



TECNOLÓGICO NACIONAL DE MÉXICO EN CELAYA



INGENIERÍA QUÍMICA

SIMULACIÓN DE PROCESOS

PROBLEMAS 1º PARCIAL

Presenta:

JIMENEZ RAMOS DANIELA ARELI
PATIÑO MANCERA ANA MARIA
RAMIREZ CHAVEZ MARIA ALEJANDRA

Profesor:

DR. LUIS FABIÁN FUENTES CORTÉS

Fecha de entrega: 12 de abril de 2021

Problemas Primer Parcial

Utilice Julia para resolver los siguientes problemas. Genere el modelo para todos los problemas que se requiera. Considere que varios de ellos llevan un método numérico asociado para su resolución. Incluya el código y un análisis ingenieril comentando sobre el comportamiento de las variables con las que está trabajando y los principios físicos asociados a cada fenómeno. Incluya las gráficas correspondientes en los problemas que lo requieran. No olvide incluir los resultados y su análisis.

1. Se sabe que el aire frío se siente mucho más frío cuando hace viento, que lo que indica el termómetro; eso se debe al “efecto frigorífico” del viento, el cual está asociado al aumento en el coeficiente de transferencia de calor por convección al aumentar la velocidad del aire. La temperatura equivalente por enfriamiento de viento, en °F, se determina con la ecuación:

$$T_{\text{equiv}} = 91.4 - (91.4 - T_{\text{ambiente}}) \times 0.475 - 0.0203V + 0.304\sqrt{V}$$

donde V es la velocidad del viento, en mi/h, y T_{ambiente} la temperatura del aire ambiente, en °F. Se supone que el aire ambiente es inmóvil cuando los vientos son ligeros, hasta de 4 mi/h. La constante 91.4 °F en esta ecuación es la temperatura promedio de la piel de una persona en reposo, en un ambiente confortable. La temperatura equivalente con aire a T_{ambiente} , en movimiento a la velocidad V , se sentirá como si el aire estuviera a la temperatura T_{equiv} . Aplique los factores de conversión adecuados para obtener una ecuación equivalente en unidades SI, donde V sea la velocidad del viento, en km/h, y T_{ambiente} sea la temperatura del aire ambiente en °C. Grafique las temperaturas equivalentes por enfriamiento de viento, en °F y °C, en función de la velocidad del viento, entre los límites de 4 a 40 mi/h y sus correspondientes en km/h, para temperaturas ambiente de 20, 40 y 60 °F. Describa los resultados.

Código en Julia

```
clearconsole()
using PyPlot

d=3
TenF=zeros(d)
TenC=zeros(d)

TenF[1]=20
TenF[2]=40
TenF[3]=60
n=401

TeqF=zeros(n)
TeqC=zeros(n)
VF=zeros(n)
VC=zeros(n)
VF[1]=4

for j=1:d
    global TenC[j]=(TenF[j]-32)*5/9
    for i=1:n-1
        global VC[i]=VF[i]*1.60934
        global TeqF[i]=(91.4-(91.4-TenF[j]))*(0.0475-
0.0203*VF[i]+0.304*sqrt(VF[i]))
```

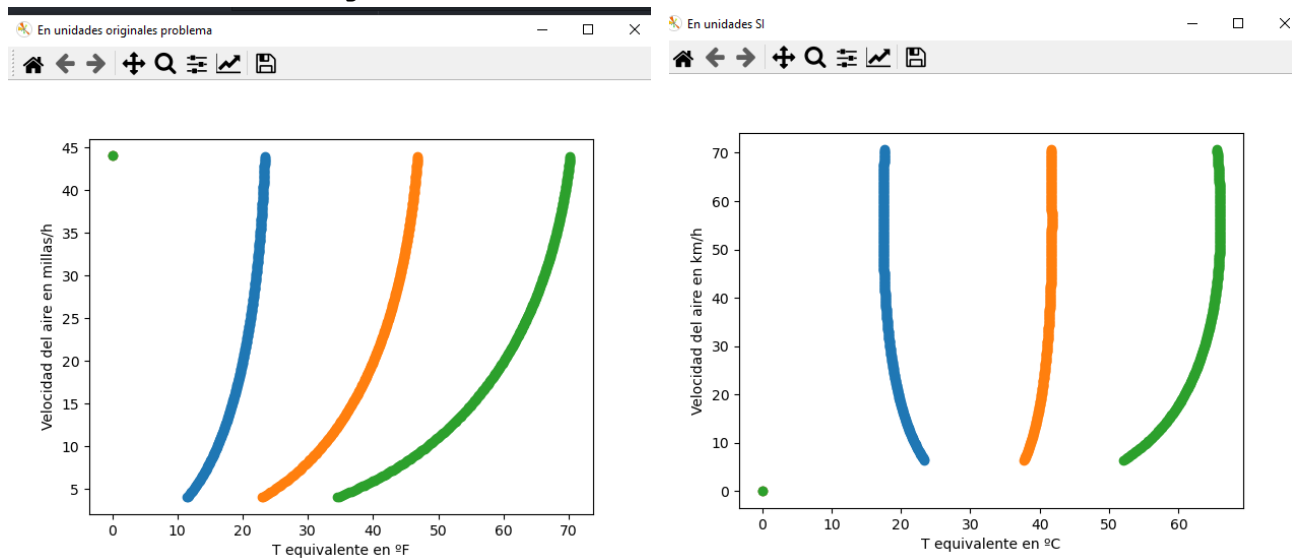
```

        global TeqC[i]=((91.4-32)*5/9-((91.4-32)*5/9-
TenC[j]))*(0.0475*1.60934-0.0203*VC[i]+0.304*sqrt(VC[i]))*9/5+32
        global VF[i+1]=VF[i]+0.1
    end
    figure("En unidades originales problema ")
    scatter(TeqF, VF)
    xlabel("T equivalente en °F")
    ylabel("Velocidad del aire en millas/h")

    figure("En unidades SI")
    scatter(TeqC, VC)
    xlabel("T equivalente en °C")
    ylabel("Velocidad del aire en km/h")
end

```

Resultados del código



Análisis de resultados

Los factores de conversión son:

$$C = \frac{5(F - 32)}{9}$$

$$1km = 1.60934 \text{ mi}$$

La ecuación equivalente en unidades de SI es:

$$T_{equiv} = 33 - (33 - T_{amb}) * (0.475 - 0.0126V + 0.240 \sqrt{V})$$

- La solución de la expresión:

$$\frac{\partial T}{\partial t} = \frac{\partial^2 T}{\partial x^2}$$

Considerando la condición inicial $T(x,0) = 1$ y las condiciones de frontera $T(0,t) = 0$ y $T(1,t) = 0$, la temperatura está dada por la serie de Fourier:

$$T(x,t) = \sum_{k=1}^{\infty} \frac{4}{(2k-1)\pi} \sin[(2k-1)\pi x] \exp[-(2k-1)^2 \pi^2 t]$$

Escriba un código que permita calcular las temperaturas para $x = 0.1, 0.25, 0.5, 0.75$ y 0.9 . Considere 100 formas diferentes de la transformada ($k=1 \rightarrow 100$). Considere un minuto de operación con t en segundos y grafique los resultados de T con tiempos $t=1 \rightarrow 60$.

Código en Julia

```
clearconsole();
```

```
T=0
x1=0.1
for t=1:60
    for k=1:100
        global T=T+(4/((2*k-1)*pi))*sind((2*k-1)*pi*x1)*exp(-1*(2*k-1)^2*pi^2*t)
        println(T)
    end
end
println("\n")
println("En un tiempo de 60 segundos la temperatura alcanzada en x = ",x1," es de: ",T)
```

```
T=0
x1=0.25
for t=1:60
    for k=1:100
        global T=T+(4/((2*k-1)*pi))*sind((2*k-1)*pi*x1)*exp(-1*(2*k-1)^2*pi^2*t)
    end
end
println("En un tiempo de 60 segundos la temperatura alcanzada en x = ",x1," es de: ",T)
```

```
T=0
x1=0.5
for t=1:60
    for k=1:100
        global T=T+(4/((2*k-1)*pi))*sind((2*k-1)*pi*x1)*exp(-1*(2*k-1)^2*pi^2*t)
    end
end
println("En un tiempo de 60 segundos la temperatura que se alcanza en x = ",x1," es de: ",T)
```

```
T=0
x1=0.75
for t=1:60
    for k=1:100
        global T=T+(4/((2*k-1)*pi))*sind((2*k-1)*pi*x1)*exp(-1*(2*k-1)^2*pi^2*t)
```

```

        end
    end
    println("En un tiempo de 60 segundos la temperatura alcanzada en x =
    ",x1," es de: ",T)

    T=0
    x1=0.9
    for t=1:60
        for k=1:100
            global T=T+(4/((2*k-1)*pi))*sind((2*k-1)*pi*x1)*exp(-1*(2*k-
            1)^2*pi^2*t)
        end
    end
    println("En un tiempo de 60 segundos la temperatura alcanzada en x =
    ",x1," es de: ",T)

```

Resultados del código

```

3.61094150190531e-7
3.61094150190531e-7
3.61094150190531e-7
3.61094150190531e-7
3.61094150190531e-7
...
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7
3.6111282809657925e-7

```

En un tiempo de 60 segundos la temperatura alcanzada en x = 0.1 es de:
3.6111282809657925e-7

En un tiempo de 60 segundos la temperatura alcanzada en x = 0.25 es de:
9.027583213740956e-7

En un tiempo de 60 segundos la temperatura que se alcanza en x = 0.5 es
de: 1.805347014101301e-6

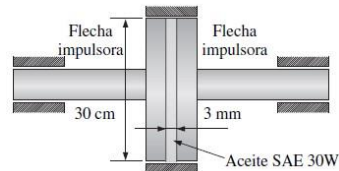
En un tiempo de 60 segundos la temperatura alcanzada en x = 0.75 es de:
2.7075964814080166e-6

En un tiempo de 60 segundos la temperatura alcanzada en x = 0.9 es de:
3.2487128042426277e-6

Análisis de resultados

Después de considerar las 100 formas diferentes de la transformada en la expresión dada se obtiene una temperatura en $x=0.1$ de $3.6111282809657925e-7$, en $x=0.25$ de $9.027583213740956e-7$, en $x=0.5$ de $1.805347014101301e-6$, en $x=0.75$ de $2.7075964814080166e-6$ y en $x = 0.9$ de $3.2487128042426277e-6$

3. El sistema de embrague que se muestra en la figura se usa para transmitir par de torsión mediante una película de aceite con $\mu = 0.38 \text{ N s/m}^2$ que está entre dos discos idénticos de 30 cm de diámetro. Cuando la flecha impulsora gira a una velocidad de 1450 rpm, se observa que la flecha impulsada gira a 1398 rpm. Suponiendo un perfil lineal de velocidad para la película de aceite, determine el par de torsión transmitido.



Con Julia, investigue el efecto del espesor de la película de aceite en el par de torsión transmitido. Haga que el espesor de la película varíe desde 0.1 mm hasta 10 mm. Trace la gráfica de los resultados que obtenga y exprese sus conclusiones.

Código en Julia

```
clearconsole()

miu=0.38
D=0.3 #m
w1=1450 #en rpm
w2=1398 #en rpm

w=2*pi*abs(w1-w2)/60 #Velocidad angular relativa

ls=10
s=0.1/1000
a=ls/s

h=zeros(100)
T=zeros(100)

for i=1:99
    global h[i+1]=h[i]+s
    global T[i+1]=pi*miu*w*(D^4)/(32*h[i+1])
end

h[1]=0.000099
T[1]=pi*miu*w*(D^4)/(32*h[1])

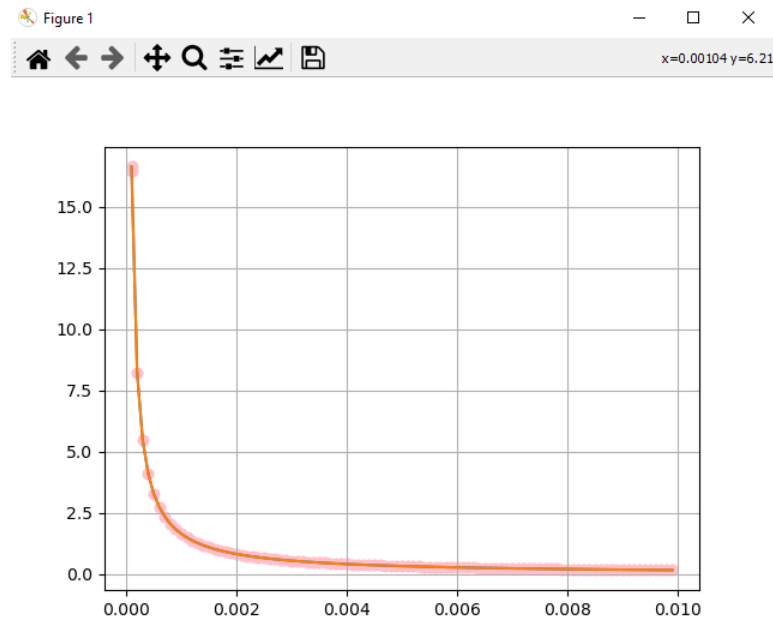
using PyPlot
figure(1)
scatter(h,T,color="pink",label="")
```

```
plot(h,T)
grid("on")
```

```
println(" h = ", h[31]," T = ", T[31]) #valor medio
```

Resultados del código

```
h = 0.0029999999999999988 T = 0.5485032645905413
```



Análisis de resultados

La viscosidad se refiere a la resistencia que poseen algunos líquidos durante su fluidez y deformación.

