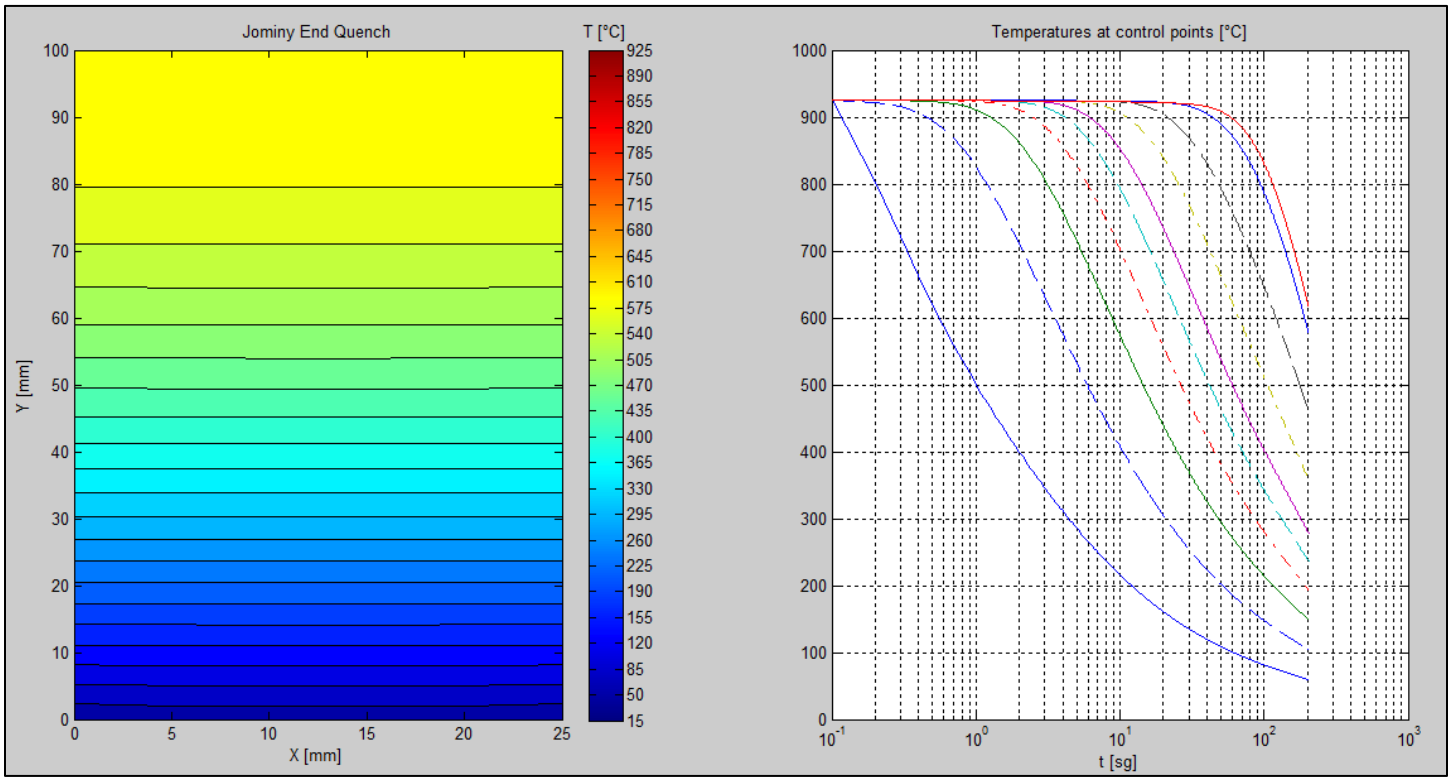
Resultados:

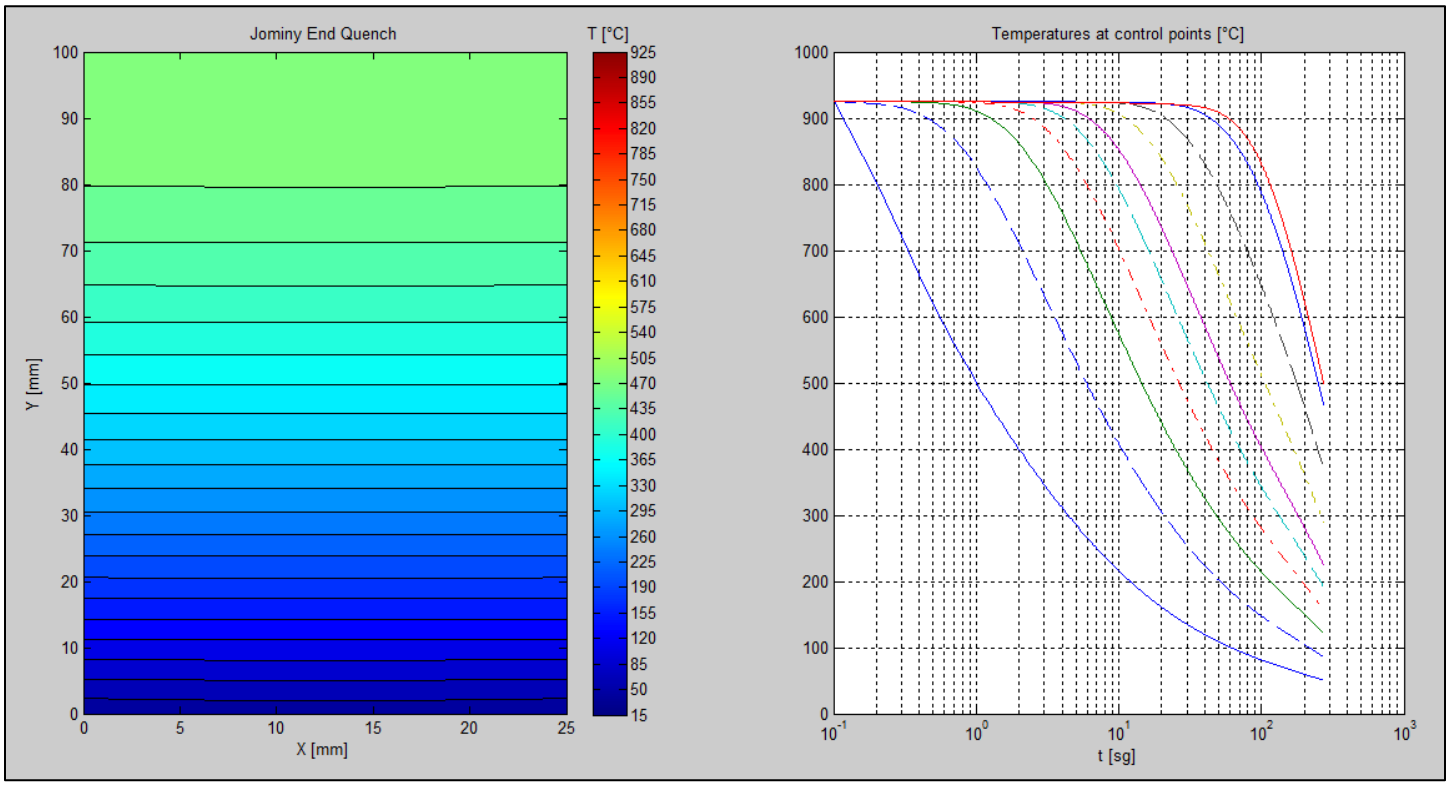
A continuación se muestran algunos resultados parciales del proceso transitorio

