

1.1 ANEXO SCRIPT PRINCIPAL

```
%% ----- DATOS DE EQUILIBRIO -----
datos_EQ_etanol_agua_1atm_chemsep;

PMA=46.07;PMB=18.02;          %pesos moleculares

%% ----- DATOS OPERACION -----

%Alimento
F.x=0.01; F.q=1;
F.xm=(F.x*f.PMA)/(F.x*f.PMA+(1-F.x)*f.PMB); %fraccion masica
F.pm=F.x*f.PMA+(1-F.x)*f.PMB; %peso molecular promedio
F.F=(9.235)/(3600*F.pm);          % kmol/s base de cálculo, la
produccion anual actual en una industria ronda las 21500 al año, lo q q equivale a
0,0213kmoles/ segundo o consumo          % se introduce una
mezclaa liquida azeotropica rica en acetato con metanol y agua con un 44% en metanol con un
pto de ebullicion de 62,3°C
F.T=f.Tx(F.x); % Temperatura plato alimentacion
F.den=f.denL(F.xm,F.T); %densidad plato alimentacion
F.Q=(F.F*F.pm)/(F.den);%Caudal volumetrico plato alimentacion
F.vis=f.visL(F.x,F.T);%viscosidad plato alimentacion

%Destilado
D.x=0.90;
D.T=f.Tx(D.x);

%Residuo
W.x=0.003;
W.T=f.Tx(W.x);

R1=118/94;          % dato extrado a raiz de la patente
se devuelven al plato mas alto 118 partes y se destilan 94 de metanol
%% a) BALANCE DE MATERIA GLOBAL

D.D=F.F*((F.x-W.x)/(D.x-W.x));
W.W=F.F-D.D;
fprintf('RESULTADOS \n');
fprintf('Destilado D = %.3e kmol/s \n',D.D);
fprintf('Residuo W = %.3e kmol/s \n',W.W);

%% b) RELACION DE REFLUJO MINIMA Y NUMERO MINIMO DE ETAPAS

% REFLUJO MINIMO
Rmin =reflujo_min_MT_est(F,D,f);
fprintf('Reflujo mínimo \tRmin = %.3f \n',Rmin);

NETmin =etapas_min_MT_est(1,F,D,W,f);
fprintf('Número mínimo de etapas, NETmin = %d\n',NETmin);
```

```

%% c) NUMERO DE ETAPAS ideales (No es necesario)

% R=[1.847*Rmin]; % el 1,847 sale de diividir el reflujo que
tenemos como dato entre el Rmin, obteniendo asi la r=1,847, de R=r^Rmin
% for i=1:length(R)
%   r=R(i);
%   [NET(i) NF(i) x y] =etapas_MT_est(1,r,F,W,D,f);
%   xW(i)=x(end);
%   NP=NET-1; %El numemro de platos sera el de etapas menos
1
% end

%% c) NUMERO DE ETAPAS REALES

P=101325;
%
% R=[1.001 1.005 1.007 1.009 1.05 1.10 1.15 1.20 1.25 1.45 1.65 1.847 2.6
% 3.34 3.50 4.09 5 6];
R=[1.1 2 3.5];
R=R*Rmin;
for i=1:length(R)

r=R(i);
[NET(i) NF(i) x y xr(i) yr(i)] =etapas_MT_EML(1,r,F,W,D,f);
xW(i)=x(end);
NP(i)=NET(i)-1;%El numemro de platos sera el de etapas menos 1
N(i)=NET(i);

end

fprintf('Número real de etapas, NET = %d\n',NET);

figure()
hold on %como varia el QC con el reflujo
plot(R, NET, '-r') %a mas reflujo mas calor
xlabel('R');ylabel('Nº Etapas reales');

for i=1:length(R)
%% POTENCIAS O CARGAS TERMICAS
% condiciones condensador

%Para poder determinar las cargas termicas primero debemos determinar las
%entalpias de las corrientes del sistema
D.h=f.hL(D.x,f.Tx(D.x)); % entapia del liquido despues de salir del condensador
HG1=f.hV(D.x,f.Ty(D.x)); %Entalpia de la corrinte que entra en el condensador
total,