Buscando en la literatura encontramos que la fuerza requerida para mover una placa es directamente proporcional a la velocidad a la que se mueve y al área de dicha placa, e inversamente proporcional a la distancia que los separa.

$$F \propto \frac{A * v}{h}$$

El factor de proporcionalidad en este caso es descrito como la viscosidad, por lo que podemos hacer el siguiente desarrollo:

$$F = \mu \frac{A * v}{h} \rightarrow \frac{F}{A} = \frac{\mu * v}{h} = \tau$$

Donde τ es el esfuerzo cortante.

Para el calculo de la velocidad dadas las velocidades angulares (w_1 y w_2) se hace lo siguiente ya que rotan a diferentes velocidades:

$$v = (w_1 - w_2) * r$$

Sustituyendo v :

$$\tau = \mu \frac{dv}{dr} = \frac{\mu * v}{h} = \frac{\mu * (w_1 - w_2) * r}{h}$$

Suponiendo un esfuerzo cortante en un área diferencial donde después se debe integrar, obtuvimos lo siguiente:

$$T = \frac{\pi * \mu * (w_1 - w_2) * D^4}{32 h}$$

En el código de Julia, se vario la distancia entre las placas entre 0.1 y 10 mm, y se construyo un perfil de torsión en función de la distancia de las placas.

4. La variación de la viscosidad dinámica del agua con la temperatura absoluta se da como:

T, K	$\mu, Pa \cdot s$
273.15	1.787×10^{-3}
278.15	1.519×10^{-3}
283.15	1.307×10^{-3}
293.15	1.002×10^{-3}
303.15	7.975×10^{-4}
313.15	6.529×10^{-4}
333.15	4.665×10^{-4}
353.15	3.547×10^{-4}
373.15	2.828×10^{-4}

Con los datos de la tabla desarrolle una relación para la viscosidad en la forma de $\mu = \mu(T) = A + BT + CT^2 + DT^3 + ET^4$. Use la relación desarrollada, prediga las viscosidades dinámicas del agua a 50°C, a la cual el valor reportado en la literatura es de $5.468 \times 10^{-4} Pa \cdot s$. Compare su resultado con los de la ecuación de Andrade, la cual se da en la forma de $\mu = D \cdot e^{B/T}$, donde D y B son constantes cuyos valores se deben determinar usando los datos de la viscosidad.

Código en JULIA

```
clearconsole()

y=[1.787e-3 1.519e-3 1.307e-3 1.002e-3 7.975e-4 6.529e-4 4.665e-4
3.547e-4 2.828e-4]
x=[273.15 278.15 283.15 293.15 303.15 313.15 333.15 353.15 373.15]

m=4
n=9

xi=0
yi=0
xi2=0
xi3=0
xi4=0
xi5=0
xi6=0
xi7=0
xi8=0
xiyi=0
xi2yi=0
xi3yi=0
xi4yi=0

a0=a1=a2=a3=a4=1

for i=1:n
```



```

    global xi=xi+(x[i])
    global yi=yi+(y[i])
    global xi2=xi2+(x[i]^2)
    global xi3=xi3+(x[i]^3)
    global xi4=xi4+(x[i]^4)
    global xi5=xi5+(x[i]^5)
    global xi6=xi6+(x[i]^6)
    global xi7=xi7+(x[i]^7)
    global xi8=xi8+(x[i]^8)
    global xiyi=xiyi+(x[i]*y[i])
    global xi2yi=xi2yi+((x[i]^2)*y[i])
    global xi3yi=xi3yi+((x[i]^3)*y[i])
    global xi4yi=xi4yi+((x[i]^4)*y[i])
end

E=[n xi xi2 xi3 xi4;xi xi2 xi3 xi4 xi5;xi2 xi3 xi4 xi5 xi6;xi3 xi4 xi5
xi6 xi7;xi4 xi5 xi6 xi7 xi8]
M=[yi;xiyi;xi2yi;xi3yi;xi4yi]

z=inv(E)*M
A=0.4892920416314155
B=-0.0056890474184001505
C=2.4915226980226635e-5
D=-4.861560886151861e-8
E=3.56198345152119e-11

println("Valores de las constantes de la relación desarrollada ")
println(z)
println("Valor de A: ",A)
println("Valor de B: ",B)
println("Valor de C: ",C)
println("Valor de D: ",D)
println("Valor de E: ",E)

```

Resultados del código

```

Valores de las constantes de la relación desarrollada
[0.48929204122396186, -0.005689047416581161, 2.491522697667392e-5, -
4.8615608903151974e-8, 3.56198345152119e-11]
Valor de A: 0.4892920416314155
Valor de B: -0.0056890474184001505
Valor de C: 2.4915226980226635e-5
Valor de D: -4.861560886151861e-8
Valor de E: 3.56198345152119e-11

```

Análisis de resultados

Al tener un polinomio de cuarto orden ocupamos la regresión de polinomios porque necesitábamos conocer el valor de las constantes para obtener la viscosidad a la temperatura que nos piden.

Evaluando los valores que imprime el programa, podemos comprobar que tienen cierta variación.

$$u(T) = A + BT + CT^2 + DT^3 + ET^4$$
 Forma polinomial.

$$Y = a_0 + a_1X + a_2X^2 + a_3X^3 + a_4X^4$$

$$E_r = \sum_{i=1}^n (y_i - a_0 - a_1x - a_2x^2 - a_3x^3 - a_4x^4)^2$$

$$a_0n + a_1\sum x_i + a_2\sum x_i^2 + a_3\sum x_i^3 + a_4\sum x_i^4 = \sum y_i$$

$$a_0\sum x_i + a_1\sum x_i^2 + a_2\sum x_i^3 + a_3\sum x_i^4 + a_4\sum x_i^5 = \sum x_i y_i$$

$$a_0\sum x_i^2 + a_1\sum x_i^3 + a_2\sum x_i^4 + a_3\sum x_i^5 + a_4\sum x_i^6 = \sum x_i^2 y_i$$

$$a_0\sum x_i^3 + a_1\sum x_i^4 + a_2\sum x_i^5 + a_3\sum x_i^6 + a_4\sum x_i^7 = \sum x_i^3 y_i$$

Error regresión

$$\frac{Y}{X} = \sqrt{\frac{S_r}{n - (m+1)}}$$
 ↑
 grado del polinomio
 que queremos ajustar.

- Entra vapor estacionariamente a una turbina a 7 MPa y 600 °C con una velocidad de 60 m/s, y sale a 25 kPa con una calidad de 95 %. Durante el proceso ocurre una pérdida de calor de 20 kJ/kg. El área de entrada de la turbina es de 150 cm², y el área de salida es de 1 400 cm². Determine el flujo másico del vapor, la velocidad de salida y la potencia desarrollada. Utilice Julia para investigar los efectos del área de salida de la turbina y la presión de salida de la turbina en la velocidad de la salida y la producción de potencia de la turbina. Suponga que la presión de salida varía de 10 a 50 kPa (con la misma calidad), y el área de salida varía de 1 000 a 3 000 cm². Grafique la velocidad de salida y la potencia desarrollada contra la presión de salida para las áreas de salida de 1 000, 2 000 y 3 000 cm², y explique los resultados.

Código en JULIA

```

clearconsole()

P1=7
T=600
Vel=60
A1=0.015
P2=25
A2=0.14
x2=0.95
Qs=20
Voles=0.055820432
DenEntr=17.91458731
h1=3650.03606

```

```

m1=Vel*DenEntr*A1

println("El valor del flujo masico del vapor en kg/s es: ",m1)

Ts=64.8239
vf=0.001019822
vg=6.312686533
hf=271.1588605
hg=2616.609112
v2=((1-x2)*vf)+(x2*(vg))
DenSal=v2^-1
Vel2=m1/(DenSal*A2)

println("El valor de la velocidad de salida en m/s es: ",Vel2)

h2=((1-x2)*hf)+(x2*(hg))
Ws=(m1*h1)+(m1*((Vel^2)/(1000*2)))-(m1*h2)-(m1*((Vel2^2)/(1000*2)))-(Qs*m1)

println("El valor de la potencia desarrollada de salida en KW es: ",Ws)

using PyPlot

Pressu=[10 15 20 25 30 40 50]
vft=[0.001010 0.001014 0.001017 0.001020 0.001022 0.001026 0.001030]
vgt=[14.670 10.020 7.6481 6.2034 5.2287 3.9933 3.2403]
hft=[191.81 225.94 251.42 271.96 289.27 317.62 340.54]
hgt=[2583.9 2598.3 2608.9 2617.5 2624.6 2636.1 2645.2]
n=7
v=zeros(1,n)
D=zeros(1,n)
vel=zeros(1,n)
for i=1:n
    global A=0.1
    global v[1,i]=((1-x2)*vft[i])+(x2*(vgt[i]))
    global D[1,i]=v[1,i]^(-1)
    global vel[1,i]=m1/(A*D[1,i])
end
println(v)
println(D)
println(vel)
h=zeros(1,n)
W=zeros(1,n)
for i=1:n
    global h[1,i]=((1-x2)*hft[i])+(x2*(hgt[i]))
    global W[1,i]=(m1*h1)+(m1*((Vel^2)/(1000*2)))-(m1*h[1,i])-(m1*((vel[1,i]^2)/(1000*2)))-(Qs*m1)
end

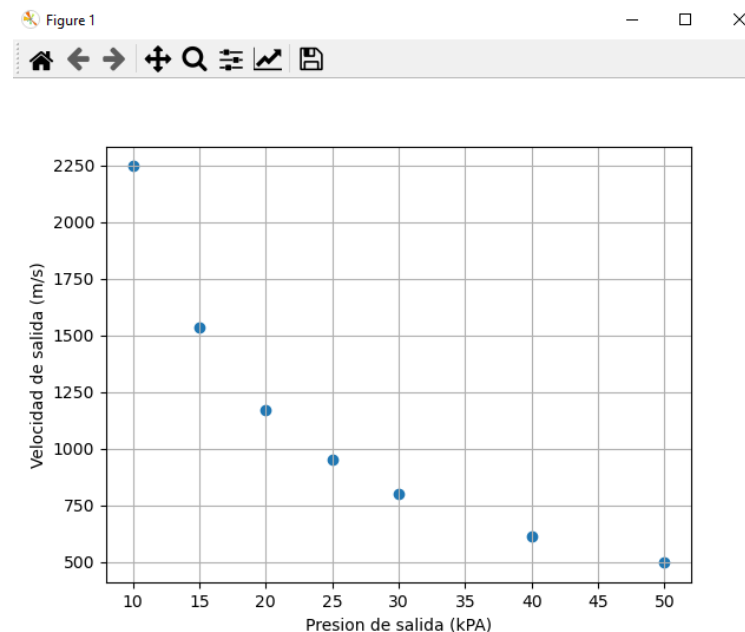
```

```
end
```

```
println(W)
figure(1)
plot(Pressu,vel)
grid("on")
scatter(Pressu, vel)
xlabel("Presion de salida (kPA)")
ylabel("Velocidad de salida (m/s)")
```

Resultados del código

El valor del flujo masico del vapor en kg/s es: 16.123128579
El valor de la velocidad de salida en m/s es: 690.657613957274
El valor de la potencia desarrollada de salida en KW es:
14414.004245867081
[13.9365505 9.519050699999998 7.26574585 5.893281 4.9673161 3.7936863
3.0783365]
[0.07175376718937732 0.1050524922616496 0.1376321193508303
0.16968476473461896 0.20131595812877703 0.2635958592569976
0.3248507757355312]
[2247.007956592267 1534.768783861199 1171.4655456188564 950.1812731517768
800.8867617283681 611.6609200329077 496.3241519892883]
[-21878.792338951695 -412.80932804188404 7330.288469751294
10966.80427606193 12951.592414851075 14907.372443695906 15779.7096175669]



Análisis de resultados

Para la solución de este problema debemos tener claro los conceptos de flujo masico de velocidades máscicas. Se cambio la presión de salida en un rango de 50 a 10 kPa, variando también el área de salida de 10 a 30 m2. Se grafico la velocidad de salida y potencia contra la presión a la salida

6. Una mezcla líquido-vapor de etilbenceno y tolueno tiene una presión parcial de 250 mm Hg de etilbenceno y 343 mm Hg de tolueno. Escriba un código que calcule la composición de la mezcla (fracciones molares líquidas y gaseosas) y la temperatura de la mezcla considerando un comportamiento ideal. La presión de saturación se puede aproximar utilizando la ecuación de Antoine, en mm Hg y Celsius:

$$\log_{10} P^{\text{sat}} = A - \frac{B}{T + C}$$

Utilizando los parámetros de la siguiente tabla:

Chemical	A	B	C
Ethylbenzene	6.95719	1424.255	213.21
Toluene	6.95464	1344.8	219.48

Código en JULIA

```
clearconsole()

# a: etilbenceno
# b: tolueno
#Datos
pa=250 #mmHg
pb=343 #mmhg
#parámetros Ley de Antoine
Aa=6.95719; Ba=1424.255; Ca=213.21
Ab=6.95464; Bb=1344.8; Cb=219.48

PT=pa+pb
println("La presión total es: ",PT, " mmHg")

ya=pa/PT
yb=pb/PT

println("La fracción molar del etilbenceno es: ",ya)
println("La fracción molar del tolueno es: ",yb)

Pa0=pa/ya
Pb0=pb/yb

println("La presión de saturación del etilbenceno es: ",Pa0)
println("La presión de saturación del tolueno es: ",Pb0)

#Para calcular la temperatura de cada compuesto podemos usar la
ecuación de Antonie despejando la T
#T=(-B-C*log(Pi0)+AC)/(log(Pi0)-A)
Ta=(-Ba-Ca*log(Pa0)+Aa*Ca)/(log(Pa0)-Aa)
Tb=(-Bb-Cb*log(Pb0)+Ab*Cb)/(log(Pb0)-Ab)
```

```
println("La Temperatura del etilbenceno es: ",Ta, " °C")
println("La Temperatura del tolueno es: ",Tb, " °C")
```

Resultados del código

```
La presión total es: 593 mmHg
La fracción molar del etilbenceno es: 0.42158516020236086
La fracción molar del tolueno es: 0.5784148397976391
La presión de saturación del etilbenceno es: 593.0
La presión de saturación del tolueno es: 593.0
La Temperatura del etilbenceno es: 2276.7654430005455 °C
La Temperatura del tolueno es: 2142.1152035331093 °C
```

Análisis de resultados

Siendo gases podemos aplicar la Ley de presión parcial de Dalton

$$P_T = \sum p_i$$

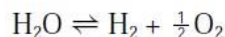
Con la ley de Raoult podemos despejar la fracción molar:

$$y_i = \frac{p_i}{p_i^s}$$

Para obtener la presión de saturación usamos la ley de Raoult.

