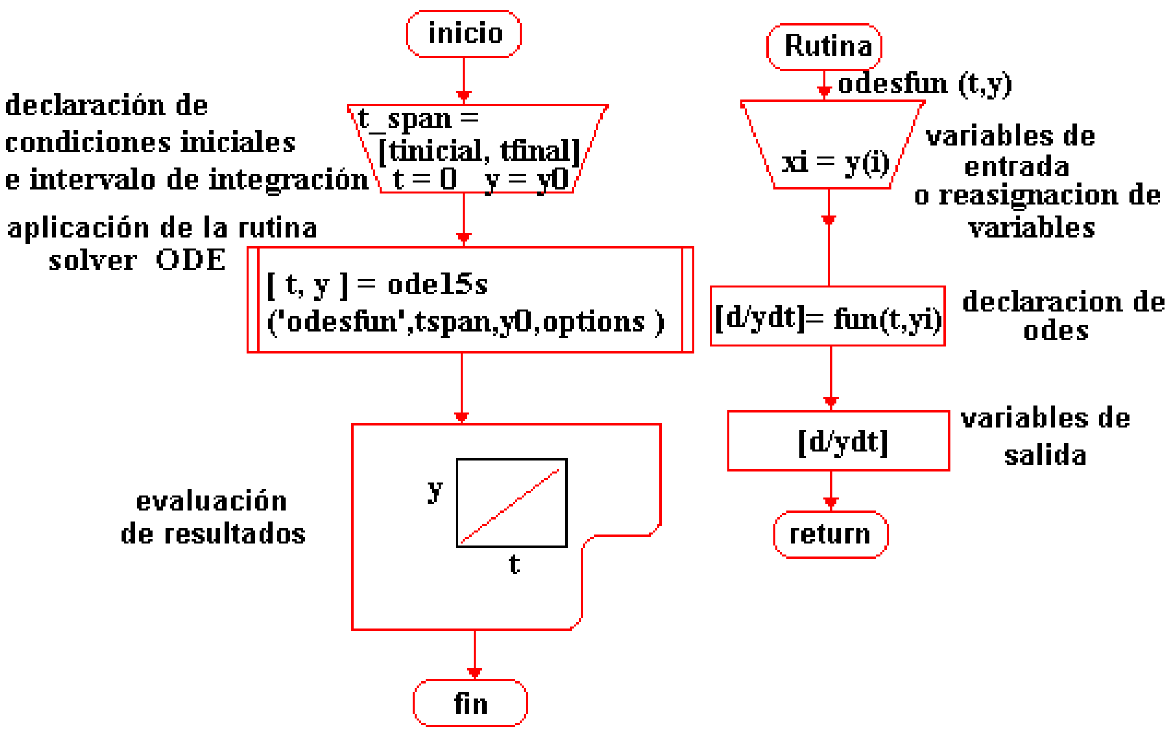