```

```

%esta se encuentra toda en estado vapor por lo que su composicion sera la misma
%que la del destilado en estado liquido
QC=D.D*(R+1)*(HG1-D.h);           %Carga termica del condensador kW

% Caracteristicas del agua fria
T_salida=(35+273.15);T_entrada=(25+273.15);
den_salida=994.08;den_entrada=998.29;   % densidad del agua, kg/m3
den_prom=(den_salida+den_entrada)/2;

cp_agua=75.4;           %capacidad calorifica del agua kJ/kmol·K

%% condiciones calderin
% Caracteristicas del vapor vivo
% El vapor vivo de calefacci3n a 2.5 bar abs tiene un calor de vaporizaci3n
W.h=f.hL(W.x,f.Tx(W.x));
F.h=f.hL(F.x,f.Tx(F.x));

% realizando un balance de energia en el sistema sin fugas.
QB=QC+D.D*D.h+W.W*W.h-F.F*F.h; %Carga termica del calder3n obtenida por el balance
de materia total

lambda_vapor=2693*18;   % kJ/kmol
mvapor=QB/(lambda_vapor); % kmol/s

end

%% Representacion grafica potencia

% potencia conjunta de calder3n y condensador

QT=QB+QC;           %kj/s o kW

figure()
hold on           %como varia el QC con el reflujo
plot(R, QC, '-r')           %a mas reflujo mas calor
plot(R, QB, '-g')
xlabel('R');ylabel('Cargas termicas');
legend('Q_c_o_n_d_e_n_s_a_d_o_r','Q_c_a_l_d_e_r_i_n');

% Representaciones gr3ficas Cargas del calderin y condensador

figure()
hold on
bar(R,NET,'b');
xlabel('R');ylabel('N_e_t_a_p_a_s');
figure()
hold on
bar(R,QT,'b');
bar(R,QB,'y');
bar(R,QC,'r');
xlabel('R');ylabel('Q, kW');
legend('QT','Q_c_a_l_d_e_r_i_n','Q_c_o_n_d_e_n_s_a_d_o_r');

```

```

plot(R,QB,'sk');plot(R,QC,'sk');
plot(R,QT,'sk');
for i=1:length(R)
rendQ(i)=(QT(i)-QT(2))/QT(2);
rendET(i)= (NET(i)-NET(2))/NETmin;
end

%% Para acabar pasaremos las unidades de los flujos a kg/h en el caso del
% vapor de agua para el calderin y m3/s para el agua refrigerante en el caso del condensador , y
comentaremos un poco las graficas obtenidas

%Flujo de agua refrigerante
magua=QC/(cp_agua*(T_salida-T_entrada)); %kmol/s
fprintf('Flujo de agua refrigerante= %.4f m3/s\n',magua*18/den_prom);

for i=1:length(R)
fprintf('Flujo de vapor de agua para el calderin, mvapor%d= %.2f kg/h\n',i,mvapor(i)*18*3600);
%pasamos el flujo de vapor de kmol/s a kg/h
end
texto=sprintf('Potencia del condensador QC= %.2e kW\n',QC);
disp(texto);

% como varia el reflujo con el numero de etapas, aqui se ve claramente como
% al aumentar el reflujo hasta el infinito, la linea de operacion se acerca
% cada vez mas a la linea o diagonal, , es
% decir a mayor R, el numero de etapas disminuye , hasta llegar a Netmin

%Contrariamente a menor reflujo, llega el punto en el que llegamos a lo que
%conocemos como rmin, en este caso la linea de operacion corta la
%interseccion de la linea de alimentacion con la de equilibrio lo que
%llevaria a tener etapas infinitas
% hold on
% plot(R./(1+R),NET./(1+NET),'-s');
% xlabel('R');ylabel('N_e_t_a_p_a_s');
figure() % como varia el reflujo en funcion de los caudales, como vemos la variacion es muy
parecida tanto para el calderin como para el condensador
hold on
plot(R./(1+R),QC,'-r');
plot(R./(1+R),QB,'ob');
ylabel('Carga termica');xlabel('Variacion R ');

% close all %descomentar para ver graficas de costes
for i=1:length(R)

%% -----Diametro platos perforados y costes -----

DPE.delta_P=0.100; % perdida de carga por plato, m
DPE.dO=3; % Estimar diámetro orificio en mm
DPE.t=0.5; % Estimar espaciado entre platos,m 0.5 D>1 m 0.6
1<D<3 0.75 D>3
DPE.rWT=0.7; % Relación del derramadero r=W/DT