Al tener ya la presión de saturación sustituimos en la Ley de Antonie y así podemos despejar la Temperatura que es lo que solicita el problema.

7. En un proceso el vapor de agua (H₂O) se calienta a temperaturas lo suficientemente altas para que una porción significativa del agua se disocie, o se rompa, para formar oxígeno (O₂) e hidrógeno (H₂):



Si se supone que ésta es la única reacción que se lleva a cabo, la fracción molar x de H₂O que se disocia se representa por:

$$K = \frac{x}{1-x} \sqrt{\frac{2p_t}{2+x}}$$

donde K = la constante de equilibrio de la reacción y p_t = la presión total de la mezcla. Si $p_t = 3$ atm y $K = 0.05$, determine el valor de x que satisfaga la ecuación.

Código en JULIA

```
clearconsole()
p=3
K=0.05
xL=-0.5
xU=0.5
fL=xL*sqrt((2*p)/(2+xL))/(1-xL)-K
fU=xU*sqrt((2*p)/(2+xU))/(1-xU)-K

er=100
xN=0
n=0

while er>=0.0001
    global xN=(xL+xU)/2
    global fN=xN*sqrt(2*p/(2+xN))/(1-xN)-K
```

```

a=fN*fU
b=fN*fL
if fN==0

end
if a<0 && b>0
    global xL=xN
    global fL=xL*sqrt(2*p/(2+xL))/(1-xL)-K
end
if a>0 && b<0
    global xU=xN
    global fU=xU*sqrt(2*p/(2+xU))/(1-xU)-K
end
global n=n+1
global er=abs((xU-xL))

end

println("la raíz exacta es ", xN)
println("La raíz aproximada es ", xN, " con un error de ", er)
println("Número de iteraciones: ", n)

using PyPlot
xg=zeros(100)
f=zeros(100)

acum=-0.51

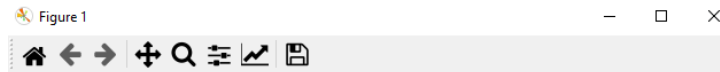
for i=1:100
    global acum=acum+0.01
    global xg[i]=acum
    global f[i]=xg[i]*sqrt(2*p/(2+xg[i]))/(1-xg[i])-K
end

figure(1)
plot(xg,f)
scatter(xN,fN)
grid("on")

```

Resultados del código

la raíz exacta es 0.02825927734375
La raíz aproximada es 0.02825927734375 con un error de 6.103515625e-5
Número de iteraciones: 14



Análisis de resultados

Para la solución de este problema se igualará la ecuación a 0, de manera que al obtener la raíz se llegue a la solución:

$$0 = \frac{x}{1-x} \sqrt{\frac{2 p_t}{2+x}} - K$$

Se buscó el valor de x correspondiente a la raíz con ayuda del método de bisección. Dadas las restricciones del dominio se seleccionó un intervalo de $[-0.5, 0.5]$, de manera que la función fuera continua.

8. La ecuación de Van der Waals para un gas real utilizando P en atm, Volumen en litros y T en K es expresada de la siguiente manera:

$$P = \frac{nRT}{V-nb} - \frac{n^2 a}{V^2}$$

n es el número de moles.

Considere que tiene 1.5 moles de nitrógeno ($a = 1.39 \text{ MPa L}^2/\text{mol}^2$ y $b = 0.03913 \text{ L/mol}$) confinados en un recipiente a presión, el cual se encuentra a 13.5 atm. Determine el volumen del recipiente. $T=20^\circ\text{C}$

Código en JULIA

```
clearconsole()

p=1.367887 #MPa
a=1.39
b=0.03913
R=0.008314
T=293.15
n=1.5
```

#Sustituyendo los valores de los datos en la ecuacion de Van Der Waals y simplificando nos queda la funcion siguiente $fv=13.5*(v^3)-$


```

4.44825615*(v*2)+5.483810475*v-0.32180610076 que resolveremos por
NewRap
v=0.06116
fv=1
i=0
while fv>0.01 || fv< -0.01
    global fv= (1.367888*(v^3))-(3.73616165*(v*2))+(5.483810475*v)-
0.321872
    global dfv= (4.103664*(v^2))-(7.4723233*v)+5.483810475
    global v=v-(fv/dfv)
    global i=i+1
    println("iteración ",i, " v=", v)
end
println("El volumen del recipiente es ",v, " litros")

```

Resultados del código

```

iteración 1 v=0.1490542981887125
iteración 2 v=0.2866268061267616
iteración 3 v=0.5202719841094468
iteración 4 v=0.9502010446431881
iteración 5 v=1.4470692236088847
iteración 6 v=1.1573730889402454
iteración 7 v=1.3728821990747018
iteración 8 v=1.2081065189051146
iteración 9 v=1.3357803270399753
iteración 10 v=1.235874440326479
iteración 11 v=1.3142749074956133
iteración 12 v=1.252483780016052
iteración 13 v=1.3012043891882157
iteración 14 v=1.2627093984039084
iteración 15 v=1.293119189355395
iteración 16 v=1.269070748087543
iteración 17 v=1.288083226721552
iteración 18 v=1.2730434480913533
iteración 19 v=1.2849378207210556
iteración 20 v=1.2755279624424718
iteración 21 v=1.2829710006335484
iteración 22 v=1.2770825666870826
iteración 23 v=1.2817405602082226
iteración 24 v=1.2780554786011844
El volumen del recipiente es 1.2780554786011844 litros

```

Análisis de resultados

Para poder encontrar el volumen, sustituimos los valores de los datos que nos dan e igualamos la ecuación a 0 para poder encontrar las raíces.

$$1.36788 = \frac{(1.5)(0.008314)(293.15)}{V - (1.5 * 0.03913)} - \frac{(1.5)^2(1.39)}{V^2}$$

$$1.36788 = \frac{3.65587365}{V - 0.058695} - \frac{5.483810475}{V^2}$$

Buscando el MCM que es $V - (1.5 * 0.03913) - V^2$, simplificando e igualando a 0 nos queda la siguiente ecuación que es con la que buscaremos las raíces en JULIA

$$f(V): 1.367888V^3 - 3.73616165V^2 + 5.483810475V - 0.321872 = 0$$

Para poder ocupar el método de Newton-Raphson necesitamos la derivada, la cual se saca manualmente.

$$f'(V) = 4.103664V^2 - 7.4723233V + 5.483810475$$

9. Las siguientes reacciones químicas se llevan a cabo en un sistema cerrado



En equilibrio, éstas pueden caracterizarse por

$$K_1 = \frac{c_c}{c_a^2 c_b}$$

$$K_2 = \frac{c_c}{c_a c_d}$$

donde la nomenclatura representa la concentración del componente i . Si x_1 y x_2 son el número de moles de C que se producen debido a la primera y segunda reacciones, respectivamente, emplee un método similar al del problema anterior para reformular las relaciones de equilibrio en términos de las concentraciones iniciales de los componentes. Después, use el método de Newton-Raphson para resolver el par de ecuaciones simultáneas no lineales para x_1 y x_2 si $K_1 = 4 \times 10^{-4}$, $K_2 = 3.7 \times 10^{-2}$, $c_{a,0} = 50$, $c_{b,0} = 20$, $c_{c,0} = 5$ y $c_{d,0} = 10$. Utilice un método gráfico para proponer los valores iniciales.

Resolviendo la ecuación 1:

$$k_1 = \frac{(C_{c0} + x_1)}{(C_{a0} - 2x_1)^2 (c_{b0} - x_1)}$$

$$4 * 10^{-4} = \frac{(5 + x_1)}{(50 - 2x_1)^2 (20 - x_1)}$$

$$0.0016x_1^3 - 0.112x_1^2 + 3.6x_1 - 15 = 0$$

Resolviendo para la reacción 2:

$$K_2 = \frac{(C_{c0} + x_2)}{(C_{a0} - x_2)(C_{d0} - x_2)}$$

$$3.7 \times 10^{-2} = \frac{(5 + x_2)}{(50 - x_2)(10 - x_2)}$$

$$3.7 \times 10^{-2} x_2^2 - 3.22x_2 + 13.5 = 0$$

Código en JULIA

```
clearconsole()
#Ecuacion 1
XC1=0
qn1=1000
er1=100
XA1=100
while er1<-0.01 || er1>0.01
```

```

global eq1=(0.0016*XC1^3)-(0.112*XC1^2)+(3.6*XC1)-15
global eqd1=(0.0048*XC1^2)-(0.224*XC1)+3.6

global XA1=XC1-(eq1/eqd1)
global er1=(eq1-qn1)/eqd1

    global XC1=XA1
    global qn1=eq1
end
println("El número de moles que se producen de C debido a la reacción
1 es: ", XA1)
println("Con un error de: ", er1)

#Ecuacion 2
XC2=0
qn2=1000
er2=100
XA2=100
while er2<-0.01 || er2>0.01
    global eq2=(0.037*XC2^2)-(3.22*XC2)+13.5
    global eqd2=(0.074*XC2)-(3.22)

global XA2=XC2-(eq2/eqd2)
global er2=(eq2-qn2)/eqd2

    global XC2=XA2
    global qn2=eq2
end
println("El número de moles que se producen de C debido a la reacción
2 es: ", XA2)
println("Con un error de: ", er2)

```

Resultados del código

El número de moles que se producen de C debido a la reacción 1 es:
4.846948886493388
Con un error de: 7.902143526329835e-6
El número de moles que se producen de C debido a la reacción 2 es:
4.4166978704735
Con un error de: 0.0006388932320051349

Análisis de resultados

Al resolver la ecuación de equilibrio para la primera reacción por el método de Newton Raphson se obtuvo un valor de 4.8469 mol de C lo cual indica que no se consumió todo el reactivo B, pero al hacerlo de acuerdo con su estequiometria se producirían 20 moles de C considerando que todos los moles de B reaccionaran. El resultado obtenido por el método de Newton Raphson nos indica que coincide con el valor de la constante de equilibrio K, porque si el valor de $K < 1$ quiere decir que la mayoría de los reactivos quedan sin reaccionar.

Al resolver la ecuación de equilibrio para la segunda reacción por el método de Newton Raphson se obtuvo un valor de 4.4169 mol de C lo cual indica que no se consumió todo el reactivo D, pero al hacerlo de acuerdo con su estequiometria se producirían 10 moles de C considerando que todos los moles de D reaccionaran. El resultado obtenido por el método de Newton Raphson nos indica que coincide con el valor de la constante de equilibrio K, porque si el valor de $K < 1$ quiere decir que la mayoría de los reactivos quedan sin reaccionar.

10. En una sección de tubo, la caída de presión se calcula así:

$$\Delta p = f \frac{L \rho V^2}{2D}$$

donde Δp = caída de presión (Pa), f = factor de fricción, L = longitud del tubo [m], ρ = densidad (kg/m³), V = velocidad (m/s) y D = diámetro (m). Para el flujo turbulento, la ecuación de Colebrook proporciona un medio para calcular el factor de fricción,

$$\frac{1}{\sqrt{f}} = -2.0 \log \left(\frac{\varepsilon}{3.7D} + \frac{2.51}{\text{Re} \sqrt{f}} \right)$$

donde ε = rugosidad (m) y Re = número de Reynolds,

$$\text{Re} = \frac{\rho V D}{\mu}$$

donde μ = viscosidad dinámica (N · s/m²).

Determine Δp para un tramo horizontal de tubo liso de 0.2 m de longitud, dadas $\rho = 1.23$ kg/m³, $\mu = 1.79 \times 10^{-5}$ N · s/m², $D = 0.005$ m, $V = 40$ m/s y $\varepsilon = 0.0015$ mm. Utilice un método numérico para determinar el factor de fricción. Obsérvese que los tubos lisos tienen $\text{Re} < 10^5$, un valor inicial apropiado se obtiene con el uso de la fórmula de Blasius, $f = 0.316/\text{Re}^{0.25}$. Repita el cálculo, pero para un tubo de acero comercial más rugoso ($\varepsilon = 0.045$ mm).