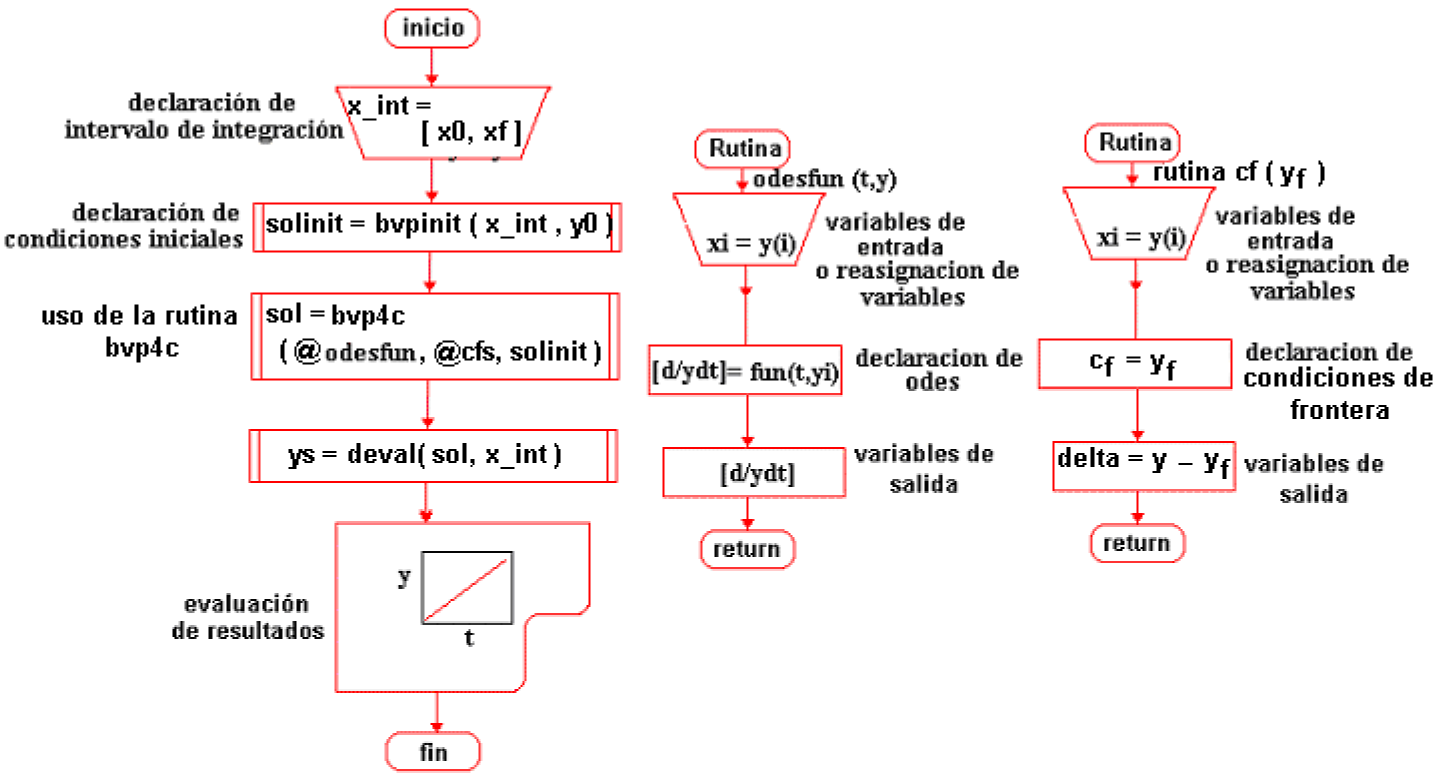
Técnicas de solución