```

```

% DPS.n=DPE.n; % número del plato
T=f.Tx(xr(i));
% Temperaturas en funcion del reflujo para el ultimo plato
% Se supone que la alimentacion esta en su punto de burbuja, y que esta
% saturado, no se requiere condensacion para calentar la alimentacion de
% forma que V = V y L=F+L0 ver pag 562 y 565
L0.L0=D.D.*R(i);
V(i)=(D.D.*R(i))+D.D;
L(i)=F.F+(D.D.*R(i));

[DPS(i),err]=calcular_diametro_plato(P,T,V(i),yr(i),L(i),xr(i),DPE,f); % xr(i) y yr(i) es la
composicion en el ultimo plato a cada reflujo

L0.x=D.x; % reflujo, dado que estamos
realizando el ejercicio por el metodo de McCabe thiele, consideremos que la fraccion de destilado
es la misma que la del reflujo
L0.xm=(L0.x*f.PMA)/(L0.x*f.PMA+(1-L0.x)*f.PMB);
L0.pm=L0.x*f.PMA+(1-L0.x)*f.PMB;
L0.T=f.Tx(L0.x); L0.den=f.denL(L0.xm,L0.T); L0.Q=(L0.L0*L0.pm)/(L0.den);
L0.vis=f.visL(L0.x,L0.T);

end

for i=1:length(R)
%% -----Valoracion economica -----
% %% Datos para el calculo de costes

% -----Argumentos de entrada-----

% C estructura de datos de la columna
C(i).NETR=NET(i);
C(i).altura=(C(i).NETR)*DPE.t./0.85; % 15% para
cabeza y fondos DPE.t linea 15 script diametro
altura(i)=C.altura;
C(i).alturaF=(C(i).NETR-NF(i))*DPE.t+0.075*C(i).altura; %
entrada alimentación
C(i).POTB=QB(i); % Potencia del
calderin, kW
C(i).POTC=QC(i); % Potencia del
condensador, kW
C(i).TCAL=f.Tx(W.x); % temperatura del
calderin, K
C(i).TCON=f.Tx(D.x); % temperatura del
condensador,K
%% Reflujo optimo

```

```
fprintf('Diámetro del plato = %.3f m \n ',DPS.DC);  
fprintf('Altura de la columna = %.3f m \n ',C.altura);
```

```
end
```

1.2 ANEXO DATOS DE EQUILIBRIO

```
clc  
clear all  
close all  
P=101325; %Presion (Pa)  
f.PMB=18.02; f.PMA=46.07;
```

```
Txy=[373.5920 0.0000 0.0000  
369.0670 0.0496 0.3335  
365.8930 0.0964 0.4827  
363.5770 0.1404 0.5662  
361.8370 0.1820 0.6191  
360.4940 0.2214 0.6555  
359.4380 0.2587 0.6819  
358.5890 0.2941 0.7021  
357.8970 0.3277 0.7179  
357.3220 0.3597 0.7308  
356.8370 0.3902 0.7416  
356.4220 0.4192 0.7507  
356.0610 0.4470 0.7587  
355.7440 0.4735 0.7659  
355.4600 0.4988 0.7723  
355.2040 0.5231 0.7783  
354.9700 0.5464 0.7839  
354.7530 0.5687 0.7892  
354.5510 0.5901 0.7943  
354.3610 0.6107 0.7992  
354.1820 0.6305 0.8041  
354.0120 0.6495 0.8089  
353.8490 0.6679 0.8137  
353.6930 0.6856 0.8185  
353.5440 0.7026 0.8233  
353.4010 0.7191 0.8282  
353.2630 0.7349 0.8332  
353.1320 0.7503 0.8382  
353.0060 0.7651 0.8434  
352.8850 0.7795 0.8487  
352.7710 0.7934 0.8540  
352.6630 0.8068 0.8596  
352.5620 0.8198 0.8652
```

```

352.4670      0.8324  0.8711
352.3800      0.8447  0.8770
352.3010      0.8566  0.8832
352.2290      0.8681  0.8895
352.1660      0.8793  0.8960
352.1130      0.8902  0.9027
352.0690      0.9007  0.9096
352.0350      0.9110  0.9166
352.0130      0.9210  0.9239
352.0020      0.9307  0.9314
352.0040      0.9402  0.9392
352.0180      0.9494  0.9471
352.0470      0.9584  0.9553
352.0910      0.9671  0.9637
352.1510      0.9757  0.9724
352.2280      0.9840  0.9813
352.3230      0.9921  0.9905
352.4370      1.0000  1.0000