Código en JULIA

```
clearconsole()
d=1.23
u=0.0000179
D=0.005
V=40
E=0.0000015
E2 = 0.000045
a=0.0001
b=0.3
Re=(d*V*D)/u

Fa=-2*log((E/3.7*D)+(2.51/Re*sqrt(a)))-(1/sqrt(a))
Fb=-2*log((E/3.7*D)+(2.51/Re*sqrt(b)))-(1/sqrt(b))

err=100
Wx=0
n=0

while err>=0.0008
    global Wx=b-((Fb*(a-b))/(Fa-Fb))
    global Wx1=-2*log((E/3.7*D)+(2.51/Re*sqrt(Wx)))-(1/sqrt(Wx))
    if Wx1==0
```

```

println("La raíz exacta es:",xR)
end
if a<0 && b>0
    global a=Wx
    global Fa=-2*log((E/3.7*D)+(2.51/Re*sqrt(a)))-(1/sqrt(a))
end
if a>0 && b>0
    global b=Wx
    global Fb=-2*log((E/3.7*D)+(2.51/Re*sqrt(b)))-(1/sqrt(b))

end
global n=n+1
global err=abs(b-a)*100/(2^n)
end

println("*****E = 0.0015 mm*****")
println("La raíz aproximada es: ", Wx)
println("Error =: ",err)
println("Iteraciones =: ",n)
println("NRe = ",Re)

d2=1.23
u2=0.0000179
D2=0.005
V2=40
E2 = 0.000045
a2=0.0001
b2=0.3
Re2=(d2*V2*D2)/u2

Fa2=-2*log((E2/3.7*D2)+(2.51/Re2*sqrt(a2)))-(1/sqrt(a2))
Fb2=-2*log((E2/3.7*D2)+(2.51/Re2*sqrt(b2)))-(1/sqrt(b2))

err2=100
Wx2=0
n2=0

while err2>=0.0008
    global Wx2=b2-((Fb2*(a2-b2))/(Fa2-Fb2))
    global Wx3=-2*log((E2/3.7*D2)+(2.51/Re2*sqrt(Wx2)))-(1/sqrt(Wx2))
    if Wx3==0
        println("La raíz exacta es:",xR2)
    end
    if a2<0 && b2>0
        global a2=Wx2
        global Fa2=-2*log((E2/3.7*D2)+(2.51/Re2*sqrt(a2)))-(1/sqrt(a2))
    end
    if a2>0 && b2>0
        global b2=Wx2
        global Fb2=-2*log((E2/3.7*D2)+(2.51/Re2*sqrt(b2)))-(1/sqrt(b2))
    end

    end
    global n2=n2+1
    global err2=abs(b2-a2)*100/(2^n2)
end

```

end

```
println("\n")
println("*****E2 = 0.045 mm*****")
println("La raíz aproximada es: ", Wx2)
println("Error2 =: ", err2)
println("Iteraciones =: ", n2)
println("NRe = ", Re2)
```

Resultados del código

```
****E = 0.0015 mm***
La raíz aproximada es: 0.027791170516339912
Error =: 0.0006760539676840799
Iteraciones =: 12
NRe = 13743.016759776536
```

```
****E2 = 0.045 mm***
La raíz aproximada es: 0.027847044411475737
Error2 =: 0.0006774180764520444
Iteraciones =: 12
NRe = 13743.016759776536
```

Análisis de resultados

Para obtener los valores iniciales del factor de fricción empleamos la ecuación de Blasius:

$$f = \frac{0.316}{13745.01676^{0.25}} = 0.0291854614$$

Los valores iniciales para el factor de fricción, a y b, en el programa que se usaron fueron 0.0001 y 0.30 respectivamente, el resultado para $\varepsilon=0.0015$ fue de $f = 0.027791170516339912$ un error de 0.0006760539676840799 y para $\varepsilon=0.045$ fue de $f = 0.027847044411475737$ con un error de 0.0006774180764520444

Como se obtiene un resultado parecido en ambos métodos podemos decir que el resultado es correcto

Caída de presión con un tubo de $\varepsilon=0.0015$

$$\Delta p = 0.027791170516339912 \frac{(0.2m) \left(1.23 \frac{kg}{m^3}\right) \left(40 \frac{m}{s}\right)^2}{(2)(0.005m)} = 1093.860472 \frac{kg}{m-s}$$

Caída de presión con un tubo de $\varepsilon=0.045$

$$\Delta p = 0.027847044411475737 * \frac{(0.2m) \left(1.23 \frac{kg}{m^3}\right) \left(40 \frac{m}{s}\right)^2}{(2)(0.005m)} = 1096.059669 \frac{kg}{m-s}$$

Se puede deducir a través de los resultados que cuanto más pequeño sea ε mayor será Δp .

11. Un intercambiador de calor que opera a contracorriente se caracteriza mediante la ecuación:

$$Q = UA\Delta T_{lm}$$

Donde Q es el calor transferido entre corrientes y A es el área del intercambiador. ΔT_{lm} se calcula a partir de la expresión:

$$\Delta T_{lm} = \frac{(T'_2 - T_2) - (T'_1 - T_1)}{\ln(T'_2 - T_2) - \ln(T'_1 - T_1)}$$

Donde T_2 y T_1 son las temperaturas de entrada de las corrientes 1 y 2. T'_2 y T'_1 son las temperaturas de salida de ambas corrientes. Por primera ley de la Termodinámica, se considera que:

$$Q = \dot{m}C_p(T_2 - T_1)$$

m es el flujo y C_p es la capacidad calorífica de la corriente interna del intercambiador. Como $T_2 > T_1$, Q es positivo. Se puede realizar un balance similar para la corriente externa.

La corriente interna tiene un flujo de 3 kg/s y un C_p de 2.3 kJ/kg °C. La corriente externa tiene valores de 5 kg/s y 4 kJ/kg °C respectivamente. El intercambiador enfría la corriente externa desde 100 hasta 50 °C. La corriente interna entra en rangos de 15 a 25 °C. Grafique las temperaturas de salida y áreas para cada temperatura de entrada de la corriente interna. Determine un área promedio para la operación del intercambiador y grafique las temperaturas de salida para la corriente interna.

Código en JULIA

```
clearconsole()
cp1=2300
m2=5
m1=3+0.75*m2
cp2=4000
Tci=0
Thi=50
Th0=100
Tc0=15
Q=1000000
n=0
U=400
i=0
n=4
B=0
Ap=0
incre=2.5
using PyPlot
clearconsole()
Tci=zeros(n+1);A=zeros(n+1)
for i in 0:n
    global Tci[i+1]=(Q/(m1*cp1))+(Tc0)
    println("Tc",i+1,"= ",Tci[i+1])
    global delT=((Thi-Tc0)-(Th0-Tci[i+1]))/log((Thi-Tc0)/(Th0-Tci[i+1]))
    println("Delta T",i+1,"= ",delT)
    global A[i+1]=Q/(U*delT)
    println("A",i+1,"= ",A[i+1])
    println()
    global B=B+A[i+1]
    global Tc0=Tc0+incre
end
```

```

global Ap=B/(n+1)
println("Área promedio= ",Ap)
figure("gráfica de temperaturas")
plot(A,Tci,color="green")
scatter(A,Tci,color="purple")
xlabel("Área de transferencia")
ylabel("Temperatura de entrada")
grid("on")

```

Resultados del código

```

Tc1= 79.4122383252818
Delta T1= 27.15952929638349
A1= 92.04872340452879

```

```

Tc2= 81.9122383252818
Delta T2= 24.594073218669973
A2= 101.65050651724448

```

```

Tc3= 84.4122383252818
Delta T3= 22.0131204278135
A3= 113.56863322481345

```

```

Tc4= 86.9122383252818
Delta T4= 19.41020149674869
A4= 128.79825077646737

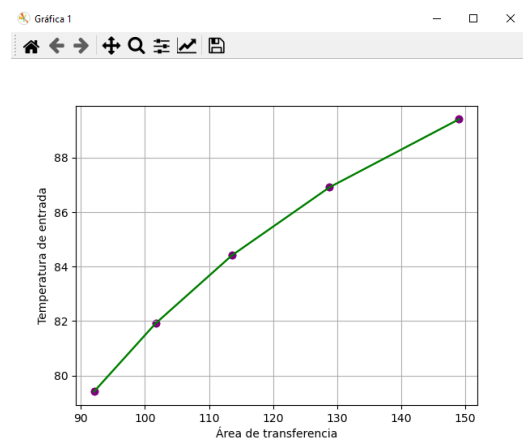
```

```

Tc5= 89.4122383252818
Delta T5= 16.774468463024668
A5= 149.03601896600517

```

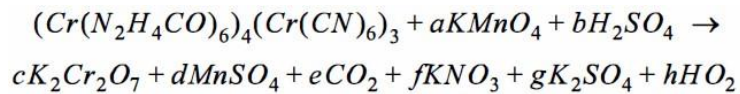
Área promedio= 117.02042657781185



Análisis de resultados

Calculamos el área de cada temperatura de entrada de la corriente interna aplicando los conocimientos obtenidos anteriormente en la materia de Procesos de Separación II y con eso obtuvimos el área promedio.

12. Considere los coeficientes $a \dots h$ en la siguiente reacción. Utilizando el método algebraico, obtenga los valores de los coeficientes para que quede balanceada.



Código en JULIA

Resultados del código

Análisis de resultados

13. Un compresor opera a una razón de compresión R_c de 3.0 (esto significa que la presión del gas en la salida es tres veces mayor que en la entrada). Los requerimientos de energía del compresor H_p se determinan por medio de la ecuación:

$$HP = \frac{zRT_1}{MW} \frac{n}{n-1} (R_c^{(n-1)/n} - 1)$$

Suponga que los requerimientos de energía del compresor son exactamente iguales a zRT_1/MW , y encuentre la eficiencia politrópica n del compresor. El parámetro z es la compresibilidad del gas en las condiciones de operación del compresor, R es la constante de los gases, T_1 es la temperatura del gas en la entrada del compresor y MW es el peso molecular del gas.

Código en Julia

```
clearconsole()
a=0.70
b=0.99
Rc=3

Fa=(a/(a-1))*((Rc^((a-1)/a))-1)
Fb=(b/(b-1))*((Rc^((b-1)/b))-1)

err=100
Wx=0
n=0

while err>=0.001
    global Wx=a-((Fa*(a-b))/(Fa-Fb))
    global Wx1=(Wx/(Wx-1))*((Rc^((Wx-1)/Wx))-1)
    if Wx1==0
        println("La raíz exacta es:",xR)
    end
    if a<0 && b>0
        global a=Wx
        global Fa=(a/(a-1))*((Rc^((a-1)/a))-1)
    end
    if a>0 && b>0
        global b=Wx
        global Fb=(b/(b-1))*((Rc^((b-1)/b))-1)
    end

    end
    global n=n+1
    global err=abs(b-a)*100/(2^n)
end
```

```
println("La raíz aproximada es= ",Wx)
println("Error= ",err)
println("Iteraciones= ",n)
```

Resultados del código

```
La raíz aproximada es= 0.8200080315737595
Error= 0.0008961525312899992
Iteraciones= 17
```

Análisis de resultados

En este problema, debido a que el cociente $\frac{zRT_1}{MW} = HP$ la ecuación que vamos a utilizar para obtener el valor de n es:

$$\frac{n}{n-1} \left(R_c^{(n-1)/n} - 1 \right) = 0$$

Como aproximaciones a la raíz utilizamos a y b con valores de 0.7 y 0.99 pues se busca que la eficiencia sea cercana a 1. A través del programa se obtiene que **$n = 0.8200080315737595$** con un error de 0.0008961525312899992.

14. El objetivo de este problema es realizar el computo de los diagramas de equilibrio P-V y P-T para un componente simple, un fluido, que se describe mediante las ecuaciones de Van der Waals:

$$P = \frac{RT}{V-b} - \frac{a}{V^2}$$

Donde P es la presión, R es la constante de gases ideales, T es la temperatura, V es el volumen molar y a y b son constantes. El fenómeno que analizaremos considera valores supercríticos, por lo cual las unidades a utilizar son MPa para la presión y L/mol para el volumen. El fluido analizado tiene valores de $a = 2.5 \text{ MPa L}^2/\text{mol}^2$ y $b = 0.2 \text{ L/mol}$. Los valores críticos pueden ser calculados utilizando las relaciones de Van der Waals

$$V_c = 3b, P_c = \frac{a}{27b^2}, T_c = \frac{8a}{27Rb}$$

También se pueden computar los valores para las siguientes funciones:

$$H^d = RT(Z-1) - \frac{a}{V}$$

$$S^d = R \ln Z + R \ln \frac{V-b}{V}$$

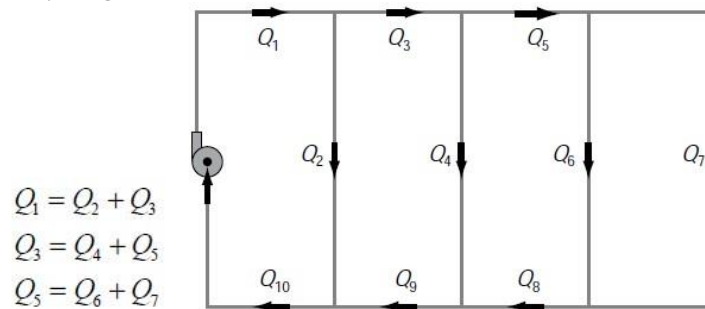
Donde $Z = PV/RT$

Considere un fluido supercrítico a 470 K y $P = 3 \text{ MPa}$. Calcule el resto de las propiedades. Para hacer su diagrama considere el diagrama de isoterma incluyendo los valores de V .

Código en Julia

Resultados del código

15. Un fluido se bombea en la red de tubos que se muestra en la figura. En estado estacionario, se cumplen los balances de flujo siguientes:



donde Q_i = flujo en el tubo i [m^3/s]. Además, la caída de presión alrededor de los tres lazos en los que el flujo es hacia la derecha debe ser igual a cero. La caída de presión en cada tramo de tubo circular se calcula por medio de la ecuación:

$$\Delta P = \frac{16}{\pi^2} \frac{f L \rho}{2 D^5} Q^2$$

donde ΔP = caída de presión (Pa), f = factor de fricción (adimensional), L = longitud del tubo (m), ρ = densidad del fluido (kg/m^3) y D = diámetro del tubo (m). Escriba un programa (o desarrolle un algoritmo en algún paquete de software de matemáticas) que permita calcular el flujo en cada tramo de tubo, dado que $Q_1 = 1 \text{ m}^3/\text{s}$ y $\rho = 1.23 \text{ kg}/\text{m}^3$. Todos los tubos tienen $D = 500 \text{ mm}$ y $f = 0.005$. Las longitudes de los tubos son: $L_3 = L_5 = L_8 = L_9 = 2 \text{ m}$; $L_2 = L_4 = L_6 = 4 \text{ m}$, y $L_7 = 8 \text{ m}$.