1. A través del método de disparo

Estructura para resolver sistemas de ecuaciones diferenciales ordinarias con valores a la frontera



2. Utilizando la rutina BVP4C



La rutina bvp4c permite resolver ecuaciones diferenciales ordinarias con dos valores a la frontera

Sintaxis:

sol = bvp4c(@odefun , @bcfun , solinit)[1]

Argumentos de entrada :

Odefun : rutina que contiene el conjunto de ecuaciones diferenciales[2]

Bcfun : rutina que calcula el residual de las condiciones de frontera cf (y(a),y(b)) y puede tener la forma res = bcfun(ya , yb)[3]

Solinit : estimado de valores en la frontera inicial, puede tener la forma

solinit = bvpinit(x_int,y0)[4]

Ejemplo 1 [Código]

Trace la respuesta de la función solución para

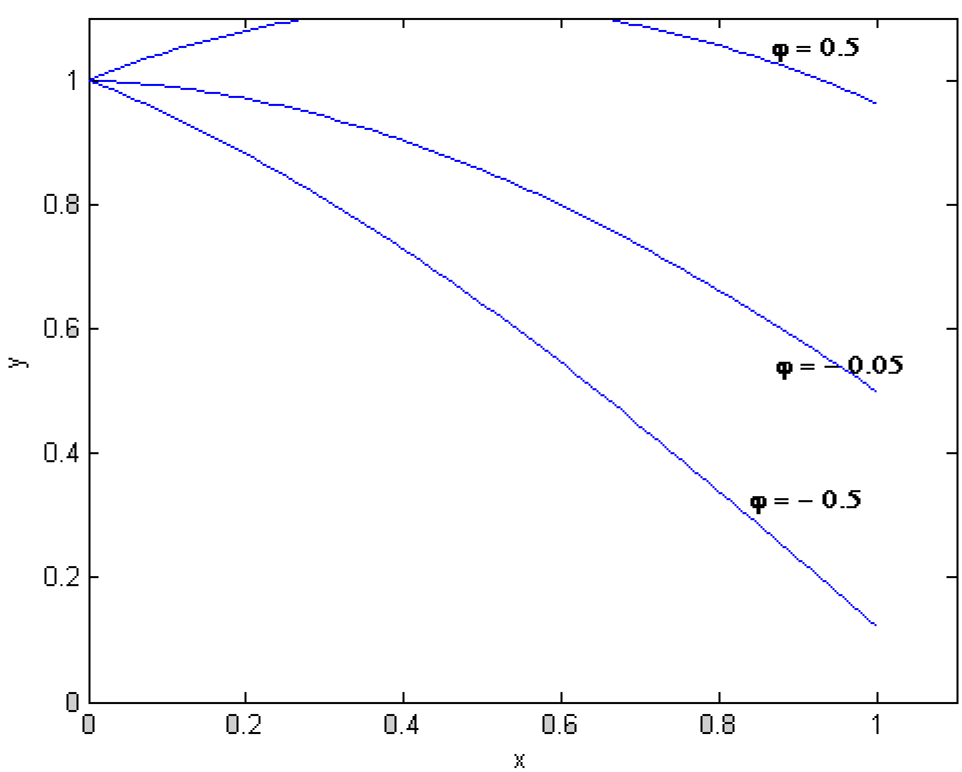
$\frac{d^2y}{dx^2} + y = 0$ con la condiciones frontera *cf1 x = 0 y = 1 & cf2 x = 1 y = 0.5*

aplicando el método de disparo y la rutina de *BVP4C*

Solución

La ecuación diferencial ordinaria de segundo orden tendrá que ser expresada como dos ecuaciones diferenciales ordinarias de primer orden como sigue

Si $\varphi = \frac{dy}{dx}$ entonces EDO1: $\frac{dy}{dx} = \varphi$ & EDO2: $\frac{d\varphi}{dx} = -y$ con las condiciones iniciales Ci1: $x = 0 \quad y = 1$, Ci2: $x = 0 \quad \varphi = \varphi_s$, en donde el valor de φ_s será supuesto hasta que se cumpla la condición de frontera $x = 1 \quad y = 0.5$



Ejemplo 2 [Código solución]

Se desea resolver el siguiente sistema de ecuaciones diferenciales ordinarias con valores a la frontera

$$\frac{du}{dx}=\frac{u(w-u)}{2v} \quad (1) \qquad \frac{dv}{dx}=-\frac{(w-u)}{2} \quad (2) \qquad \frac{dw}{dx}=\frac{0.9-1000(w-y)-0.5w(w-u)}{z} \quad (3)$$

$$\frac{dz}{dx}=\frac{w-u}{2} \quad (4) \qquad \frac{dy}{dx}=-100(y-w) \quad (5)$$

con las condiciones de frontera $u(0)=v(0)=w(0)=1 \;\; z(0)=-10 \;\&\; w(1)=y(1)$

si se definen las variables siguientes y(1) = u, y(2) = v, y(3) = w, y(4) = z, & y(5) = y