```

```
];
```

```
TEQ=Txy(:,1);xEQ=Txy(:,2);yEQ=Txy(:,3);
```

```
% crear funciones de equilibrio
```

```
f.y=@(x) interp1(xEQ,yEQ,x,'PCHIP');
```

```
f.x=@(y) interp1(yEQ,xEQ,y,'PCHIP');
```

```
f.Tx=@(x) interp1(xEQ,TEQ,x,'PCHIP');
```

```
f.Ty=@(y) interp1(yEQ,TEQ,y,'PCHIP');
```

```
% cálculo de H-1-x-Y
```

```
% Datos simplificados desde Chemsep
```

```
Tref=351;
```

```
TbB=383.79; TbA=353.24;
```

```
% capacidades calorificas kJ/kmol/K
```

```
f.cVB=@ (T) 124.38+(124.38-134.849) / (TbA-TbB)*(T-TbA);
```

```
f.cLB=@ (T) 173.168+ (173.168-183.936)/ (TbA-TbB)*(T-TbA);
```

```
f.cVA=@ (T) 99.498+ (99.498 -108.415)/ (TbA-TbB)*(T-TbA);
```

```
f.cLA=@ (T) 149.473+ (149.473-157.273) / (TbA-TbB)*(T-TbA);
```

```
% calores latentes de vaporización kJ/kmol
```

```
lambdaA=3.5112e4;
```

```
lambdaB=3.9453e4;
```

```
% datos calor de mezcla a Tref, kJ/mol no procede
```

```
fhD=@ (x) 0;
```

```
% funciones de entalpia
```

```
% h=0 Metanol Liquido puro a la temperatura Tref
```

```
f.hL=@ (x,T) (f.cLA (T).*x+f.cLB (T).*(1-x)).*(T-Tref);
```

```
f.hV=@ (y,T) lambdaA*y+lambdaB*(1-y)+(f.cVA(T).*y+f.cVB(T).*(1-y)).*(T-Tref);
```

```
H=figure();
```

```
hold on
```

```
grid on
```

```
line([0 1],[0 1],'color','k');
```

```
plot(xEQ,yEQ,'r');
```

```
x=linspace(0,1,50);
```

```
y=f.y(x);
```

```

T=f.Tx(x);
plot(x,y);

savefig(H,'datos_EQ_etanol_agua_1atm_chemsep');
%%Propiedades de transporte

% ETANOL

f.visVA=@(T)1.2467E-07*power(T,0.7862) ./ (1 + 76.034./T + -2017.3./power(T,2));
f.visLA=@(T) exp(8.061+774.76./T+-3.0701.*log(T)+-4.3408E-09.*power(T,2));
f.denLA=@(T) (1.3539./ 0.24957.^(1 + (1 - T/515.66).^0.22099))*f.PMA; % Eq 105
f.sigLA=@(T) -0.078773+ exp( 19.874./T + -2.3731+0.00067227.*T + -0.0000021132.*T.^2 );

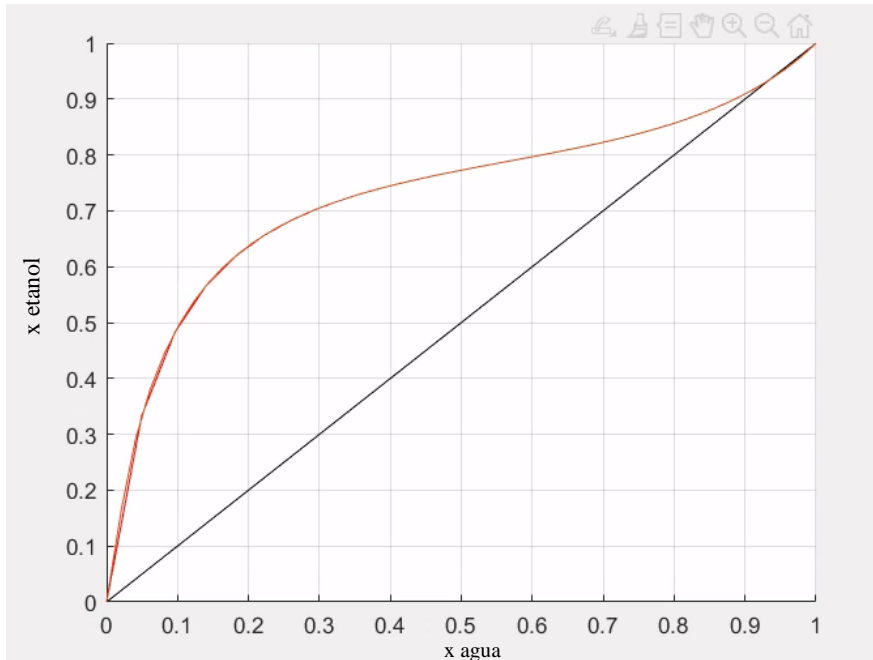
%AGUA

% f.visVB=@(T) 0.0000034514*power(T,0.35236) ./ (1 + 732.29./T + -3982.4./power(T,2));
% f.visLB=@(T) exp(16.428+-219.29./T+-4.124.*log(T)+6.0568E-07.*power(T,2));
f.denLB=@(T) (0.78926./ 0.24295.^(1 + (1 - T/523.26).^0.25762))*f.PMB; % Eq 105
% f.sigLB=@(T) -0.025331+ exp( 12.715./T + -2.6443 + -0.0005937.*T + -
0.0000027669.*T.^2 );
f.visVB=@(T) 7.002327E-08*power(T,0.934576) ./ (1 + 195.6338./T -13045.99./power(T,2));
f.visLB=@(T) exp(-133.7+6785.7./T+18.47.*log(T)+-0.000014736.*power(T,2));
% f.denLB=@(T) (32.51621./ -3.213004.^(1 + (1 - T/7.92411).^7.359898))*f.PMA;
f.sigLB=@(T) -0.031819+ exp( 167.09./T + -3.6781+ 0.0053717.*T + -0.0000084188.*T.^2 );

% MEZCLAS
% Difusividad en fase gas. Método de Fuller
% A-C2H6O B-H2O
vA=2*16.5+6*1.98+5.48;
vB=2*1.98+5.48;
f.DABV=@(p,T)(1e-
7*(T^1.75)*((1/f.PMA+1/f.PMB)^0.5)/P/1.01325e5)/((vA^(1/3)+vB^(1/3))^2);

% densidad de la mezcla
f.denL=@(xm,T) 1./(xm./f.denLA(T)+(1-xm)./f.denLB(T));
% viscosidad de la mezcla
f.visV=@(y,T) f.visVA(T). *y+f.visVB(T).*(1-y);
f.visL=@(x,T) f.visLA(T). *x+f.visLB(T).*(1-x);
% difusividad en fase líquida. Método Wilke-Chang
fi=1;
vAL=2*14.8+6*3.7+11;
DABL_298=(7.4e-12*(298.15)*((fi*f.PMB)^0.5))/(f.visLB(298.15)/1000*(vAL^0.6));
f.DABL=@(T) 7.4e-12*(T)*((fi*f.PMB)^0.5)/(f.visLB(T)/1000*(vAL^0.6));
% Tensión superficial del liquido
f.sigL=@(x,T) x.*f.sigLA(T)+(1-x). *f.sigLB(T);