Repita el problema, pero incorpore el hecho de que el factor de fricción se calcula con la ecuación de Von Karman. Obsérvese que, para un tubo circular, $V = 4Q/\pi D^2$. Asimismo, suponga que el fluido tiene una viscosidad de $1.79 \times 10^{-5} \text{ N} \cdot \text{s}/\text{m}^2$.

Código en Julia

```
clearconsole()
Q1=1
Q2=0
Q3=0
Q4=0
Q5=0
Q6=0
Q7=1
Q8=0
Q9=0
i=0;

dfQ=0
fQ=1
qn7=0

while fQ>0.01 || fQ<-0.01

    global fQ=1-(5.59586530386363*(Q7))-(10.2426406871193*((Q7)^2))
    global dfQ= -(5.59586530386363)-(2*10.2426406871193*(Q7))
    global qn7=Q7-(fQ/dfQ)
    global i=i+1
    println("iteración ",i," x=",qn7)
    global Q7=qn7
end
```

```

Q4=sqrt(5+2*(sqrt(2)))*(Q7)
Q5=(1+(sqrt(2)))*Q7
Q6=(sqrt(2))*Q7
Q3=Q4+Q5
Q2=1-Q3
Q8=Q5
Q9=Q3

println("Q9= ",Q9)
println("Q8= ",Q8)
println("Q7= ",Q7)
println("Q6= ",Q6)
println("Q5= ",Q5)
println("Q4= ",Q4)
println("Q3= ",Q3)
println("Q2= ",Q2)
println("Q1= ",Q1)

println("SEGUNDA PARTE")
f1=0
a=(4*Q1)/(pi*0.0000179*0.5)
i=0;
    x1=3
    x2=0
    fc=1
    i=0
    while fc>0.0001 || fc<-0.0001
        global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
        global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
        global f1=(x1+x2)/2
        global fc1=4*log10(a*(f1^0.5))-0.4-(1/(f1^0.5))
        global i=i+1
        global fc=fc1
        global r1=fc1*fx1
        global r2=fc1*fx2
        if r1>0
            global x1=f1
        end
        if r2>0
            global x2=f1
        end
    end
f2=0
a=(4*Q2)/(pi*0.0000179*0.5)
i=0;
    x1=3
    x2=0
    fc=1
    i=0
    while fc>0.0001 || fc<-0.0001
        global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
        global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
        global f2=(x1+x2)/2
        global fc1=4*log10(a*(f2^0.5))-0.4-(1/(f2^0.5))

```

```

        global i=i+1
        global fc=fc1
        global r1=fc1*fx1
        global r2=fc1*fx2
        if r1>0
            global x1=f2
        end
        if r2>0
            global x2=f2
        end
    end
end
f3=0
a=(4*Q3)/(pi*0.0000179*0.5)
i=0;
x1=3
x2=0
fc=1
i=0
while fc>0.0001 || fc<-0.0001
    global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
    global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
    global f3=(x1+x2)/2
    global fc1=4*log10(a*(f3^0.5))-0.4-(1/(f3^0.5))
    global i=i+1
    global fc=fc1
    global r1=fc1*fx1
    global r2=fc1*fx2
    if r1>0
        global x1=f3
    end
    if r2>0
        global x2=f3
    end
end
f4=0
a=(4*Q4)/(pi*0.0000179*0.5)
i=0;
x1=3
x2=0
fc=1
i=0
while fc>0.0001 || fc<-0.0001
    global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
    global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
    global f4=(x1+x2)/2
    global fc1=4*log10(a*(f4^0.5))-0.4-(1/(f4^0.5))
    global i=i+1
    global fc=fc1
    global r1=fc1*fx1
    global r2=fc1*fx2
    if r1>0
        global x1=f4
    end
    if r2>0

```

```

global x2=f4
end
end
f5=0
a=(4*Q5)/(pi*0.0000179*0.5)
i=0;
x1=3
x2=0
fc=1
i=0
while fc>0.0001 || fc<-0.0001
global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
global f5=(x1+x2)/2
global fc1=4*log10(a*(f5^0.5))-0.4-(1/(f5^0.5))
global i=i+1
global fc=fc1
global r1=fc1*fx1
global r2=fc1*fx2
if r1>0
global x1=f5
end
if r2>0
global x2=f5
end
end
f6=0
a=(4*Q6)/(pi*0.0000179*0.5)
i=0;
x1=3
x2=0
fc=1
i=0
while fc>0.0001 || fc<-0.0001
global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
global f6=(x1+x2)/2
global fc1=4*log10(a*(f6^0.5))-0.4-(1/(f6^0.5))
global i=i+1
global fc=fc1
global r1=fc1*fx1
global r2=fc1*fx2
if r1>0
global x1=f6
end
if r2>0
global x2=f6
end
end
f7=0
a=(4*Q7)/(pi*0.0000179*0.5)
i=0;
x1=3
x2=0

```

```

fc=1
i=0
while fc>0.0001 || fc<-0.0001
global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
global f7=(x1+x2)/2
global fc1=4*log10(a*(f7^0.5))-0.4-(1/(f7^0.5))
global i=i+1
global fc=fc1
global r1=fc1*fx1
global r2=fc1*fx2
if r1>0
global x1=f7
end
if r2>0
global x2=f7
end
end
f8=0
a=(4*Q8)/(pi*0.0000179*0.5)
i=0
x1=3
x2=0
fc=1
i=0
while fc>0.0001 || fc<-0.0001
global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
global f8=(x1+x2)/2
global fc1=4*log10(a*(f8^0.5))-0.4-(1/(f8^0.5))
global i=i+1
global fc=fc1
global r1=fc1*fx1
global r2=fc1*fx2
if r1>0
global x1=f8
end
if r2>0
global x2=f8
end
end
f9=0
a=(4*Q9)/(pi*0.0000179*0.5)
i=0;
x1=3
x2=0
fc=1
i=0
while fc>0.0001 || fc<-0.0001
global fx1=4*log10(a*(x1^0.5))-0.4-(1/(x1^0.5))
global fx2=4*log10(a*(x2^0.5))-0.4-(1/(x2^0.5))
global f9=(x1+x2)/2
global fc1=4*log10(a*(f9^0.5))-0.4-(1/(f9^0.5))
global i=i+1

```

```

global fc=fc1
global r1=fc1*fx1
global r2=fc1*fx2
if r1>0
global x1=f9
end
if r2>0
global x2=f9
end
end
    println("f1= ",f1)
    println("f2= ",f2)
    println("f3= ",f3)
    println("f4= ",f4)
    println("f5= ",f5)
    println("f6= ",f6)
    println("f7= ",f7)
    println("f8= ",f8)
    println("f9= ",f9)

```

Resultados del código

```

iteración 1 x=0.431063895536412
iteración 2 x=0.2012464197558626
iteración 3 x=0.14558159867106074
iteración 4 x=0.14188178226435968
iteración 5 x=0.1418652917908438
Q9= 0.7394226435489077
Q8= 0.3424931114714716
Q7= 0.1418652917908438
Q6= 0.20062781968062784
Q5= 0.3424931114714716
Q4= 0.39692953207743614
Q3= 0.7394226435489077
Q2= 0.26057735645109226
Q1= 1
SEGUNDA PARTE
f1= 0.004186034202575684
f2= 0.005593031644821167
f3= 0.004453182220458984
f4= 0.005087345838546753
f5= 0.005256772041320801
f6= 0.005944490432739258
f7= 0.006461441516876221
f8= 0.005256772041320801
f9= 0.004453182220458984

```

Análisis de resultados

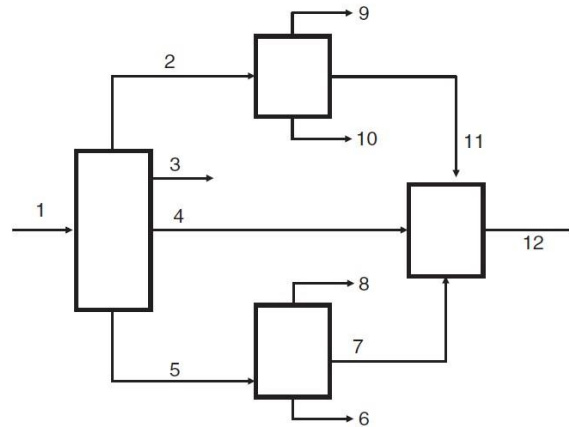
Con la ecuación para un flujo turbulento obtuvimos los flujos en cada tramo de tubo.

16. Considere un proceso que tiene la siguiente estructura y es descrito por el sistema de ecuaciones que se muestra:

$$m_1 = m_2 + m_3 + m_4 + m_5$$

$$m_2 = m_9 + m_{10} + m_{11}$$

$$m_5 = m_8 + m_7 + m_6$$



A partir de los datos siguientes, genere un código que resuelva los valores para cada una de las corrientes involucradas en el sistema.

$$m_{12} = m_4 + m_7 + m_{11}$$

$$m_1 = 100$$

$$m_5 = 5m_8$$

$$0.84m_{12} = m_4 + m_7$$

$$0.7m_1 = m_2 + m_3$$

$$0.55m_1 = m_9 + m_{12}$$

$$0.2m_9 = m_{10}$$

$$0.85m_2 = m_9 + m_{11}$$

$$3.2m_6 = m_7 + m_8$$

Código en JULIA

```

clearconsole()
A=[1 0 0 0 0 0 0 0 0 0 0 0 0
  -1 1 1 1 1 0 0 0 0 0 0 0 0
  -.7 1 1 0 0 0 0 0 0 0 0 0 0
  -.55 0 0 0 0 0 0 0 0 1 0 0 1
  0 -1 0 0 0 0 0 0 0 1 1 1 0
  0 -.85 0 0 0 0 0 0 0 1 0 1 0
  0 0 0 0 -1 1 1 1 0 0 0 0 0
  0 0 0 0 -1 0 0 5 0 0 0 0 0
  0 0 0 0 0 -3.2 1 1 0 0 0 0 0
  0 0 0 0 0 0 0 0 -.2 1 0 0 0
  0 0 0 1 0 0 1 0 0 0 0 -.84
  0 0 0 1 0 0 1 0 0 0 1 -1]
B=[100;0;0;0;0;0;0;0;0;0;0;0;0]
c=inv(A)*B

```

```
println("Valores obtenidos de las corrientes involucradas")
```

```

for i=1:12
    println("m",[i],"=",c[i])
end

```

Resultados del código

Valores obtenidos de las corrientes involucradas

```

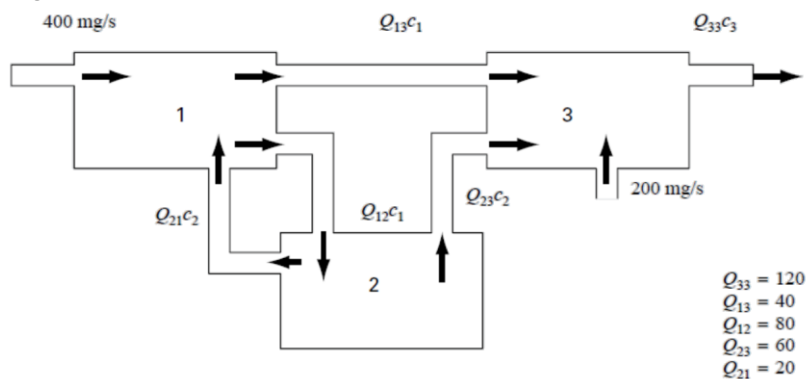
m[1]=100.0
m[2]=40.000000000000004
m[3]=29.999999999999954
m[4]=9.45652173913058
m[5]=20.543478260869414
m[6]=4.891304347826036
m[7]=11.543478260869405
m[8]=4.108695652173901
m[9]=30.000000000000003
m[10]=6.0000000000000009
m[11]=3.999999999999995
m[12]=24.999999999999982

```

Análisis de resultados

En el planteamiento del problema nos costo trabajo generar la matriz y acomodarla para que el resultado que diera fuera el correcto, al final logramos obtener el valor de las corrientes involucradas en el sistema.

17. En la figura se muestran tres reactores conectados por tubos. Como se indica, la tasa de transferencia de productos químicos a través de cada tubo es igual a la tasa de flujo (Q , en unidades de metros cúbicos por segundo) multiplicada por la concentración del reactor desde el que se origina el flujo (c , en unidades de miligramos por metro cúbico). Si el sistema se encuentra en estado estacionario (estable), la transferencia de entrada a cada reactor balanceará la de salida. Desarrolle las ecuaciones del balance de masa para los reactores y resuelva las tres ecuaciones algebraicas lineales simultáneas para sus concentraciones.



Para la solución de este problema comenzamos haciendo un balance de materia que entra a los reactores.

$$Q_{12}C_2 + 400 = Q_{12}C_1 + Q_{13}C_1 \quad \text{Reactor 1}$$

$$Q_{12}C_1 = Q_{21}C_2 + Q_{23}C_2 \quad \text{Reactor 2}$$

$$Q_{13}C_1 + Q_{23}C_2 = Q_{33}C_3 + 200 \quad \text{Reactor 3}$$

Generamos un código basado en la solución de matrices.