Problemas de Tarea

Resuelva

1. $\frac{d^3y}{dx^3}-3\frac{d^2y}{dx^2}+\frac{dy}{dx}-3=0$ con la condiciones frontera *cf1 x = 0 y = 2 & cf2 x = 0 y = 6 cf3 x = 0 y = 0 & cf3 x =10 y = 0.7*

2. $\frac{dx}{dt}=y-1 \qquad \frac{dy}{dt}=-3x+2y$ cf $x(0)=0 \qquad y(0)=0$

3. Se lleva a cabo una reacción de primer orden $\mathbf{A} \rightarrow \mathbf{B}$ en fase gaseosa en una partícula catalítica sólida, escriba los modelos de difusión-reacción no isotérmica unidireccional, considerando las geometrías

   a. Rectangular, b. Cilíndrica & c. Esférica,

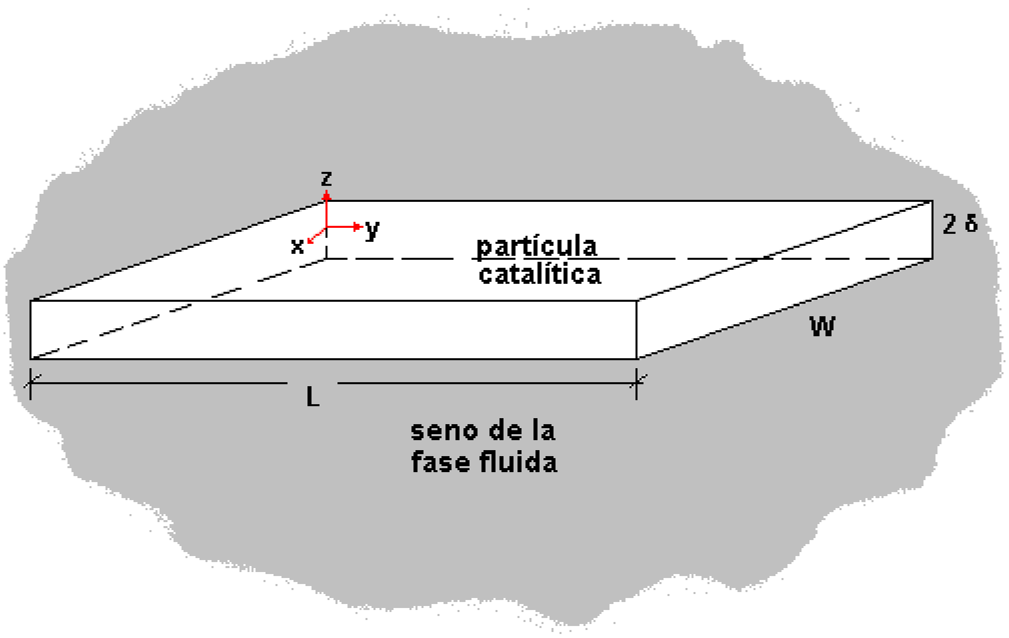
exprese su modelo en forma adimensional con las variables siguientes $\varepsilon=\frac{r}{R}$, $C=\frac{C_A}{C_{As}}$ & $\theta=\frac{T}{Ts}$

y trace la respuesta de los perfiles de concentración y temperatura

Datos adicionales

Si = 9.1m^2/g cat área superficial
ΔHr = -80000 J/mol calor de reacción
R = 8.314 J/ mol K constante universal de los gases
De = 8e-8 m^2/s difusividad efectiva
Cas =10 mol/m^3 concentración en la superficie del catalizador
Ts = 400 K temperatura en la superficie del catalizador
Ea = 120000 J/mol energía de activación
Kt =0.004 J/ m s K conductividad térmica del catalizador
kr0 = 0.1 m/s a 400 K coeficiente cinético específico
dp = 0.005 m diámetro de la partícula catalítica
ρc = 1.1e3 g/m^3 densidad del catalizador

Solución caso rectangular [Código en MATLAB]