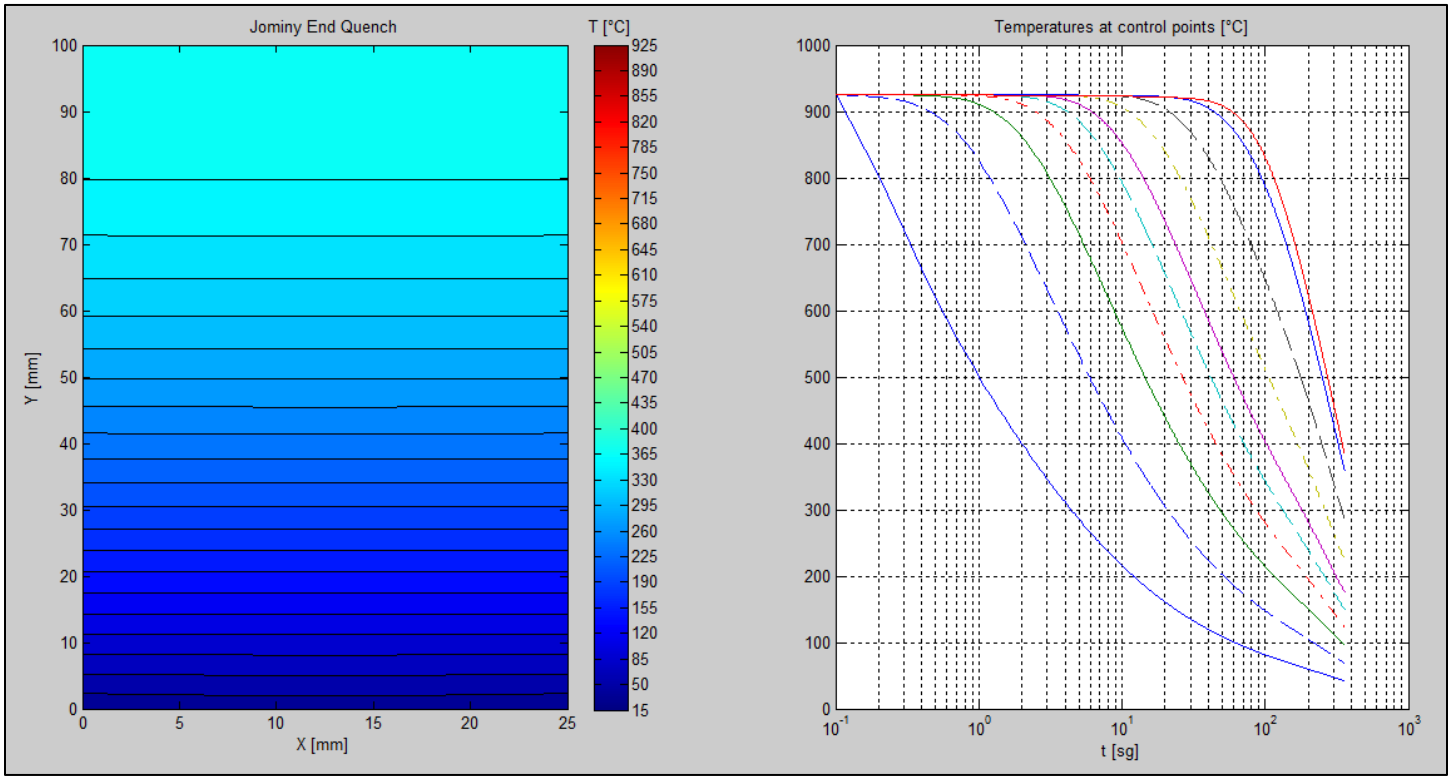


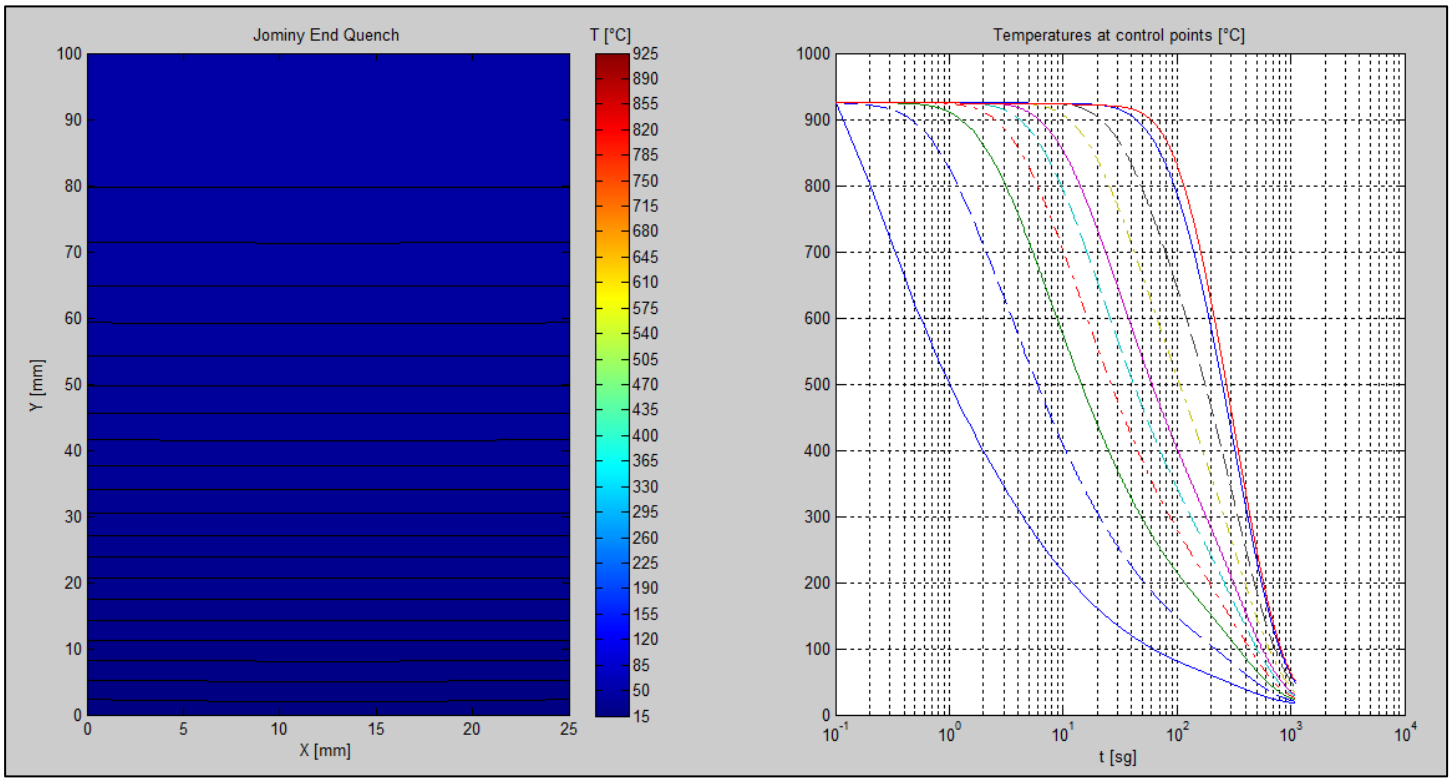

Jominy End Quench
T [°C]
925
890
855
820
785
750
715
680
645
610
575
540
505
470
435
400
365
330
295
260
225
190
155
120
85
50
15
Y [mm]
X [mm]
0
5
10
15
20
25
0
10
20
30
40
50
60
70
80
90
100
Temperatures at control points [°C]
1000
900
800
700
600
500
400
300
200
100
0
10^{-1}
10^{0}
10^{1}
10^{2}
t [sg]

