

ANEXOS

Jonás Herrero Laborda

Índice general

A. Conservación de la masa	2
B. Conservación de energía	4
C. Teorema de Prigogine	7
D. Coeficientes de la ecuación de balance entrópico	9
E. Relación entre el coeficiente de difusión y el coeficiente de linealidad	11
F. Diseño del programa	13
F.1. Discretización de la ecuación de Lamm	13
F.2. Estructura del programa	15
F.2.1. Subalgoritmos secundarios	15
F.2.2. Subalgoritmo "derivada"	17
F.2.3. Subalgoritmo "rungeKutta"	18
F.2.4. Algoritmo principal	20

Apéndice A

Conservación de la masa

Consideramos un volumen V delimitado por una superficie S en el que se encuentra una mezcla de n componentes, cada componente k tiene una densidad de masa ρ_k tal que $\rho = \sum_{k=1}^n \rho_k$, por lo que la variación temporal de la masa de la componente k es:

$$\frac{dm_k}{dt} = \frac{d}{dt} \int_V \rho_k dV = \int_V \frac{\partial \rho_k}{\partial t} dV = - \int_S \rho_k \vec{v}_k d\vec{S} \quad (\text{A.1})$$

Aplicando el teorema de la divergencia a la ecuación A.1 tenemos:

$$\int_V \frac{\partial \rho_k}{\partial t} dV = - \int_V \vec{\nabla} \cdot (\rho_k \vec{v}_k) dV \Rightarrow \frac{\partial \rho_k}{\partial t} = - \vec{\nabla} \cdot (\rho_k \vec{v}_k) \quad (\text{A.2})$$

Con ayuda de la velocidad de centro de masas \vec{v}_{CM} introducimos la derivada material representada mediante la ecuación A.4 para una función a (utilizamos \vec{v}_{CM} como la velocidad de flujo del sistema para la componente k) y el flujo de difusión \vec{J}_k para cada componente k con respecto a \vec{v}_{CM} :

$$\vec{v}_{CM} = \sum_{k=1}^n \frac{\rho_k \vec{v}_k}{\rho} \quad (\text{A.3})$$

$$\frac{da}{dt} = \frac{\partial a}{\partial t} + \vec{v}_{CM} \vec{\nabla} a \quad (\text{A.4})$$

$$\vec{J}_k = \rho_k (\vec{v}_k - \vec{v}_{CM}) \quad (\text{A.5})$$

Con todo esto la ecuación A.2 se transforma en:

$$\begin{aligned} \frac{d\rho_k}{dt} - \vec{v}_{CM} \vec{\nabla} \rho_k &= - \vec{\nabla} \cdot (\vec{J}_k + \rho_k \vec{v}_{CM}) = - \vec{\nabla} \cdot \vec{J}_k - \vec{\nabla} \cdot (\rho_k \vec{v}_{CM}) \\ \frac{d\rho_k}{dt} - \vec{v}_{CM} \vec{\nabla} \rho_k &= - \vec{\nabla} \cdot \vec{J}_k - \rho_k \vec{\nabla} \cdot \vec{v}_{CM} - \vec{v}_{CM} \vec{\nabla} \rho_k \end{aligned}$$

$$\frac{d\rho_k}{dt} = - \vec{\nabla} \cdot \vec{J}_k - \rho_k \vec{\nabla} \cdot \vec{v}_{CM} \quad (\text{A.6})$$

Sumando sobre las n componentes en la ecuación A.2 llegamos a:

$$\frac{\partial \rho}{\partial t} = - \vec{\nabla} \cdot (\rho \vec{v}_{CM}) \quad (\text{A.7})$$

Y aplicando A.4 a esta última expresión tenemos:

$$\frac{d\rho}{dt} = -\rho \vec{\nabla} \cdot \vec{v}_{CM} \quad (A.8)$$

Introduciendo las concentraciones de masa de cada componente $\rho = \frac{\rho_k}{c_k}$ en la ecuación A.6 usando A.4:

$$\begin{aligned} \rho \frac{dc_k}{dt} + c_k \frac{d\rho}{dt} &= -\vec{\nabla} \cdot \vec{J}_k - \rho c_k \vec{\nabla} \cdot \vec{v}_{CM} \\ \rho \frac{dc_k}{dt} - \rho c_k \vec{\nabla} \cdot \vec{v}_{CM} &= -\vec{\nabla} \cdot \vec{J}_k - \rho c_k \vec{\nabla} \cdot \vec{v}_{CM} \end{aligned}$$

$$\rho \frac{dc_k}{dt} = -\vec{\nabla} \cdot \vec{J}_k \quad (A.9)$$

Donde se ha usado la ecuación A.8 en el último paso. De esta forma llegamos a la ecuación A.9, que se usará para despejar a la ecuación de la conservación de la energía.