El conjunto de ecuaciones en forma dimensional de los balances de masa y energía para la partícula catalítica son

$$D_{AM}\frac{d^2C_A}{dz^2} = kr\rho cScC_A \qquad kt\frac{d^2T}{dz^2} = -(-\Delta Hr)kr\rho cScC_A \qquad (1)$$

$$CF1 \qquad z = 0 \qquad \frac{dC_A}{dz} = 0 \qquad \frac{dT}{dz} = 0 \quad (2)$$

$$CF2 \qquad z = L \qquad CA = C_{AS} \qquad T = T_S \quad (3)$$

y en forma adimensional se tiene

$$\frac{d^2C}{d\varepsilon^2} = \frac{kr\rho cScL^2}{D_{AM}}C \qquad \frac{d^2\theta}{d\varepsilon^2} = -\frac{L^2(-\Delta Hr)kr\rho cScC}{kt} \quad (4)$$

$$kr = kr_0 e^{-\frac{Ea}{RTr}\left(1-\frac{Tr}{Ts}\frac{1}{\theta}\right)} \quad (5)$$

$$CF1 \qquad \varepsilon = 0 \qquad \frac{dC}{d\varepsilon} = 0 \qquad \frac{d\theta}{d\varepsilon} = 0 \quad (6)$$

$$CF2 \qquad \varepsilon = 1 \qquad C = 1 \qquad \theta = 1 \quad (7)$$

este problema matemático de ecuaciones diferenciales ordinarias de segundo orden con valores a la frontera se pueden expresar en un sistema EDO de primer orden como sigue

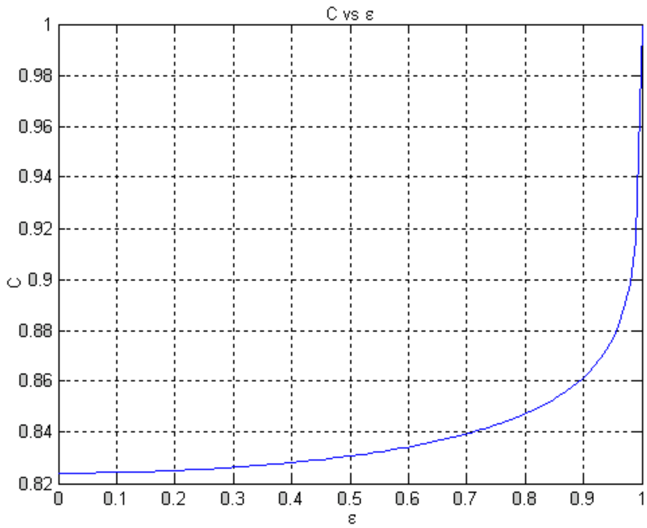
$$\frac{dC}{d\varepsilon} = \varphi \quad (8) \qquad \frac{d\varphi}{d\varepsilon} = \frac{kr\rho cScL^2}{D_{AM}}C \quad (9) \qquad \frac{d\theta}{d\varepsilon} = \psi \quad (10)$$

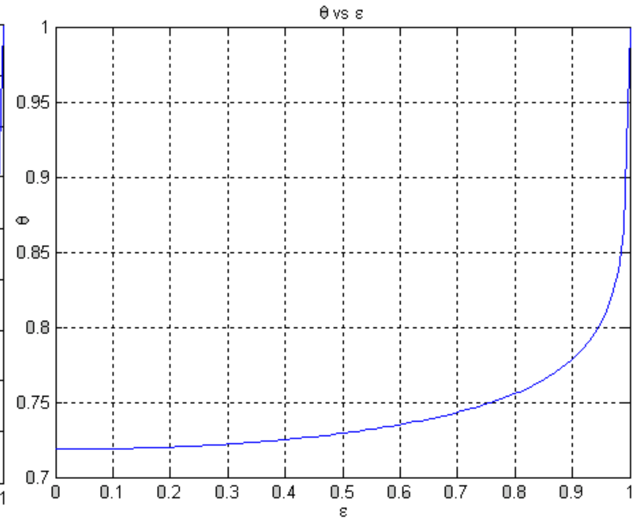
$$\frac{d\psi}{d\varepsilon} = -\frac{L^2(-\Delta Hr)kr\rho cScC}{kt} \quad (11)$$