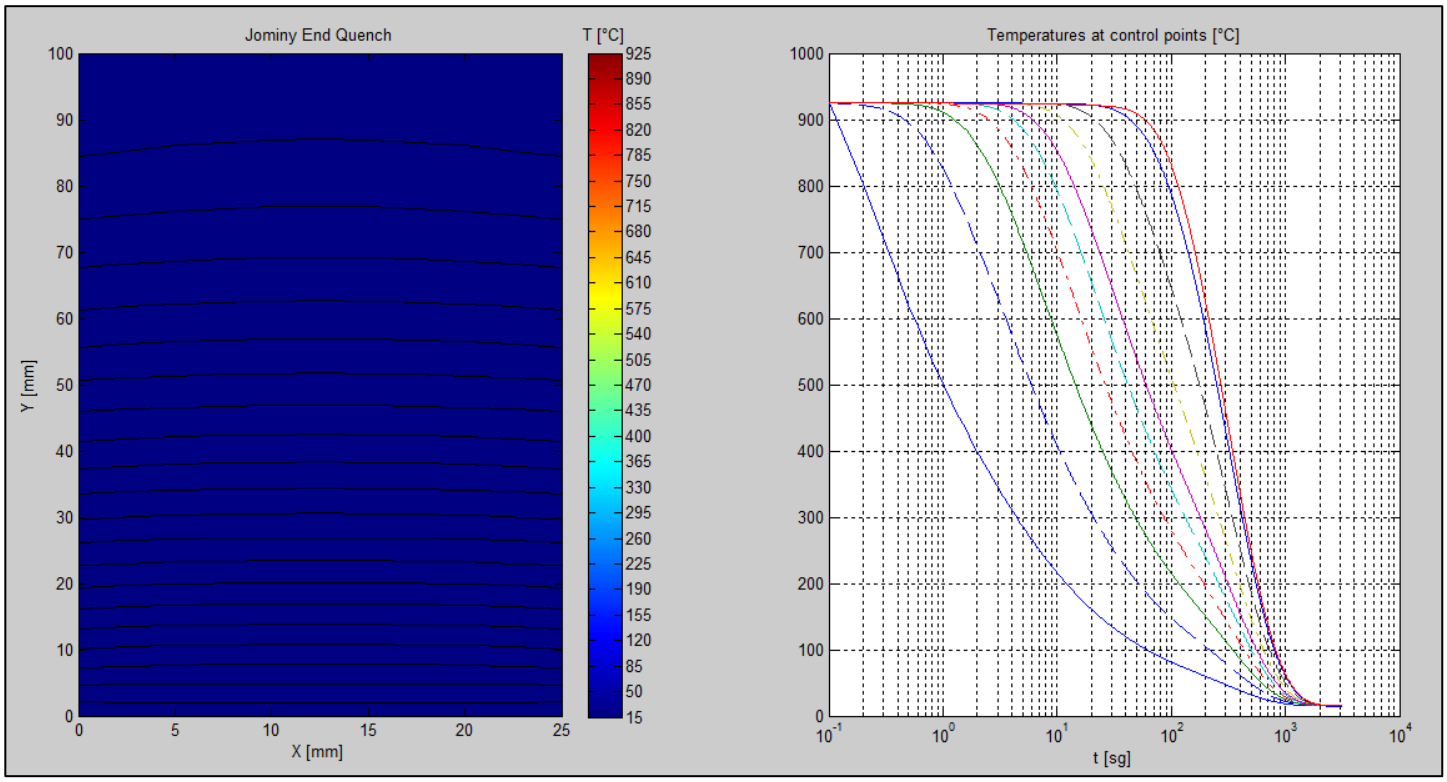

Jominy End Quench
T [°C]
925
890
855
820
785
750
715
680
645
610
575
540
505
470
435
400
365
330
295
260
225
190
155
120
85
50
15
Y [mm]
X [mm]
0
5
10
15
20
25
0
10
20
30
40
50
60
70
80
90
100
Temperatures at control points [°C]
1000
900
800
700
600
500
400
300
200
100
0
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
t [sg]


Jominy End Quench
T [°C]
925
890
855
820
785
750
715
680
645
610
575
540
505
470
435
400
365
330
295
260
225
190
155
120
85
50
15
Y [mm]
X [mm]
0
5
10
15
20
25
0
10
20
30
40
50
60
70
80
90
100
Temperatures at control points [°C]
t [sg]
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
0
100
200
300
400
500
600
700
800
900
1000