```

Gráfica 1. Diagrama de equilibrio etanol-agua

1.3 ANEXO Rmin

```
function Rmin = reflujo_min_MT_est(F,D,f)
xmin=fzero(@(x) (F.q-1)*f.y(x)-F.q*x+F.x,F.x);
ordenada=D.x-((D.x-f.y(xmin))/(D.x-xmin))*D.x;
Rmin=D.x/ordenada-1;
```

end

1.4 ANEXO NÚMERO DE ETAPAS MÍNIMAS

```
function NET = etapas_min_MT_est(sg,F,D,W,f)
i=1;
y(i)=D.x;
x(i)=f.x(y(i));
while(x(i)>W.x)
    i=i+1;
    y(i)=x(i-1);
    x(i)=f.x(y(i));
end
NET=i;
```

sg=0;

if sg>0

% Salida gráfica de la operación de destilación continua por etapas de
% equilibrio utilizando el método de McCabe Thiele

```

figure()
z=linspace(0,1,50);
hold on
%ponemos las etiquetas principales del gráfico x-y
texto=sprintf('Número mínimo de etapas ideales= %d',NET);
ylabel('y, fracción molar');xlabel('x, fracción molar');
plot(z,f.y(z))

%dibujamos la diagonal
line([0 1],[0 1],'color','r');

%puntos clave
plot(D.x,D.x, '.');
plot(W.x,W.x, '.');
plot(F.x,F.x, '.');
i=1;
line([D.x x(i)],[D.x y(i)];plot(x(i),y(i),'o');
for i=2:NET
line([x(i-1) x(i)],[y(i-1) y(i)],'LineStyle','--','Color','k');
line([x(i-1) x(i)],[y(i) y(i)],'LineStyle','--');plot(x(i),y(i),'o');
plot(x(i),y(i),'o');
end
end
end
end

```

1.5 ANEXO NÚMERO DE ETAPAS IDEALES

```

function [NET NF x y] =etapas_MT_est(sg,R,F,W,D,f)
xint=fzero(@(x) ((F.q*x-F.x)*(R+1)-(R*x+D.x)*(F.q-1)),F.x);
yint=(R*xint+D.x)/(1+R);
% definimos la linea de operacion de agotamiento con su pendiente
pendW=(yint-W.x)/(xint-W.x);

i=1;
y(i)=D.x; % valor mas alto, coincide con la fraccion del destilado
x(i)=f.x(y(i));
while(x(i)>W.x)
i=i+1;
if x(i-1)>xint
y(i)=R*x(i-1)/(R+1)+D.x./(R+1); % Zona de enriquecimiento
NF=i;
else
y(i)=W.x+pendW*(x(i-1)-W.x); % Zona de agotamiento
end

x(i)=f.x(y(i));
end
NET=i;

% sg=1;

if sg>0