Apéndice B

Conservación de energía

Para obtener una expresión que determine la variación de energía interna consideraremos que la energía total por unidad de masa e del sistema tiene tres contribuciones, la cinética $\frac{1}{2}\vec{v}_{CM}^2$, la potencial ϕ y la interna u :

$$e = \frac{1}{2}\vec{v}_{CM}^2 + \phi + u \quad (\text{B.1})$$

De igual forma que se hizo para la masa vamos a considerar que trabajamos en un sistema con volumen V de forma que la variación de e con el tiempo es:

$$\frac{d}{dt} \int_V \rho e dV = \int_V \frac{\partial \rho e}{\partial t} dV = - \int_S \vec{J}_e d\vec{S} = - \int_V \vec{\nabla} \cdot \vec{J}_e dV$$

$$\frac{\partial \rho e}{\partial t} = -\vec{\nabla} \cdot \vec{J}_e \quad (\text{B.2})$$

Con \vec{J}_e el flujo de energía con el entorno. Para calcular el valor de las dos primeras contribuciones debemos plantear la ecuación de movimiento la cual expresa la dependencia de la velocidad de cada componente de la mezcla con la presión, que tiene que ver con las interacciones de corto alcance entre las partículas, y las fuerzas externas, que dependerán de la componente que tratemos ya que variará en función del tipo de material que se estudia:

$$\rho \frac{d\vec{v}_{CM}}{dt} = -\vec{\nabla} \cdot \bar{P} + \sum_k \rho_k \vec{F}_k \quad (\text{B.3})$$

Donde \bar{P} es el tensor presión y \vec{F}_k es la fuerza externa por unidad de masa que se ejerce sobre la componente k de la mezcla, las cuales se van a considerar conservativas de la forma $\vec{F}_k = -\vec{\nabla} \phi$, siendo ϕ el potencial que genera dicha fuerza independiente del tiempo. Multiplicando por \vec{v}_{CM} a ambos lados de la ecuación B.3 tenemos:

$$\begin{aligned} \rho \frac{d\vec{v}_{CM}}{dt} \vec{v}_{CM} &= -(\vec{\nabla} \cdot \bar{P}) \vec{v}_{CM} + \sum_k \rho_k \vec{F}_k \cdot \vec{v}_{CM} \\ \rho \frac{d(\frac{1}{2}\vec{v}_{CM}^2)}{dt} &= -\vec{\nabla} \cdot (\bar{P} \cdot \vec{v}_{CM}) + \bar{P} : \vec{\nabla} \vec{v}_{CM} + \sum_k \rho_k \vec{F}_k \cdot \vec{v}_{CM} \end{aligned} \quad (\text{B.4})$$

Con $\bar{P} : \vec{\nabla} \vec{v}_{CM} = \sum_{i,j} P_{ij} \frac{\partial}{\partial x_j} v_i$ siendo x_i y x_j las coordenadas espaciales cartesianas. Usando la expresión B.5 en la ecuación B.4:

$$\rho \frac{da}{dt} = \frac{\partial a \rho}{\partial t} + \vec{\nabla} \cdot (a \rho \vec{v}_{CM}) \quad (B.5)$$

$$\frac{\partial(\frac{1}{2}\vec{v}_{CM}^2 \rho)}{\partial t} + \vec{\nabla} \cdot (\frac{1}{2}\vec{v}_{CM}^2 \rho \vec{v}_{CM}) = -\vec{\nabla} \cdot (\bar{P} \cdot \vec{v}_{CM}) + \bar{P} : \vec{\nabla} \vec{v}_{CM} + \sum_k \rho_k \vec{F}_k \cdot \vec{v}_{CM}$$

$$\frac{\partial(\frac{1}{2}\vec{v}_{CM}^2 \rho)}{\partial t} = -\vec{\nabla} \cdot (\bar{P} \cdot \vec{v}_{CM} + \frac{1}{2}\vec{v}_{CM}^2 \rho \vec{v}_{CM}) + \bar{P} : \vec{\nabla} \vec{v}_{CM} + \sum_k \rho_k \vec{F}_k \cdot \vec{v}_{CM} \quad (B.6)$$

La ecuación B.6 se puede interpretar como el ritmo de cambio de la energía cinética por unidad de masa. A continuación haremos el mismo desarrollo para el ritmo de cambio de la energía potencial por unidad de masa (ecuación B.7), donde ϕ y ϕ_k son la energía potencial por unidad de masa de toda la mezcla y de la componente k respectivamente, utilizando la ecuación A.2 y los flujos de difusión definidos en la expresión A.5:

$$\rho\phi = \sum_k \rho_k \phi_k \quad (B.7)$$

$$\begin{aligned} \frac{\partial \rho \phi}{\partial t} &= \frac{\partial \sum_k \rho_k \phi_k}{\partial t} = \sum_k \phi_k \frac{\partial \rho_k}{\partial t} + \sum_k \rho_k \underbrace{\frac{\partial \phi_k}{\partial t}}_0 = \sum_k \phi_k (-\vec{\nabla} \cdot \rho_k \vec{v}_k) = \\ &\sum_k \phi_k (-\vec{\nabla} \cdot (\vec{J}_k + \rho_k \vec{v}_{CM})) = -\sum_k \phi_k \vec{\nabla} \cdot \vec{J}_k - \sum_k \phi_k \vec{\nabla} \cdot \rho_k \vec{v}_{CM} = \\ &-\vec{\nabla} \cdot (\sum_k \phi_k \vec{J}_k) + \sum_k \vec{J}_k \vec{\nabla} \phi_k - \vec{\nabla} \cdot (\underbrace{\sum_k \phi_k \rho_k \vec{v}_{CM}}_{\rho\phi}) + \sum_k \rho_k \vec{v}_{CM} \vec{\nabla} \phi_k \end{aligned}$$