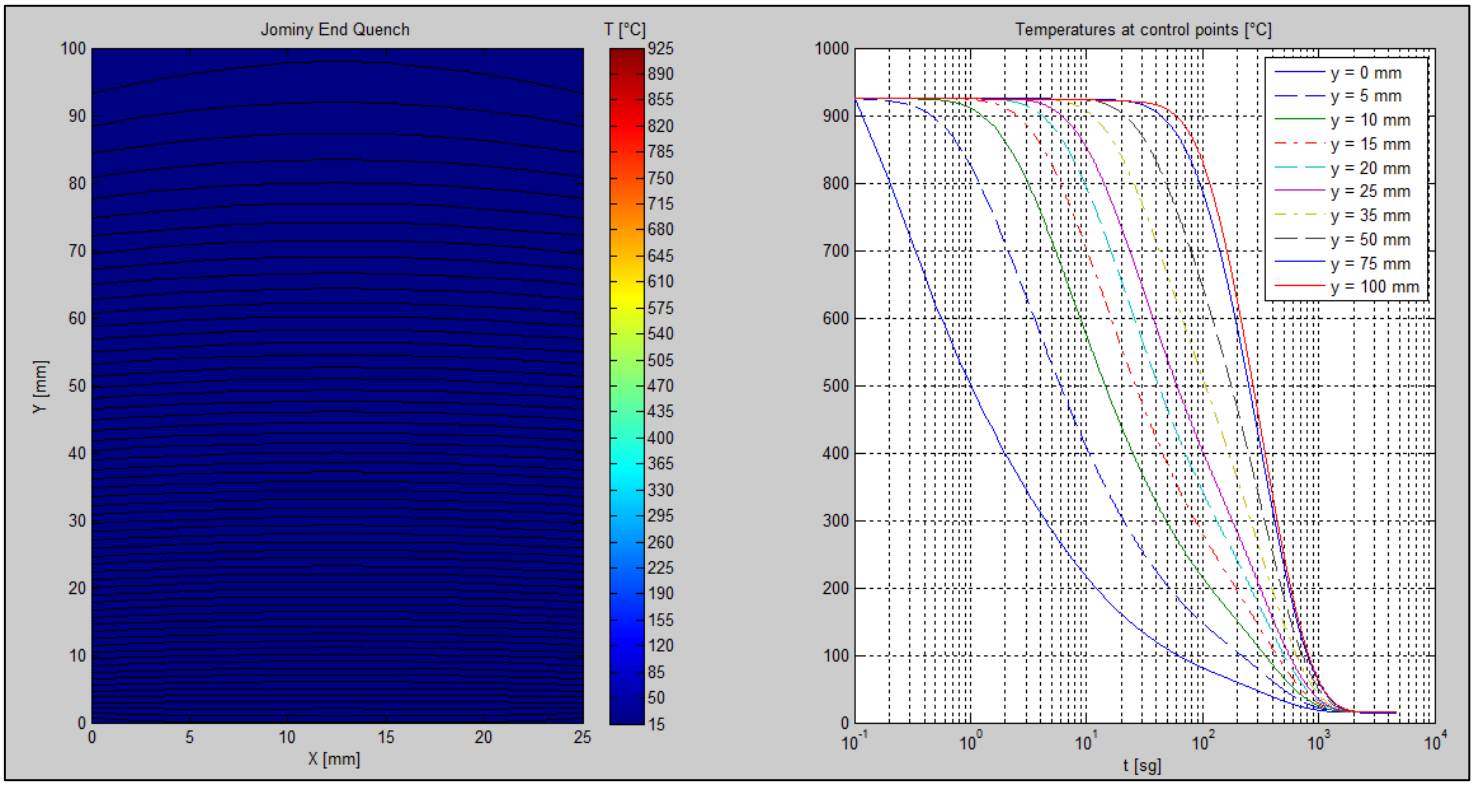

Jominy End Quench
T [°C]
925
890
855
820
785
750
715
680
645
610
575
540
505
470
435
400
365
330
295
260
225
190
155
120
85
50
15
Y [mm]
X [mm]
0
5
10
15
20
25
0
10
20
30
40
50
60
70
80
90
100
Temperatures at control points [°C]
t [sg]
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
0
100
200
300
400
500
600
700
800
900
1000


Jominy End Quench
T [°C]
925
890
855
820
785
750
715
680
645
610
575
540
505
470
435
400
365
330
295
260
225
190
155
120
85
50
15
Y [mm]
X [mm]
0
5
10
15
20
25
0
10
20
30
40
50
60
70
80
90
100
Temperatures at control points [°C]
t [sg]
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
0
100
200
300
400
500
600
700
800
900
1000


Jominy End Quench
T [°C]
925
890
855
820
785
750
715
680
645
610
575
540
505
470
435
400
365
330
295
260
225
190
155
120
85
50
15
Y [mm]
X [mm]
0
5
10
15
20
25
0
10
20
30
40
50
60
70
80
90
100
Temperatures at control points [°C]
t [sg]
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
0
100
200
300
400
500
600
700
800
900
1000



Y, el resultado final:



Código (Implementado en Matlab R2013b):

```
% PROBLEMA 2D - TRANSITORIO
% JOMINY
%
% Probeta:
%   |-  2.5 cm -|  (1 in.)
    %%%%%%%%%%%% -
    %           % |
    %           % |
    %           % 10 cm. (4 in.)
    %           % |
    %           % |
    %           % |
    %%%%%%%%%%%% -

clear
clc

%Condiciones del Experimento
h_agua = 10000; % [W/K*m^2]
T_agua = 15; %°C
h_aire = 5; %[W/K*m^2]
T_aire = 25; %°C

%Temperatura inicial de la probeta
To = 925; %[°C]

%Propiedades del material (AISI 1018)
K = 51.9; % [W/m*K]
densidad = 7.872*10^3; % [kg/m^3]
Cp = 486 ;% [J/(kg·K)]
alpha = K /(densidad*Cp); %[m^2/s]

%Dimensiones de la Probeta
B = 0.025; % base de la geometría [m]
H = 0.1; % altura de la geometría [m]

%Generación de la red nodal
%(dx = dy = 0.0050 -> tamaño: 21 fil. x  6 col.) -> dt máx: 0.2346 [sg]
dx = 0.0025;  % m
dy = 0.0025;  % m
nX = (B / dx)+1
nY = (H / dy)+1

x = 1000.*(0:dx:B);
y = 1000.*(H:-dy:0);
[X,Y] = meshgrid(x,y);

%Declaración de vectores vacíos para almacenar temperaturas en
%bordes superior e inferior de la geometría.
bot_edge_Temp = [];
top_edge_Temp = [];

%Declaración de vectores vacíos para almacenar temperaturas en
%puntos específicos de la geometría.
punto_control_1 = [];
punto_control_2 = [];
punto_control_3 = [];
punto_control_4 = [];
punto_control_5 = [];
punto_control_6 = [];
```

```
punto_control_7 = [];
punto_control_8 = [];

%Fo, Bi, Time Step

dt = 0.1; %[sg] -- max: 0.2346 [sg]
tMax = 10^4; %[sg]
t = []; %Vector vacío para guardar los tiempos
p = 1;
maxSteps = uint32(tMax/dt);
tol = 1e-7;

Fo = alpha*dt /(dx^2)
Bi_agua = h_agua*dx/K
Bi_aire = h_aire*dx/K

T  = To*ones(nY*nX,1);
mat_T = zeros(nY,nX);
C = zeros(nY*nX,1);
A = zeros(nX*nY, nX*nY);
display('Simulación en curso ...');



while p < maxSteps+1
  if(p>1)
      for i = nX+2:nX*nY-nX-1
           A (i,: )= [ zeros(1,i-nX-1) -Fo zeros(1,nX-2) -Fo 1+4*Fo -Fo zeros(1,nX-2) ...
               -Fo zeros(1,nX*nY-i+1+nX-2*(nX-2)-5)];
      end

    %Se actualizan todos los nodos que no sean interiores
      for i = 1:nX*nY

          if(i == 1)
          %Esquina superior izquierda
            A(i,:) = [1+4*Fo*(1+Bi_aire) -2*Fo zeros(1,nX-2) -2*Fo zeros(1,nX*nY-(3+nX-2))];

          elseif(i == nX)
          %Esquina superior derecha
            A(i,:) = [ zeros(1,nX-2) -2*Fo 1+4*Fo*(1+Bi_aire) zeros(1,nX-1) -2*Fo ...
                zeros(1,nX*nY - 3 - (nX-1) - (nX-2))];
          %Nodo (nX+1) - justo debajo de la primera fila, pared izquierda.
            A(i+1,:) = [zeros(1,((i/nX)-1)*nX) -Fo zeros(1,nX-1) (1+2*Fo*(2+Bi_aire)) ...
                -2*Fo zeros(1,nX-2) -Fo zeros(1,nX*nY-(nX-1)-(nX-2)-4-( ((i/nX)-1)*nX))];

          elseif (i == nX*nY)
          %Esquina inferior derecha
            A(i,:) = [zeros(1,nX*nY-(3+nX-2))  -2*Fo zeros(1,nX-2) -2*Fo  1+4*Fo*(1+Bi_agua)];

          elseif (i == nX*nY-nX+1)
          %Esquina inferior izquierda
          A(i,:) = [zeros(1,nX*nY-2*nX) -2*Fo zeros(1,nX-1) 1+4*Fo*(1+Bi_agua) -2*Fo ...
              zeros(1,nX*nY-3-(nX-1)-(nX*nY-2*nX))];

          elseif (i > nX && i <nX*nY && rem(i,nX) == 0)
           %Convección en Pared derecha
            A(i,:) = [zeros(1, ((nX-1)+((i/nX)-2)*nX)) -Fo zeros(1,nX-2) -2*Fo (1+2*Fo*(2+Bi_aire))
...
                zeros(1,nX-1) -Fo zeros(1,nX*nY-(nX-2)-(nX-1)-4-((nX-1)+((i/nX)-2)*nX))];

            if(i < (nX*nY-nX))
```

```
            %Convección en Pared izquierda
            A(i+1,:) = [zeros(1,((i/nX)-1)*nX) -Fo zeros(1,nX-1) (1+2*Fo*(2+Bi_aire)) ...
                -2*Fo zeros(1,nX-2) -Fo zeros(1,nX*nY-(nX-1)-(nX-2)-4-( ((i/nX)-1)*nX))];
          end

        elseif (i > 1 && i < nX)
         %Convección en Pared superior
         A(i,:) = [zeros(1,i-2) -Fo (1+2*Fo*(2+Bi_aire)) -Fo zeros(1,nX-2) -2*Fo ...
             zeros(1,nX*nY-(i-2)-4-(nX-2)) ];

        elseif (i > nX*nY-nX+1 && i < nX*nY)
         %Convección en Pared inferior
         A(i,:) =  [zeros(1,i-nX-1)  -2*Fo zeros(1,nX-2) -Fo (1+2*Fo*(2+Bi_agua)) -Fo ...
             zeros(1, nX*nY -4 -(nX-2) -( i-nX-1 ))];

        end
    end

    C = T;

    for i=1:nX*nY
        %Se actualizan todos los nodos que no sean interiores

        if(i == 1 || i == nX)
        %Convección en esquinas Superiores
          C(i,1) = T(i,1)+4*Bi_aire*Fo*T_aire;