```

%Salida gráfica de la operación de destilación continua por etapas de
%equilibrio utilizando el método de McCabe Thiele

```

figure()
z=linspace(0,1,50);
hold on
%ponemos las etiquetas principales del gráfico x-y
texto=sprintf('Número de etapas ideales= %d',NET);
ylabel('y, fracción molar');xlabel('x, fracción molar');
plot(z,f.y(z));

%dibujamos la diagonal
line([0 1],[0 1],'color','r');

%puntos clave
plot(D.x,D.x, '!');
plot(W.x,W.x, '!');
plot(F.x,F.x, '!');
%representacion de las lineas de operación
ordenada0=D.x/(R+1);
line([D.x 0],[D.x ordenada0],'Color','k');
ycambio=xint*R/(R+1)+ordenada0;
line([xint W.x],[yint W.x],'Color','k');
%representacion de la linea de alimentación
line([xint F.x],[yint F.x],'Color','k');

i=1;
line([D.x x(i)],[D.x y(i)]);plot(x(i),y(i),'!');
for i=2:NET
line([x(i-1) x(i-1)],[y(i-1) y(i)],'LineStyle','--','Color','m');
line([x(i-1) x(i)],[y(i) y(i)],'LineStyle','--','Color','m');
plot(x(i),y(i),'!');
end
end

end

```

1.6 ANEXO NÚMERO DE ETAPAS REALES

```

function [NET NF x y xr yr] =etapas_MT_EML(sg,R,F,W,D,f)
xint=fzero(@(x) ((F.q*x-F.x)*(R+1)-(R*x+D.x)*(F.q-1)),F.x);
yint=(R*xint+D.x)/(1+R);
% definimos la linea de operacion de agotamiento con su pendiente
pendW=(yint-W.x)/(xint-W.x);

EML=0.7; %Eficacia de murphree basada en la corriente liquida
i=1;
y(i)=D.x; % valor mas alto, coincide con la fraccion del destilado
xeq(i)=f.x(y(i));
x(i)=D.x-EML*(D.x-xeq(i));

```

```

while(x(i)>W.x)
    i=i+1;
    if x(i-1)>xint
        y(i)=R*x(i-1)/(R+1)+D.x./(R+1); % Zona de enriquecimiento
        xeq(i)=f.x(y(i));
        x(i)=x(i-1)-EML*(x(i-1)-xeq(i));

        NF=i;
    else
        y(i)=W.x+pendW*(x(i-1)-W.x); % Zona de agotamiento
        xeq(i)=f.x(y(i));
        x(i)=x(i-1)-EML*(x(i-1)-xeq(i));
    end

% x(i)=f.x(y(i));
end
NET=i;
xr=x(i-1);
yr=y(i-1);

% sg=1;

if sg>0

%Salida gráfica de la operación de destilación continua por etapas de
%equilibrio utilizando el método de McCabe Thiele

figure()
z=linspace(0,1,50);
hold on
%ponemos las etiquetas principales del gráfico x-y
texto=sprintf('Número de etapas ideales= %d',NET);
ylabel('y, fracción molar');xlabel('x, fracción molar');
plot(z,f.y(z));

%dibujamos la diagonal
line([0 1],[0 1],'color','r');

%puntos clave
plot(D.x,D.x, '!');
plot(W.x,W.x, '!');
plot(F.x,F.x, '!');
%representacion de las lineas de operación
ordenada0=D.x/(R+1);
line([D.x 0],[D.x ordenada0],'Color','k');
ycambio=xint*R/(R+1)+ordenada0;
line([xint W.x],[yint W.x],'Color','k');
%representacion de la línea de alimentación
line([xint F.x],[yint F.x],'Color','k');

i=1;
line([D.x x(i)],[D.x y(i)]);plot(x(i),y(i),'!');
for i=2:NET

```

```

line([x(i-1) x(i-1)],[y(i-1) y(i)],'LineStyle','--','Color','m');
line([x(i-1) x(i)],[y(i) y(i)],'LineStyle','--','Color','m');
plot(x(i),y(i),'');
end
end