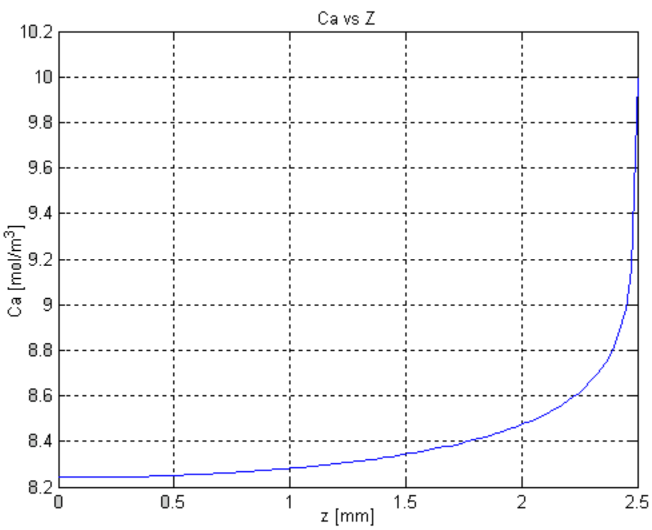
$$CF1 \qquad \varepsilon = 0 \qquad \varphi = 0 \qquad \psi = 0 \quad (12)$$

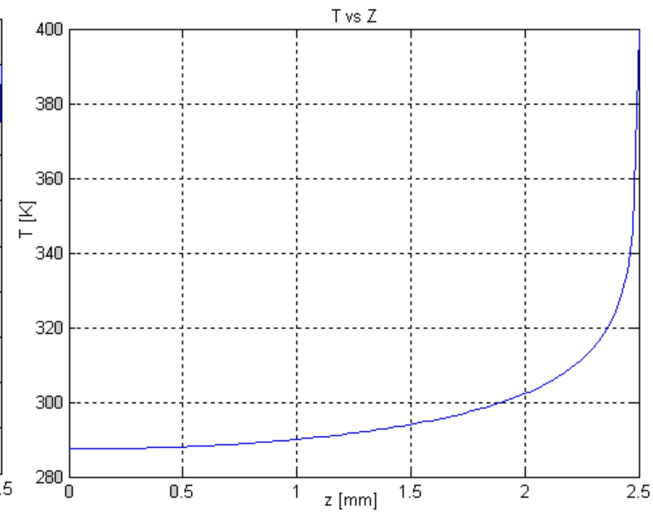
$$CF2 \qquad \varepsilon = 1 \qquad C = 1 \qquad \theta = 1 \quad (13)$$

este sistema se puede resolver numéricamente a través del método de disparo o con la rutina bvp4c de MATLAB


C vs ε
C
ε


θ vs ε
θ
ε


Ca vs Z
Ca [mol/m3]
z [mm]


T vs Z
T [K]
z [mm]