        %Nodo (nX+1) - justo debajo de la primera fila, pared izquierda.
          C(i+1,1) = T(i+1,1)+2*Bi_aire*Fo*T_aire;

        elseif (i == nX*nY || i == nX*nY-nX+1)
        %Convección en esquinas Inferiores
          C(i,1) = T(i,1)+4*Bi_agua*Fo*T_agua;

         elseif (i > 1 && i < nX)
         %Convección en Pared superior
         C(i,1) = T(i,1)+2*Bi_aire*Fo*T_aire;

        elseif (i > nX*nY-nX && i < nX*nY)
         %Convección en Pared inferior
         C(i,1) = T(i,1)+2*Bi_agua*Fo*T_agua;

        elseif (rem(i,nX) == 0)
         %Convección en Pared derecha
          C(i,1) = T(i,1)+2*Bi_aire*Fo*T_aire;

          if(i < (nX*nY-nX-1))
              %Convección en Pared izquierda
              C(i+1,1) = T(i+1,1)+2*Bi_aire*Fo*T_aire;
          end
        end

    end

      %Resolución simultánea del sistema de ecuaciones
      T = A\C;
end
  %Para ordenar las temperaturas en la forma de la geometría
  for j = 1:nY
     mat_T(j,:) = T(nX*(j-1)+1:j*nX,1);
  end
```

```
    t = [t (p*dt)]; %anexar cada tiempo al vector de tiempos.

    bot_edge_Temp = [bot_edge_Temp;mat_T(nY,3);];
    punto_control_1 = [punto_control_1;mat_T(nY-0.005/dy,3)];
    punto_control_2 = [punto_control_2;mat_T(nY-0.010/dy,3)];
    punto_control_3 = [punto_control_3;mat_T(nY-0.015/dy,3)];
    punto_control_4 = [punto_control_4;mat_T(nY-0.020/dy,3)];
    punto_control_5 = [punto_control_5;mat_T(nY-0.025/dy,3)];
    punto_control_6 = [punto_control_6;mat_T(nY-0.035/dy,3)];
    punto_control_7 = [punto_control_7;mat_T(nY-0.050/dy,3)];
    punto_control_8 = [punto_control_8;mat_T(nY-0.075/dy,3)];
    top_edge_Temp = [top_edge_Temp;mat_T(1,3);];

    %Graficar cada 350 pasos
     if(p == 1 || uint16(p/50) == p/50)

        h1 = subplot(1,2,1);
        ax1 = get(h1,'position');
        set(h1,'position',ax1);
        contourf(X,Y,mat_T,20)
        title('Jominy End Quench');
        xlabel('X [mm]');
        ylabel('Y [mm]');
        colormap(jet(250))
        caxis( [T_agua To] );
        barra = colorbar;
        set(barra,'YTick',[T_agua:35:To]);
        set(get(barra,'title'),'string','T [°C]');

        subplot(1,2,2);
        semilogx(t,bot_edge_Temp,'b',t,punto_control_1,'--',...
            t,punto_control_2,'-',t,punto_control_3,'-.',t,punto_control_4,'--',...
            t,punto_control_5,'-',t,punto_control_6,'-.',t,punto_control_7,'--',...
            t,punto_control_8,t,top_edge_Temp,'r');
        grid on
        title('Temperatures at control points [°C]');
        xlabel('t [sg]');
        drawnow
     end

     %Criterio de parada :: ¿Es la diferencia de temperaturas de un mismo
     %punto en dos puntos temporales consecutivos menor a la tolerancia?
     % Si -> break
     % No -> seguir iterando
    if length(top_edge_Temp)>=2
         if (abs((top_edge_Temp(length(top_edge_Temp))-top_edge_Temp(length(top_edge_Temp)-
1)))<=tol)
            break
         end
    end

    p = p + 1;

end


display('Simulación Terminada!');
Tiempo_transcurrido = p*dt
```

```
h1 = subplot(1,2,1);
ax1 = get(h1,'position');
set(h1,'position',ax1);
contourf(X,Y,mat_T,60)
title('Jominy End Quench');
xlabel('X [mm]');
ylabel('Y [mm]');
colormap(jet(250))
caxis( [T_agua To] );
barra = colorbar;
set(barra,'YTick',[T_agua:35:To]);
set(get(barra,'title'),'string','T [°C]');

subplot(1,2,2);
semilogx(t,bot_edge_Temp,'b',t,punto_control_1,'--',...
            t,punto_control_2,'-',t,punto_control_3,'-.',t,punto_control_4,'--',...
            t,punto_control_5,'-',t,punto_control_6,'-.',t,punto_control_7,'--',...
            t,punto_control_8,t,top_edge_Temp,'r');
grid on
legend('y = 0 mm','y = 5 mm','y = 10 mm','y = 15 mm','y = 20 mm','y = 25 mm','y = 35 mm','y = 50
mm','y = 75 mm','y = 100 mm')
title('Temperatures at control points [°C]');
xlabel('t [sg]');
```

Conclusiones

Los resultados obtenidos con este método se asemejan mucho a los obtenidos con el método explícito, esto podría deberse a que el mallado escogido con el anterior método ya era relativamente fino. Sin embargo, explorar la solución con el método implícito permitió comprobar los efectos de una malla más fina debido a que no existen las restricciones que surgen con el método explícito.

Referencias

Incropera, F. P. (2013). *Fundamentals of heat and mass transfer*. Hoboken, NJ: Wiley.