```

```
end
```

1.7 ANEXO CÁLCULO DE DIÁMETRO DE PLATOS

```
function [DPS,err]=calcular_diametro_plato(P,T,V,y,L,x,DPE,f)
```

```
% Archivo para el diseño de platos perforados.
```

```
% Argumentos de entrada
```

```
% P, presión, Pa
```

```
% T, temperatura, K
```

```
% V, flujo molar de vapor, kmol/s
```

```
% y, fracción molar del vapor
```

```
% L, flujo molar de líquido, kmol/s
```

```
% x, fracción molar del líquido x?
```

```
% DPE, estructura de datos estimados del plato
```

```
DPE.delta_P=0.100; % pérdida de carga por plato, m
```

```
DPE.dO=3; % Estimar diámetro orificio en mm
```

```
DPE.t=0.5; % Estimar espaciado entre platos, m 0.5 D>1 m 0.6 1<D<3 0.75 D>3
```

```
DPE.rWT=0.7; % Relación del derramadero r=W/DT
```

```
% f, estructura de datos y funciones de equilibrio y propiedades de
```

```
% transporte
```

```
% Argumentos de salida
```

```
% DPS, estructura de dimensiones del plato
```

```
% err, vector de códigos de los tres posibles errores
```

```
%% PROPIEDADES -----
```

```
% Argumentos entrada
```

```
% Propiedades de los fluidos:
```

```
MV=y*f.PMA+(1-y)*f.PMB; % peso molecular promedio del gas
```

```
Vm=V.*MV; % flujo másico del gas
```

```
den_V=P.*MV./(8314*T); % densidad del gas (kg/m3)
```

```
ML=y*f.PMA+(1-y)*f.PMB; % peso molecular promedio del líquido
```

```
Lm=L.*ML; % flujo másico del gas (kg/s)
```

```

xm=y*f.PMA./(y*f.PMA+(1-y)*f.PMB);

vis_V= f.visV(y,T);      % viscosidad del vapor, kg/m s
vis_L=f.visL(x,T);      % viscosidad del líquido, kg/m·s
den_L= f.denL(xm,T);    % densidad del líquido, kg/m3
sig_L= f.sigL(x,T);     % tensión superficial del líquido,P·m

%% DISEÑO DEL PLATO -----
err=[0 0 0];
%% CALCULO DIAMETRO. CONDICIONES DE INUNDACION
% dO, diámetro de orificio (m)
p=3*DPE.dO;      % distancia entre centros de perforaciones,mm
                % 2.5 a 5 diámetros de los orificios
rLV=(Lm./Vm).*power(den_V./den_L,0.5);
AOAA=0.907.*power(DPE.dO/p,2); % relación de área de perforaciones a área activa

% Cálculo de los parámetros del coeficiente de inundación

if rLV<0.1 rLV=0.1; end
if AOAA>=0.1
    alfa=0.0744*DPE.t+0.01173;
    beta=0.0304*DPE.t+0.015;
else alfa=(0.0744*DPE.t+0.01173)*(5*AOAA+0.5);
     beta=(0.0304*DPE.t+0.015)*(5*AOAA+0.5);
end

CF=(alfa.*log10(1./rLV)+beta).*(sig_L/0.02).^0.2;% coeficiente de inundación
% velocidad superficial del líquido en condiciones de inundación
vGF=CF.*power((den_L-den_V)./den_V,0.5);

% Diámetro de la columna
% El factor principal que determina el diámetro de la columna es el caudal
% de vapor. La velocidad del vapor debe estar por debajo de la que causaría
% un arrastre excesivo de líquido o una caída de presión alta.

vG=0.85*vGF;      % velocidad del gas operativa suponiendo el 85%
AN=Vm./(den_V.*vG); % Area para el paso del gas

% Relación del derramadero reW=W/DT y area ocupada por el derramadero
% a partir de la relación de la distancia del derramadero al centro, rWT*DT,
% igual a 0.7;
[reW pArea]=relacion_derramadero(DPE.rWT);
AC=AN/(1-pArea); % Area de la columna (m)
DC=sqrt(4*AC/pi); % Diametro de la torre

% Diametro de la torre corregido utilizando tubos normalizados varían de
% 152 mm a 3000 mm (6" a 120").

for i=6:6:236
    % Aproximación al diametro standar hasta 236 in (6000 mm)
    if DC*100/2.54<=i
        DC=i*2.54/100; break
    end
end
%% Diseño provisional

```

% Comprobar el lloriqueo (check weeping)

LW=DC*DPE.rWT; % Longitud del derramadero
AC=pi*DC^2/4; % área de columna corregida, m2
AD=AC*pArea; % area del derramadero