uso de bvp4c para el problema en régimen permanente

código ejemplo 1 [Regresar]

```
% inicio del archivo edo2o.m
% solucion de la ODE de 2° orden
%      y'' + y = 0
% cf1  y(0) = 1
% cf2  y(1) = 0.5
% a partir de los metodos de disparo y
% la rutina vbp4c de matlab

% ---------------------
   function edo2o
% ---------------------
clc; clear all; format compact
x_int = linspace(0,1);% intervalo de integracion
% condiciones iniciales
y10 = 1; phi0 = 1;
% vector de condiciones iniciales
y0 = [y10 phi0 ];
caso=input('Metodo a utilizar: [1] disparo, [2] BVP4C :
');
if caso==1
   fprintf('----  utilizo el metodo del disparo ----\n')

   [x,ys] = ode15s (@odevi,x_int,y0);

   y = ys(:,1);
   % trazo de la respuesta
   figure(1), plot(x,y),xlabel(' x ')
   ylabel(' y '), grid
   axis ([0 1.1 0 1.1])
elseif caso==2
   fprintf('----  utilizo el metodo BVP4C ----\n')
   % propuesta para valores en la frontera inicial
   % [regresa]
   solinit = bvpinit(x_int,y0);
   % aplicacion de la rutina  bvp4c para resolver el
ODE
%  [regresar]
   sol = bvp4c(@odevi,@cfs,solinit);
   % evaluacion de la solucion en el intervalo de la
posicion
   ys = deval(sol,x_int)';
   % reasignacion de variables
   y = ys(:,1);
   figure(1), plot(x_int,y),xlabel(' x ')
   ylabel(' y '), grid
   axis ([0 1.1 0 1.1])
else
   fprintf('Error en la especificacion del metodo ...')
end

% fin de la funcion edo2o
```

```
% [regresa]
% -------------------------------
   function dvardx = odevi(x,var)
% -------------------------------
y = var(1); phi = var(2);

dydx = phi;
dphidx = -y;

dvardx = [dydx dphidx]';
% fin de la funcion odevi
```

```
% inicio del archivo cfs.m [regresa]
% ---------------------------------------------------
function [rescf]=cfs(y0,yL)
% ---------------------------------------------------

% rutina para la declaracion de las condiciones de
frontera
% expresado en forma residual
% - entrada      y0: condiciones de frontera en extremo
inicial
%               yL: condiciones de frontera en extremo
final
% - salida       rescf: residual de las condiciones de
frontera
%                    y0 - cf1
%                    yL - cf2
%

y_x0 = 1;      % condicion de frontera x = 0
y_xL = 0.5;     % condicion de frontera x = 1

%calculo de los residuales para las condiciones de
frontera
rescf(1) = y0(1) - y_x0;
rescf(2) = yL(1) - y_xL ;

rescf=rescf';
% fin del archivo cfs.m
```

Ejemplo 2 [Regresar]

```
% inicio del archivo ex1paper.m

%-------------------------
function ex1paper
%-------------------------

clc; clear all; format compact;

% definicion del intervalo de integracion
x_int = linspace(0,1,5);

% estimado inicial para la busqueda de la solucion
solinit = bvpinit(x_int,[1 1 1 -10 0.91]);

% uso de la rutina bvp4c para resolver el sistema
sol = bvp4c(@ex1ode,@ex1bc,solinit);

% evaluacion de la solucion en el intervalo de
integracion
ys = deval(sol,x_int)';

% trazo de la solucion
for i=1:length(ys)
   figure(i), plot(x_int,ys(:,i))
   xlabel(' x '),ylabel(' y_i '), grid
end

%-------------------------
function dydx = ex1ode(x,y)
%-------------------------

dydx = [ 0.5*y(1)*(y(3) - y(1))/y(2)
-0.5*(y(3) - y(1))
(0.9 - 1000*(y(3) - y(5)) - 0.5*y(3)*(y(3) - y(1)))/y(4)
0.5*(y(3) - y(1))
100*(y(3) - y(5))];


%-------------------------
function res = ex1bc(ya,yb)
%-------------------------

res = [ ya(1) - 1
        ya(2) - 1
        ya(3) - 1
        ya(4) + 10
        yb(3) - yb(5)];

% fin del archivo ex1paper.m
```