Código en Julia

```
clearconsole()
A=[120 -20 1;-80 80 1;-40 -60 120]
B=[400;0;200]

x=inv(A)*B

println("Las concentraciones son:")
println("C1= ",x[1])
println("C2= ",x[2])
println("C3= ",x[3])
```

Resultados del código

```
Las concentraciones son:
C1= 3.9385245901639343
C2= 3.8770491803278686
C3= 4.918032786885246
```

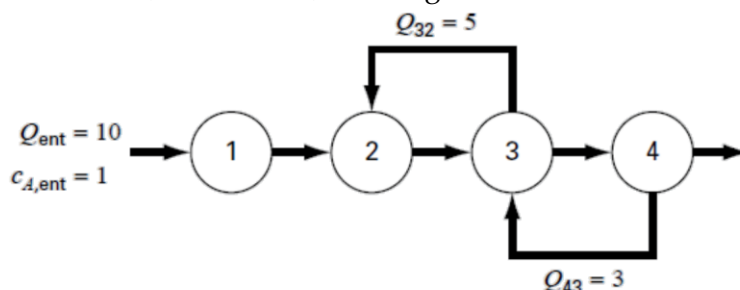
Análisis de resultados

Para la solución de este problema comenzamos haciendo un balance de materia y obtuvimos un sistema de ecuaciones presentado a continuación:

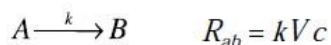
$$\begin{aligned} Q_{12}C_2 + 400 &= Q_{12}C_1 + Q_{13}C_1 && \text{Reactor 1} \\ Q_{12}C_1 &= Q_{21}C_2 + Q_{23}C_2 && \text{Reactor 2} \\ Q_{13}C_1 + Q_{23}C_2 &= Q_{33}C_3 + 200 && \text{Reactor 3} \end{aligned}$$

Generamos un código basado en la solución de matrices para así obtener las concentraciones C1, C2, C3.

18. Una reacción de primer orden, irreversible, tiene lugar en cuatro reactores.



Así, la tasa a la cual A se transforma en B se representa por



Los reactores tienen volúmenes diferentes, y debido a que se operan a temperaturas diferentes, cada uno tiene distinta tasa de reacción:

Reactor	V, L	k, h^{-1}
1	25	0.05
2	75	0.1
3	100	0.5
4	25	0.1

Determine la concentración de A y B en cada uno de los reactores en estado estable.

Código en JULIA

```
clearconsole()

C1=zeros(5,5)
C1R=[1;0;0;0;0]
C1[1]=1
V=[25 75 100 25]
k=[0.05 0.1 0.5 0.1]
Ref=[0 0 5 3 0]

#balance
E=[1 0 0 0 0;-1 1 0 0 0;0 -1 1 0 0;0 0 -1 1 0; 0 0 0 -1 1]
R=[10; 0; 5; -2; -3]
Q=inv(E)*R

#balance reactores
for i=1:4
    global C1[i+1,i]=Q[i]-Ref[i]-k[i]*V[i]
    global C1[i+1,i+1]=Ref[i+1]-Q[i+1]
end

C1s=inv(C1)*C1R

#No se pueden tener concentraciones negativas
for i=1:5
    if C1s[i]<0
        C1s[i]=0
    end
end

println("Las corrientes del sistema son ",Q)
println("Las concentraciones de A en el reactor son ",C1s)
```

Resultados del código

```
Las corrientes del sistema son [10.0, 10.0, 15.0, 13.0, 10.0]
Las concentraciones de A en el reactor son [1.0, 0.875, 0.21875, 0.0,
0.0]
```

Análisis de resultados

Suponiendo que los reactores son de flujo continuo (CSTR), podemos hacer balances de materia teniendo en cuenta las ecuaciones de diseño y la ley de velocidad de reacción.

Balance de materia en el sistema

Construimos un sistema de ecuaciones para encontrar los flujos volumétricos Q y se resuelve por el método de Gauss Jordan.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ Q_4 \\ Q_5 \end{bmatrix} = \begin{bmatrix} 10 \\ 0 \\ -5 \\ -2 \\ -3 \end{bmatrix}$$

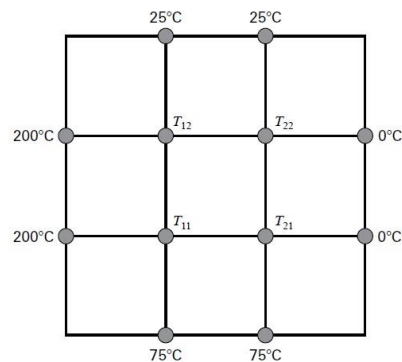
Se hace un balance para general que aplique a cualquiera de los reactores del diagrama, tomando en cuenta el reflujo y considerando estado estacionario, nos queda de la siguiente manera:

$$f_A = C_A Q \quad Ref = C_A R \quad Q_n C_{An} + R_{n+1} C_{An+1} - Q_{n+1} C_{An+1} - R_n C_{An} = k_n C_{An} V_n$$

19. La distribución de temperatura de estado estable en una placa caliente está modelada por la ecuación de Laplace:

$$0 = \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}$$

Si se representa la placa por una serie de nodos, las diferencias finitas divididas se pueden sustituir por las segundas derivadas, lo que da como resultado un sistema de ecuaciones algebraicas lineales. Utilice el método de Gauss-Seidel para resolver cuáles son las temperaturas de los nodos que se aprecian en la figura.



Código en JULIA

```
clearconsole()
t12=0
t22=0
t21=0
t11=0
E=100;x=0
a=75;b=25;c=0;d=200
while E>0.0001
    global t12=(b+d+t22+t11)/4
    global t22=(b+c+t21+t12)/4
    global t11=(t12+t21+a+d)/4
    global t21=(t22+c+a+t11)/4
    global E=abs((t12-x)/t12)
    global x=t12
end
println("Las temperaturas de los nodos son:")
println("T12 = ",t12)
```

```
println("T22 = ",t22)
println("T21 = ",t21)
println("T11 = ",t11)
```

Resultado del código

Las temperaturas de los nodos son:

```
T12 = 93.74713897705078
T22 = 43.74856948852539
T21 = 56.249284744262695
T11 = 106.24856948852539
```

Análisis de resultados

Generamos una matriz y la acomodamos de tal manera que pudiéramos optimizar los valores.

20. Se sabe que el esfuerzo a la tensión de un plástico se incrementa como función del tiempo que recibe tratamiento a base de calor. Se obtuvieron los datos siguientes:

Tiempo	10	15	20	25	40	50	55	60	75
Esfuerzo a la tensión	5	20	18	40	33	54	70	60	78

Ajuste una línea recta a estos datos y utilice la ecuación para determinar el esfuerzo a la tensión en un tiempo de 32 min. Repita el análisis para una línea recta con intersección en el origen.

Código en JULIA

```
clearconsole()
using PyPlot

x=[10;15;20;25;40;50;55;60;75]
y=[5;20;18;40;33;54;70;60;78]
scatter(x,y, color="red", label="datos")
grid("on")

n=9
xyi=0
xi=0
yi=0
xi2=0

#Sumatorias
for i=1:n
    global xyi=xyi+(x[i]*y[i])
    global xi=xi+x[i]
    global yi=yi+y[i]
    global xi2=xi2+(x[i]^2)
end

a1=(n*xyi-xi*yi)/(n*xi2-xi^2)
xp=xi/n
```

```

yp=yi/n
a0=yp-a1*xp

println(" a0 = ",a0,", a1 = ",a1)

#función
f=zeros(n)
for i=1:n
    global f[i]=a1*x[i]+a0
end
plot(x,f)

println("La ecuación resultante es y = ",a1,"x + ",a0)

#Evaluar función en 32 min

mn=32
fp=a1*mn+a0
println("El esfuerzo de tensión en ",mn," min es de ",fp)
scatter(mn,fp,color= "purple", label="32min")
legend(bbox_to_anchor=(0.1,1), loc=2)

```

Resultados del código

```

a1 = 1.0589673913043478
La ecuación resultante es y = 1.0589673913043478x + 0.8179347826087024
El esfuerzo de tensión en 32 min es de 34.70489130434783

```



Análisis de datos

Hicimos una regresión por mínimos cuadrados y se obtuvo que la ecuación lineal que mejor se ajusta a los datos experimentales es $y =$

$1.0589673913043478x + 0.8179347826087024$ y el esfuerzo de tensión a los 32 minutos es de 34.70489130434783.

21. Los siguientes datos corresponden a valores de k y T para la reacción $\text{CH}_4 + \text{O} \rightarrow \text{CH}_3 + \text{OH}$.

$T(\text{K})$	595	623	761	849	989	1076	1146	1202	1382	1445	1562
$k \times 10^{20} \text{ (m}^3/\text{s)}$	2.12	3.12	14.4	30.6	80.3	131	186	240	489	604	868

Utilice el método de mínimos cuadrados para obtener una función de la forma:

$$\ln k = C + b \ln T - \frac{D}{T}$$

Considere que $f_1(T)=1$, $f_2(T)=\ln(T)$ y $f_3(T)=-1/T$ para obtener los valores de b , C y D

Código en JULIA

```
clearconsole()
T=[595 623 761 849 989 1076 1146 1202 1382 1445 1562]
k=[2.12 3.12 14.4 30.6 80.3 131 186 240 489 604 868]

using PyPlot
scatter(T,k,color="green",label="Datos")
xlabel("Temperatura (K)")
ylabel("k")
n=11
SlnT=0
SinvT=0
SlnTc=0
SlnTinvT=0
SinvTc=0
Slnk=0
SlnTlnk=0
SinvTlnk=0

for i=1:n
    global SlnT=SlnT+log(T[i])
    global SinvT=SinvT+(-1/T[i])
    global SlnTc=SlnTc+((log(T[i]))^2)
    global SlnTinvT=SlnTinvT+((log(T[i]))*(-1/T[i]))
    global SinvTc=SinvTc+((-1/T[i])^2)
    global Slnk=Slnk+log(k[i])
    global SlnTlnk=SlnTlnk+((log(T[i]))*(log(k[i])))
    global SinvTlnk=SinvTlnk+((-1/T[i])*(log(k[i])))
end

A=[n SlnT SinvT; SlnT SlnTc SlnTinvT; SinvT SlnTinvT SinvTc]
B=[Slnk; SlnTlnk; SinvTlnk]
R=inv(A)*B
```



```

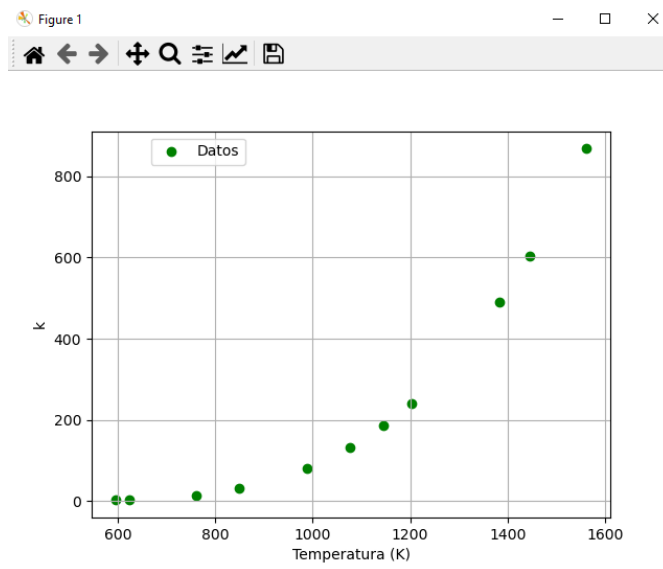
legend(bbox_to_anchor=(0.1,1), loc=2)
grid("on")
clearconsole()

println("La función es ln(k)= ",R[1]," + ",R[2]," ln(T) +
",R[3]," /T")
println("      donde b= ",R[2]," , C= ",R[1]," , D= ",R[3])

```

Resultados del código

La función es $\ln(k) = -6.391021719429773 + 2.1215975439879458 \ln(T) + 3815.3416919058654/T$
donde $b = 2.1215975439879458$, $C = -6.391021719429773$, $D = 3815.3416919058654$



Análisis de resultados

Obteniendo los valores de b , c y D con el método de mínimos cuadrados y sustituyendo en la función obtenemos que $\ln(k)$ es igual a $-6.391021719429773 + 2.1215975439879458 \ln(T) + 3815.3416919058654/T$

22. A continuación, se presentan datos de la vasija de un reactor de crecimiento bacterial (una vez que terminó la fase de retraso). Se permite que las bacterias crezcan tan rápido como sea posible durante las primeras 2.5 horas, y después se les induce a producir una proteína recombinante, la cual disminuye el crecimiento bacterial en forma significativa. El crecimiento teórico de las bacterias se describe por medio de:

$$\frac{dX}{dt} = \mu X$$

donde X es el número de bacterias, y μ es la tasa de crecimiento específico de las bacterias durante el crecimiento exponencial. Con base en los datos, estime la tasa de crecimiento específico de las bacterias durante las primeras 2 horas de crecimiento, así como durante las siguientes 4 horas de crecimiento.