% zonas no perforadas para soporte y distribución de líquido
% diseño de acuerdo con el esquema de la figura 6.14 Treybal

e_soportes=40e-3; % espacio para de soportes circular(m)
e_distribucion=50e-3; % espacio para distribución (m)

% Area soportes y zona de distribución:

LD=(asin(DPE.rWT)*2)*DC/2; % Longitud de la cuerda del vertedero
Area_sop_dis=(2*LW+DC+2*reW*DC)*e_distribucion+pi*DC*e_soportes;
% restando los solapes
Area_sop_dis=Area_sop_dis-3*e_distribucion*e_distribucion...
-6*e_distribucion*e_soportes-2*LD*e_soportes;

AA=AC-2*AD-Area_sop_dis; % área activa
AO=AOAA/AA; % Area de orificios

%% ANALISIS DEL FLUJO VOLUMETRICO ESPECIFICO (q/W)

% Se ha de cumplir que $q/W < 0.015 \text{ m}^3/(\text{m}*\text{seg})$

% fprintf('Diámetro= %.4f\n',DT);

q=Lm./den_L;

if q/LW < 0.015

% fprintf('Se cumple la condición. $q/W < 0.015$ \n');

else

% fprintf('Se debe recalcular un valor mayor de LW\n');

err(1)=1;

end

%% Altura del derramadero hW.

% Se recomienda un valor de h_1+h_w comprendido entre 0.1 m y 0.05 m

hW=50; % mm líquido

% Cresta líquida sobre el vertedero, hOW

hOW=750*power(Lm./(den_L*LW),2/3); % mm líquido

% con la velocidad de líquido al 70%

hOW_min=750*power(Lm*0.7./(den_L*LW),2/3); % mm líquido

% Tasa de lloriqueo

hLL=hW+hOW_min;

K2=-9.90E-08*(hLL).^4+2.98E-05*(hLL).^3-3.49E-03*(hLL).^2+2.06E-01*(hLL)+2.54E+01;

uOmin=(K2-0.9*(25.4-DPE.dO))./(den_V.^0.5);

uOa=0.7*Vm./(den_V.*AO);

uO=uOa;

if uOa < uOmin

AO=AO*uOa./(1.1*uOmin);

uO=Vm./(den_V*AO);

end

%% CALCULO DE LA PERDIDA DE CARGA GLOBAL

% Caída de presión en seco, hD

```

ahap=AO/AA;
% se toma una relación espesor del plato lp/dO de 0.6
lp=0.6*DPE.dO;
k0=0.7+((0.7-0.78)/(5-15))*(ahap*100-5);
hD=(51*power(uO./k0,2).*den_V./den_L);

% Caída de presión residual, hR
hR=(12.5E3./den_L);

% Caída de presión total
ht=hW+hOW+hD+hR;

% Caída de presión derramadero

p_V=(hW-10)/1000; % Paso para el líquido en el vertedero, (m)
ADA=p_V*LW; % Area del faldon del vertedero (m2)
Am=min(ADA,AD);

hDC=166*power(q/Am,2);
% Retroceso del vertedero,
hB=hW+hOW+ht+hDC;

%

%% ANALISIS DE LAS PERDIDAS DE PRESION GLOBALES
% Se ha de cumplir que La suma de caidas de presión
% debe ser inferior al 50% de la separación entre platos t

if hB <=(DPE.t*1000+hW)/2
    % fprintf('Se cumple la condición. Suma_h=h1+h3+hW <t/2\n');
else
    % fprintf('No se cumple la condición. Se debe recalcular un valor de t mayor de %.3f\n',t);
    err(2)=2;
end

% Tiempo de residencia del liquido
tR=(AD*hB/1000.*den_L)/Lm;

if tR<3
    % fprintf('Tiempo de residencia del liquido inferior a 3 s\n');
    err(3)=3;
end

%% Dimensiones del plato
DPS.DC=DC;DPS.LW=LW; % diámetro del plato, m
DPS.hW=hW; % altura del derramadero, m
DPS.p=p; % distancia entre centros de perforaciones,mm
DPS.lp=lp; % espesor del plato, mm
DPS.AA=AA; % area activa, m2
DPS.AO=AO; % area orificios, m2
DPS.AAAC=AA/AC; % relación área activa/área de columna corregida
DPS.PMAX=max(P); % presión máxima, Pa
DPS.ht=max(ht)/1000; % caída de presión total, m

```


DPS.VP=V; DPS.LP=L; % flujos de vapor y líquido, kmol/s
%DPS.n=DPE.n; % número del plato
End