Codigo problema 3 [Regresar]

```
% inicio del archivo catalitico2.m
%
% codigo para resolver el problema de reacción de primer orden A -> B en fase
% gaseosa en una partícula catalítica sólida con problema de difusión-reacción
% no isotérmica unidireccional en una tableta rectangular
%

% ----------------------------------------------------
function catalitico2
% ----------------------------------------------------
clc; clear all; format compact;
global Si dHr De Ea kt kr0 R Tr rhoc Ts Cas Rp
% ----------------------------------------------------
%datos del problema
% ----------------------------------------------------
Si=9.1;%m2/g cat
dHr=-80000;%J/mol
R=8.314;%J/ mol K
De=8e-8;%m2/s
Cas=10;%mol/m3
Ts=400;%K
```

```
Tr=400;%K
Ea=120000;%J/mol
kt=0.004;%J/ m s K
kr0=0.1;%m/s a 400 K
dp=0.005;%m
Rp=dp/2;%m
rhoc=1.1e3;%g/m3
% ------------------------------------------------------
% especificacion del vector de posicion
E=linspace(0,1);
% propuesta para valores en la frontera inicial
solinit = bvpinit(E,[ 0 0 0.75 0]);
% aplicacion de la rutina  bvp4c para resolver el ODE
sol = bvp4c(@slab2,@cf2,solinit);
% evaluacion de la solucion en el intervalo de la posicion
y = deval(sol,E)';
% reasignacion de variables
C=y(:,1);   Teta=y(:,3);
% calculo de concentracion y temperatura dimensionales
z=E*Rp*1000; Ca=C*Cas; T=Ts*Teta;
% ------------------------------------------------------
% impresion de resultados
% ------------------------------------------------------
figure(1), plot(E,C),grid
xlabel ('\epsilon'), ylabel ('C')
title('C vs \epsilon')

figure(2), plot(E,Teta), grid
xlabel ('\epsilon'), ylabel ('\theta')
title('\theta vs \epsilon')

figure(3), plot(z,Ca), grid
xlabel ('z [mm]'), ylabel ('Ca [mol/m^{3}]')
title('Ca vs Z')

figure(4), plot(z,T), grid
xlabel ('z [mm]'), ylabel ('T [K]')
title('T vs Z')
% fin del archivo catalitico2.m


% inicio del archivo cf2.m
% ------------------------------------------------------
function [rescf]=cf2(y0,yL)
% ------------------------------------------------------

% rutina para la declaracion de las condiciones de frontera
% expresado en forma residual
% - entrada       y0: condiciones de frontera en extremo inicial
%                 yL: condiciones de frontera en extremo final
% - salida        rescf: residual de las condiciones de frontera
%                   y0 - cf1
%                   yL - cf2
%

C = 1;       Teta  = 1;      % condicion de frontera E=1
dCdE0 = 0;   dTetadE0 = 0;   % condicion de frontera E=0
```

```
%calculo de los residuales para las condiciones de frontera
rescf(1)=y0(2) - dCdE0;
rescf(2)=y0(4) - dTetadE0 ;
rescf(3)=yL(1) - C;
rescf(4)=yL(3) - Teta;
rescf=rescf';
% fin del archivo cf2.m

%inicio del archivo slab2.m
% ----------------------------------------------------
function dfdE=slab2(E,f)
% ----------------------------------------------------
% rutina de declaracion del modelo diferencial
global Si dHr De Ea kt kr0 Tr rhoc R Ts Cas Rp
% reasignacion de variables
C=f(1);        dCdE=f(2);
Teta=f(3);     dTetadE=f(4);

EaR=Ea/(R*Tr);
kc=kr0*exp(EaR*(1-Tr/(Ts*Teta)));
ra=-Rp^2*kc*rhoc*Si*C;
Rgen=-ra/De;
Qgen=(-dHr)*(-ra)/kt;
% ecuaciones diferenciales ordinarias de primer orden
dfdE(1)=dCdE;       % variable phi
dfdE(2)=Rgen;       % balance de masa especie A
dfdE(3)=dTetadE;    % variable psi
dfdE(4)=Qgen;       % balance de energia
dfdE=dfdE';
%fin del archivo slab2.m
```