Tiempo, h	0	1	2	3	4	5	6
[Células], g/L	0.100	0.332	1.102	1.644	2.453	3.660	5.460

Código en JULIA

```
clearconsole()
X=[0 1 2 3 4 5 6]
Y=[0.1 0.332 1.102 1.644 2.453 3.660 5.460]

using PyPlot

scatter(X,Y,color="green",label="Datos")
xlabel("X- Tiempo")
ylabel("Y- Colonias de Bacterias")

Sx=0
Sy=0
Sx2=0
Sx3=0
Sx4=0
Sx5=0
Sx6=0
Sxy=0
Sx2y=0
Sx3y=0
n=7
rr=0
SC3=0
SC4=0

for i=1:n
    global Sx=Sx+X[i]
    global Sy=Sy+Y[i]
    global Sx2=Sx2+(X[i]^2)
    global Sx3=Sx3+(X[i]^3)
    global Sx4=Sx4+(X[i]^4)
    global Sx5=Sx5+(X[i]^5)
    global Sx6=Sx6+(X[i]^6)
    global Sxy=Sxy+(X[i]*Y[i])
    global Sx2y=Sx2y+((X[i]^2)*Y[i])
    global Sx3y=Sx3y+((X[i]^3)*Y[i])
end
yp=Sy/n
xp=Sx/n

A=[n Sx Sx2 Sx3;Sx Sx2 Sx3 Sx4;Sx2 Sx3 Sx4 Sx5;Sx3 Sx4 Sx5 Sx6]
B=[Sy ;Sxy ;Sx2y ;Sx3y]
R=inv(A)*B

for i=1:n
    global SC3=SC3+(((Y[i])-yp)^2)
    global SC4=SC4+(((Y[i])-R[1])-(R[2])*X[i])-(X[i]^2)*R[3])-(X[i]^3)*R[4])^2)
end

YN=zeros(n)

for i=1:n
```

```

    global YN[i]=R[1]+(R[2]*X[i])+(R[3]*(X[i]^2))+(R[4]*(X[i]^3))
end

plot(transpose(X),YN,color="blue",label="Polinomial: Tercer grado")
legend(bbox_to_anchor=(0.1,1), loc=2)
grid("on")

println("La función es y= ",R[4],"x^3 + ",R[3],"x^2 + ",R[2],"x + ",R[1])
rr=(SC3-SC4)/SC3
println("El coeficiente de determinacion (R^2) es de ",rr)

h=0.001

function funf(x)
    f=(R[4]*(x^3))+(R[3]*(x^2))+(R[2]*(x))+(R[1])
    return f
end

fpp=zeros(n)

for i=1:n
    fpp[i]=((-funf(X[i]+2h))+(8*funf(X[i]+h))-(8*funf(X[i]-h))+funf(X[i]-2h))/12h
end

miu=zeros(n)
for i=1:n
    global miu[i]=fpp[i]*(1/(Y[i]))
    println("El valor de miu a t= ",X[i]," es de ",miu[i]," h^-1")
end
println("\n")
println("El valor de miu a t= ",X[3]," h es de ",miu[3]," h^-1")
println("El valor de miu a t= ",X[5]," h es de ",miu[5]," h^-1")
println("El valor de miu a t= ",X[7]," h es de ",miu[7]," h^-1")

```

Resultados del código

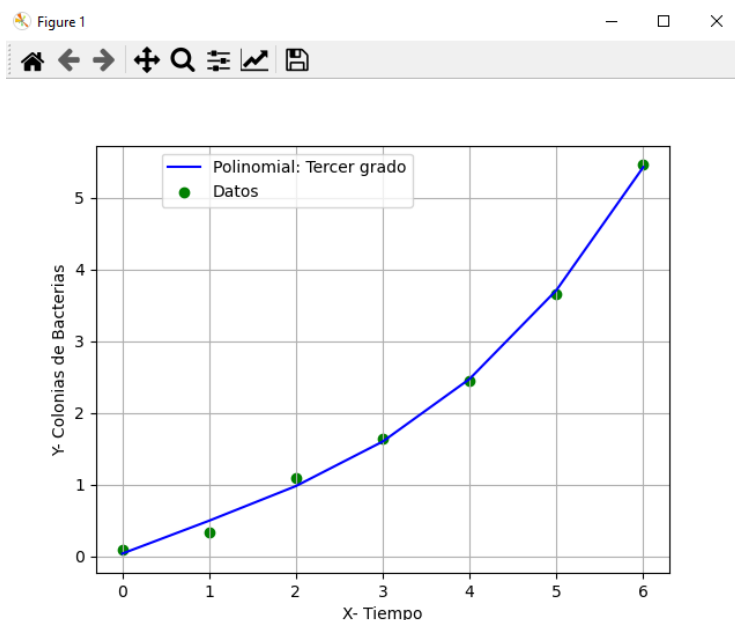
```

x^3 + -0.04454761904764837x^2 + 0.4843690476190545x + 0.0415476190475772
El coeficiente de determinacion (R^2) es de 0.9976654070504796
El valor de miu a t= 0 es de 4.8436904761905755 h^-1
El valor de miu a t= 1 es de 1.361517498565496 h^-1
El valor de miu a t= 2 es de 0.48382810474443927 h^-1
El valor de miu a t= 3 es de 0.442721005677076 h^-1
El valor de miu a t= 4 es de 0.4223351387029825 h^-1
El valor de miu a t= 5 es de 0.3982630757222233 h^-1
El valor de miu a t= 6 es de 0.36498124890999645 h^-1

El valor de miu a t= 2 h es de 0.48382810474443927 h^-1
El valor de miu a t= 4 h es de 0.4223351387029825 h^-1

```

El valor de μ a $t= 6$ h es de $0.36498124890999645 \text{ h}^{-1}$



Análisis de resultados

Con los datos obtenidos vemos que en un tiempo de dos horas la tasa de crecimiento es de $0.48382810474443927 \text{ h}^{-1}$, después de dos horas disminuye a $0.4223351387029825 \text{ h}^{-1}$ y en las dos últimas horas disminuye hasta $0.36498124890999645 \text{ h}^{-1}$. Comparando los resultados llegamos a la conclusión de que disminuye aproximadamente 0.059 h^{-1} cada dos horas.

23. La entalpía de un gas real es función de la presión como se describe a continuación. Los datos se tomaron para un fluido real. Estime la entalpía del fluido a 400 K y 50 atm (evalúe la integral de 0.1 atm a 50 atm).

$$H \int_0^P \left(V - T \left(\frac{\partial V}{\partial T} \right)_P \right) dP$$

P, atm	V, L		
	T=350 K	T=400 K	T=450 K
0.1	220	250	282.5
5	4.1	4.7	5.23
10	2.2	2.5	2.7
20	1.35	1.49	1.55
25	1.1	1.2	1.24
30	0.90	0.99	1.03
40	0.68	0.75	0.78
45	0.61	0.675	0.7
50	0.54	0.6	0.62

Código en JULIA

```
clearconsole()
P=[0.1 5 10 20 25 30 40 45 50]
Temp=[350 400 450]
VT350=[220 4.1 2.2 1.35 1.1 0.9 0.68 0.61 0.54]
VT400=[250 4.7 2.5 1.49 1.2 0.99 0.75 0.675 0.6]
VT450=[282.5 5.23 2.7 1.55 1.24 1.03 0.78 0.7 0.62]

Sxy=0
```

```

Sx=0
Sy=0
Sy2=0
Sx2=0
Sx3=0
Sx4=0
Sx5=0
Sx6=0
Sx2y=0
Sx3y=0
n=9

Bet=zeros(n)

for i=1:n
    Bet[i]=( (VT450[i]-VT350[i]) / (Temp[3]-Temp[1]) ) / (VT400[i])
end

V=zeros(n)

for i=1:n
    global V[i]=VT400[i]*(1-Temp[2]*Bet[i])
    global Sxy=Sxy+(P[i]*V[i])
    global Sx=Sx+P[i]
    global Sy=Sy+V[i]
    global Sy2=Sy2+(V[i]^2)
    global Sx2=Sx2+(P[i]^2)
    global Sx3=Sx3+(P[i]^3)
    global Sx4=Sx4+(P[i]^4)
    global Sx5=Sx5+(P[i]^5)
    global Sx6=Sx6+(P[i]^6)
    global Sx2y=Sx2y+((P[i]^2)*V[i])
    global Sx3y=Sx3y+((P[i]^3)*V[i])
end

yp=Sy/n
xp=Sx/n

A=[n Sx Sx2 Sx3;Sx Sx2 Sx3 Sx4;Sx2 Sx3 Sx4 Sx5;Sx3 Sx4 Sx5 Sx6]
B=[Sy ;Sxy ;Sx2y ;Sx3y]
C=inv(A)*B

a=P[1]
b=P[9]
N=11
N2=N-1
h=(b-a)/N2

```

```

mult3=0
part2=0

fa=C[4]*(a^3)+C[3]*(a^2)+C[2]*(a)+C[1]
fb=C[4]*(b^3)+C[3]*(b^2)+C[2]*(b)+C[1]

ff=zeros(N2)
x=a

for i=1:N2
    global x=x+h
    global ff[i]=C[4]*(x^3)+C[3]*(x^2)+C[2]*(x)+C[1]
    println("Iteración: ",i," X= ",x," f(x)= ",ff[i])
end

for i=2:1:N2-1
    if i%3 ==0
        global mult3=mult3+ff[i]
    else
        global part2=part2+ff[i]
    end
end

Integral=(3/8)*h*(fa+fb+2*mult3+3*part2)
println("La integral es: ",Integral," atm*L")

v=Integral*(101.325/1000)
println("Equivalente a: ",v," kJ ")

```

Resultados del código

```

Iteración: 1, X= 5.09, f(x)= 0.2808417357530062
Iteración: 2, X= 10.08, f(x)= 0.4923213578069704
Iteración: 3, X= 15.07, f(x)= 0.6027401478598395
Iteración: 4, X= 20.060000000000002, f(x)= 0.6327401075459327
Iteración: 5, X= 25.050000000000004, f(x)= 0.6029632384995678
Iteración: 6, X= 30.040000000000006, f(x)= 0.5340515423550645
Iteración: 7, X= 35.030000000000001, f(x)= 0.4466470207467408
Iteración: 8, X= 40.020000000000001, f(x)= 0.36139167530891525
Iteración: 9, X= 45.010000000000001, f(x)= 0.2989275076759079
Iteración: 10, X= 50.000000000000014, f(x)= 0.279896519482036
La integral es: 20.035818799240438 atm*L
Equivalente a: 2.0301293398330373 kJ

```

Análisis de resultados

Con los datos que nos proporciona el problema y realizando algunas iteraciones obtenemos que la integral de 0.1 a 50 atmosferas es igual a 2.0301293398330373 kJ.

24. La primera ley de la difusión de Fick establece que

$$\text{Flujo de masa} = -D \frac{dc}{dx}$$

donde el flujo de masa = cantidad de masa que pasa a través de una unidad de área por unidad de tiempo ($\text{g}/\text{cm}^2/\text{s}$), D = coeficiente de difusión (cm^2/s), c = concentración y x = distancia (cm). Se mide la concentración, que se presenta a continuación, de un contaminante en los sedimentos en el fondo de un lago ($x = 0$ en la interfase sedimento-agua y aumenta hacia abajo):

x, cm	0	1	3
$c, 10^{-6} \text{ g}/\text{cm}^3$	0.06	0.32	0.6

Utilice la mejor técnica numérica de diferenciación disponible para estimar la derivada en $x = 0$. Emplee esta estimación junto con la ecuación de la ley de Fick para calcular el flujo de masa del contaminante que se desprende de los sedimentos hacia las aguas superiores ($D = 1.52 \times 10^{-6} \text{ cm}^2/\text{s}$). Para un lago con $3.6 \times 10^6 \text{ m}^2$ de sedimentos, ¿cuánto contaminante será transportado hacia el lago durante un año?

Código en JULIA

```
clearconsole()

x=[0;1;3] #cm
y=[0.06;0.32;0.6] #g cm(-3)

using PyPlot
plot(x,y,color="red",label="datos")
grid("on")

#Ajuste lineal
n=3
xyi=0
xi=0
yi=0
xi2=0

#sumatorias necesarias para el método

for i=1:n
    global xyi=xyi+(x[i]*y[i])
    global xi=xi+x[i]
    global yi=yi+y[i]
    global xi2=xi2+(x[i]^2)
end

a1=(n*xyi-xi*yi)/(n*xi2-xi^2)
xp=xi/n
yp=yi/n
a0=yp-a1*xp
```

```

println("a0= ",a0," a1= ",a1 )

#Generar la función obtenida y graficarla
f=zeros(n)
for i=1:n
    global f[i]=a1*x[i]+a0
end
plot(x,f,color="green",label="funcion")
legend(bbox_to_anchor=(0.1,1), loc=2)

#Cálculos adicionales
D=1.52*10^(-6) #cm^2 s^-1
A=3.6*10^(6) #m^2
t=60*20*24*365 #años
Fm=a1*D
println("El flujo de masa es aproximadamente ",Fm,"g cm^-2 s^-1")
S=Fm*A*t/1000
println("Los sedimentos desprendidos son ",S," kg")

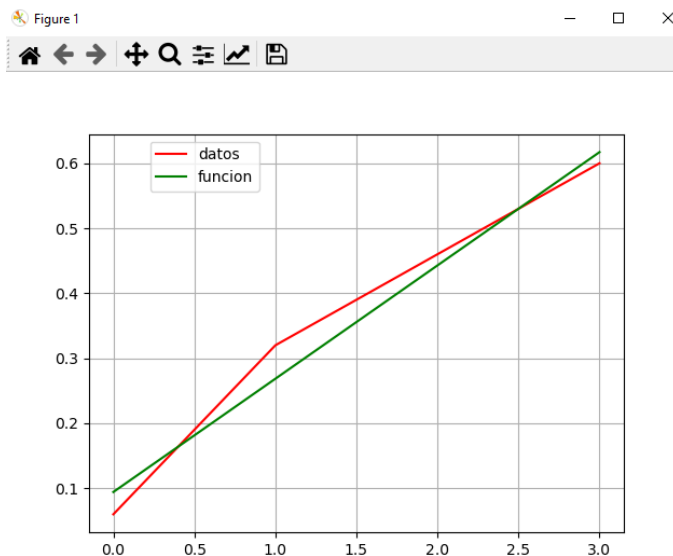
```

Resultados del código

```

a0= 0.09428571428571436
a1= 0.17428571428571424
El flujo de masa es aproximadamente 2.6491428571428577e-7g cm^-2 s^-1
Los sedimentos desprendidos son 10025.20429714286 kg

```



Análisis de resultados

Hicimos un ajuste lineal suponiendo que la derivada es constante y obtuvimos que el valor de la derivada es 0.17428571428571424.

Calculamos el flujo de masa aproximado, obteniendo que es igual a $2.6491428571428577 \times 10^{-7} \text{ g cm}^{-2} \text{ s}^{-1}$ y los sedimentos desprendidos en un año son 10025.20429714286 kg

25. Teniendo en cuenta que la presión de vapor para el n-butano a 350 K es 9.4573 bar, encuentre los valores molares de vapor saturado y líquido saturado de n-butano en estas condiciones utilizando la ecuación de Redlich-Kwong:

$$q = \frac{\Psi}{\Omega} T_r^{-\frac{3}{2}}$$

$$\beta = \Omega \frac{P_r}{T_r}$$

Para vapor saturado:

$$Z = 1 + \beta - q\beta \frac{(Z - \beta)}{Z(Z + \beta)}$$

Para líquido saturado:

$$Z = \beta + \frac{Z(Z + \beta)(1 + \beta - Z)}{q\beta}$$

Código en JULIA

```
clearconsole()
psi=0.42748
omega=0.08664
Pc=37.96
P=9.4573
Tc=425.1
T=350
Tr=T/Tc
Pr=P/Pc
q=(psi/omega)*(Tr^(-3/2))
B=omega*(Pr/Tr)

println("\n")
println("vapor saturado")
println("\n")

ep=100
i=0
Z0vs=1

while ep>0.001
    global Zvs=1+B-(q*B)*((Z0vs-B)/(Z0vs*(Z0vs+B)))
    global ep=abs((Zvs-Z0vs)/Zvs)*100
    global i=i+1
    println("Número de iteración ",i," , Z= ",Zvs," , ep= ",ep)
    global Z0vs=Zvs
end
println("\n")
println("El factor de compresibilidad para vapor saturado ",Zvs," , con un
error relativo porcentual de: ",ep,"\n")

println("\n")
println("Líquido saturado")
println("\n")
ep=100
```

```

i=0
Z0ls=0.01
while ep>0.001
    global Zls=B+((Z0ls*(Z0ls+B)*(1+B-Z0ls))/(q*B))
    global ep=abs((Zls-Z0ls)/Zls)*100
    global i=i+1
    println("Número de iteración ",i," Z= ",Zls," ep= ",ep)
    global Z0ls=Zls
end
println("\n")
println("El factor de compresibilidad para liquido saturado ",Zls," con un
error relativo porcentual de: ",ep)

```

Resultados del código

vapor saturado

```

Número de iteración 1, Z= 0.8619170202542762, ep= 16.02044935891713
Número de iteración 2, Z= 0.8371912502486977, ep= 2.953419544009022
Número de iteración 3, Z= 0.8319581969307948, ep= 0.6290043582968889
Número de iteración 4, Z= 0.8308133713350396, ep= 0.13779575958382806
Número de iteración 5, Z= 0.8305611229270911, ep= 0.03037084219155263
Número de iteración 6, Z= 0.8305054556788315, ep= 0.0067028154816948455
Número de iteración 7, Z= 0.8304931665375964, ep= 0.00147974019898471
Número de iteración 8, Z= 0.8304904533705809, ep= 0.00032669454591423156

```

El factor de compresibilidad para vapor saturado 0.8304904533705809,
con un error relativo porcentual de: 0.00032669454591423156

Líquido saturado

```

Número de iteración 1, Z= 0.028342580751008612, ep= 64.71739786912619
Número de iteración 2, Z= 0.03512889522670369, ep= 19.31832593054726
Número de iteración 3, Z= 0.038552208320708437, ep= 8.879680939485644
Número de iteración 4, Z= 0.04046033888971194, ep= 4.7160518704619525
Número de iteración 5, Z= 0.04157596526883512, ep= 2.6833445042331703
...
Número de iteración 16, Z= 0.043302915098833064, ep= 0.012868841545660443
Número de iteración 17, Z= 0.04330638712119349, ep= 0.008017344764195944
Número de iteración 18, Z= 0.04330855053760464, ep= 0.004995356307934023
Número de iteración 19, Z= 0.043309898623185974, ep= 0.0031126500504211007
Número de iteración 20, Z= 0.04331073867699528, ep= 0.0019395970490625315
Número de iteración 21, Z= 0.04331126216219854, ep= 0.0012086583884380008
Número de iteración 22, Z= 0.04331158837912433, ep= 0.0007531862441276921

```

El factor de compresibilidad para liquido saturado 0.04331158837912433,
con un error relativo porcentual de: 0.0007531862441276921

Análisis de resultados

Para poder resolver este problema debimos comprobar primero que el cociente entre la presión y presión crítica debe de ser menor que la mitad del cociente entre la temperatura y la temperatura crítica.

26. Utilice el método de Euler para calcular el perfil de concentración para una reacción de segundo orden que se lleva a cabo en un reactor batch. La concentración inicial $c_0 = 3$ mol/litro y la constante k tiene un valor de 2 L/mol s. La ecuación diferencial que describe este proceso es:

$$\frac{dc}{dt} = -kc^2$$

Código en JULIA

```
clearconsole()
C0=3
k=2
a=0
b=60
n=400
h=(b-a)/n
t=zeros(n+1)
C=zeros(n+1)

C[1]=C0
t[1]=0

for i=1:n
    global fxy=(-k*(C[i]^2))
    global C[i+1]=C[i]+(h*fxy)
    global t[i+1]=t[i]+h
    println("Iteración ",i," [C] = ",C[i+1]," en un tiempo de = ",t[i+1],"
    fxy = ",fxy)
end

println("La concentración [C]= ",C[n]," mol/L, corresponde cuando pasan
",t[n]," segundos = ", t[n]/60," minutos ")

using PyPlot

figure(1)
plot(t,C,label="Concentración")
scatter(t,C)
xlabel("t Tiempo (seg)")
ylabel("C Concentración (mol/L)")
grid("on")
legend(bbox_to_anchor=(0.1,1), loc=2)
```

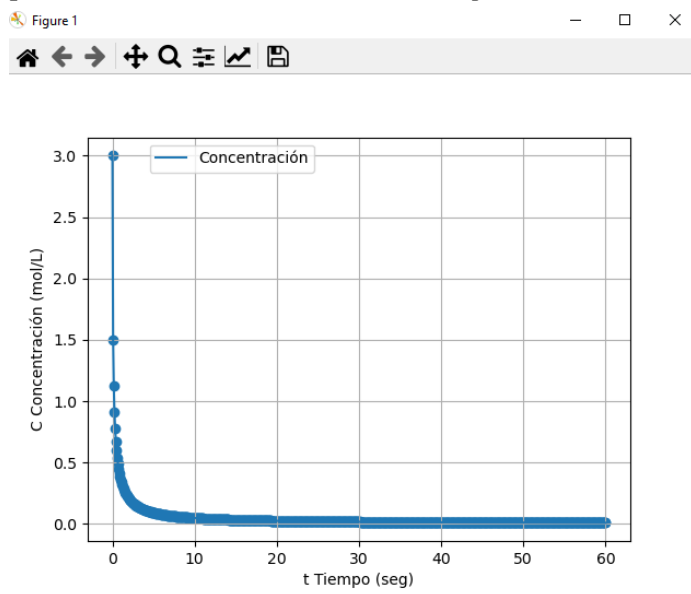
Resultados del código

```
Iteración 1 [C] = 0.300000000000000027 en un tiempo de = 0.15 fxy = -
18.0
Iteración 2 [C] = 0.273000000000000024 en un tiempo de = 0.3 fxy = -
0.180000000000000033
```

Iteración 3 [C] = 0.2506413000000002 en un tiempo de = 0.4499999999999996 fxy = -0.14905800000000027
 Iteración 4 [C] = 0.23179498162029316 en un tiempo de = 0.6 fxy = -0.12564212253138018
 Iteración 5 [C] = 0.21567630756898754 en un tiempo de = 0.75 fxy = -0.10745782700870408

...

Iteración 395 [C] = 0.00815468309257777 en un tiempo de = 59.2499999999999616 fxy = -0.00013365245255182428
 Iteración 396 [C] = 0.008134733435675657 en un tiempo de = 59.3999999999999615 fxy = -0.00013299771268074747
 Iteración 397 [C] = 0.008114881269254807 en un tiempo de = 59.549999999999961 fxy = -0.00013234777613899896
 Iteración 398 [C] = 0.008095125879850576 en un tiempo de = 59.699999999999961 fxy = -0.00013170259602820502
 Iteración 399 [C] = 0.008075466560947387 en un tiempo de = 59.849999999999961 fxy = -0.00013106212602125313
 Iteración 400 [C] = 0.008055902612894293 en un tiempo de = 59.999999999999961 fxy = -0.00013042632035395886
 La concentración [C]= 0.008075466560947387 mol/L, corresponde cuando pasan 59.849999999999961 segundos = 0.997499999999935 minutos



Análisis de resultados

La concentración disminuye mientras el tiempo avanza y llega un punto en el que se hace constante.

27. A partir de los análisis de los ejemplos 4.4-1 y 4.4-2 del texto de Fenómenos de Transporte de Bird (2ª Edición), compare el espesor de capa límite (δ) para la solución aproximada de la ecuación de Von Karman y la teoría de Blasius. Utilice el método Runge-Kutta para resolver la ecuación:

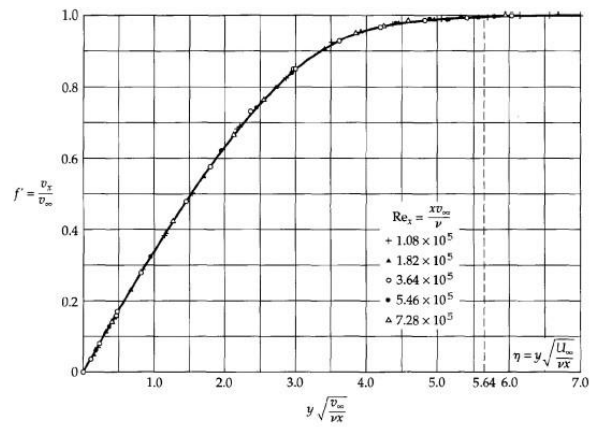
$$-ff'' = f'''$$

Sujeta a las condiciones de frontera:

$$\text{En } \eta = 0, f = 0 \text{ y } f' = 0$$

$$\text{En } \eta \rightarrow \infty, f' \rightarrow \infty$$

Compare su resultado con la curva obtenida por Blasius.

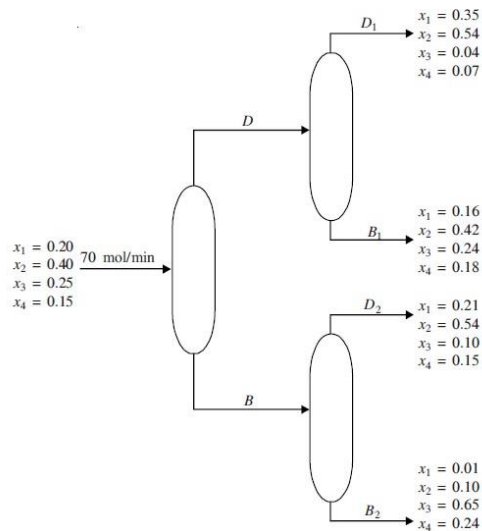


Código en JULIA

Resultados del código

Análisis de resultados

28. Benceno (1), tolueno (2), estireno (3) y xileno (4) se separan en la serie de columnas que se muestran en la figura. Determine la tasa de flujo molar en las corrientes D1, B1, D2 y B2. Determine también las composiciones en B y D así como las tasas de flujo molar de dichas corrientes.



Código en JULIA

```
clearconsole()
```

```
pm=[78.11,92.14,104.15,106.16]
```

```
nE=70
```

```
xE=[0.20,0.40,0.25,0.15]
```

```
mE=zeros(4)
```

```

mTE=0

for i=1:4
    global mE[i]=(nE*xE[i])*pm[i]
    global mTE=mTE+mE[i]
end

println("Composición inicial= ",mE)
println("Flujo másico inicial= ",mTE)

Acoef=[0.35 0.16 0.21 0.01;0.54 0.42 0.54 0.1;0.04 0.24 0.1 0.65;0.07
0.18 0.15 0.24]
Bflu=[mE[1];mE[2];mE[3];mE[4]]

FsF=inv(Acoef)*Bflu

println("D1,B1,D2,B2:",FsF)

FsE=[FsF[1]+FsF[2],FsF[3]+FsF[4]]

println("D, B:",FsE)

xD=zeros(4)
xB=zeros(4)
xD1=[0.35 0.54 0.04 0.07]
xB1=[0.16 0.42 0.24 0.18]
xD2=[0.21 0.54 0.10 0.15]
xB2=[0.01 0.1 0.65 0.24]

for i=1:4
    xD[i]=((FsF[1]*xD1[i]+FsF[2]*xB1[i]))/FsE[1]
    xB[i]=((FsF[3]*xD2[i]+FsF[4]*xB2[i]))/FsE[2]
end

println("Fracción en D: ", xD)
println("Fracción en B: ",xB)

```

Resultados del código

```

Composición inicial= [1093.54, 2579.92, 1822.625, 1114.68]
Flujo másico inicial= 6610.765

```

D1,B1,D2,B2:[1071.3298185483868, 1153.8085887096968,
2450.5446169354836, 1935.0819758064463]

D, B:[2225.1384072580836, 4385.62659274193]

Fracción en D: [0.25147865357958576, 0.47777599173447516,
0.14370668044254126, 0.12703867424339768]

Fracción en B: [0.12175345484228209, 0.3458576006530206,
0.34267799918372427, 0.18971094532097305]

Análisis de resultados

Para poder resolver el problema necesitábamos conocer el peso molecular de las especies involucradas en el proceso, para encontrar los flujos en la corriente realizamos balances de materia.