$$\frac{\partial \rho \phi}{\partial t} = -\vec{\nabla} \cdot (\sum_k \phi_k \vec{J}_k) + \rho \phi \vec{v}_{CM} - \sum_k \vec{J}_k \vec{F}_k - \sum_k \rho_k \vec{v}_{CM} \vec{F}_k \quad (B.8)$$

Donde se ha utilizado $\vec{F}_k = -\vec{\nabla} \phi_k$. De esta forma sumando las ecuaciones B.6 y B.8 obtenemos el ritmo de cambio de la energía potencial y cinética:

$$\frac{\partial \rho(\frac{1}{2}\vec{v}_{CM}^2 + \phi)}{\partial t} = -\vec{\nabla} \cdot \left[\rho(\frac{1}{2}\vec{v}_{CM}^2 + \phi) \vec{v}_{CM} + \bar{P} \cdot \vec{v}_{CM} + \sum_k \phi_k \vec{J}_k \right] + \bar{P} : \vec{\nabla} \vec{v}_{CM} - \sum_k \vec{J}_k \vec{F}_k \quad (B.9)$$

Así, ya tenemos dos de las tres contribuciones a la energía total del sistema. Ahora vamos a considerar el flujo de energía total \vec{J}_e , el cual asumimos que tiene cuatro términos distintos:

- Flujo convectivo asociado al transporte de masa, de la forma $\rho e \vec{v}_{CM}$
- Flujo asociado a las fuerzas mecánicas ejercidas en el sistema, de la forma $\bar{P} \cdot \vec{v}_{CM}$
- Flujo de difusión producido por las fuerzas externas, de la forma $\sum_k \phi_k \vec{J}_k$
- Flujo de calor \vec{J}_q

$$\vec{J}_e = \rho e \vec{v}_{CM} + \bar{P} \cdot \vec{v}_{CM} + \sum_k \phi_k \vec{J}_k + \vec{J}_q \quad (B.10)$$

Si comparamos las ecuaciones B.2 y B.9 obtenemos que para que se cumpla la expresión B.10 la variación de energía interna con el tiempo debe ser:

$$\frac{\partial \rho u}{\partial t} = -\vec{\nabla} \cdot [\rho u \vec{v}_{CM} + \vec{J}_q] - \bar{P} : \vec{\nabla} \vec{v}_{CM} + \sum_k \vec{J}_k \vec{F}_k \quad (B.11)$$

Ahora separamos en dos componentes el tensor presión, uno definido por un escalar hidrostático y otro definido un tensor asociado a las fuerzas viscosas tal que $\bar{P} = p\bar{U} + \bar{\Pi}$ con \bar{U} la matriz unidad. También escribimos el flujo de calor \vec{J}_q en función de la cantidad de calor por unidad de masa q de tal forma que $\rho \frac{dq}{dt} = \vec{\nabla} \cdot \vec{J}_q$. Con todo esto utilizamos la ecuación B.5 para separar ρ y u obteniendo así la ecuación B.12:

$$\frac{du}{dt} = \frac{dq}{dt} - p \frac{d\nu}{dt} - \nu \bar{\Pi} : \vec{\nabla} \vec{v} + \nu \sum_{k=1}^n \vec{J}_k \cdot \vec{F}_k \quad (B.12)$$

Donde $\nu = \rho^{-1}$ es el volúmen específico y se ha utilizado $\bar{U} : \vec{\nabla} \vec{v} = \vec{\nabla} \cdot \vec{v} = \rho \frac{d\nu}{dt}$. De esta forma hemos obtenido una expresión que nos proporciona el cambio de energía interna con el tiempo.

Apéndice C

Teorema de Prigogine

El teorema de Prigogine establece que bajo equilibrio mecánico para la fuente de entropía interna σ expresada en la forma de la ecuación C.1 se puede sustituir la velocidad de centro de masas \vec{v}_{CM} con la que definimos los flujos de difusión \vec{J}_k por otra velocidad aleatoria \vec{v}_a sin producir cambios en los valores de σ .

$$\sigma = \frac{1}{T} \sum_{k=1}^n \vec{J}_k \left[\vec{F}_k - (\vec{\nabla} \mu_k)_T \right] \quad (\text{C.1})$$

Para demostrar este teorema desarrollamos los flujos de difusión:

$$\sigma = \frac{1}{T} \sum_{k=1}^n (\vec{v}_k - \vec{v}_{CM}) \rho_k \left[\vec{F}_k - (\vec{\nabla} \mu_k)_T \right] \quad (\text{C.2})$$

Así el problema se reduce a demostrar que:

$$\sum_{k=1}^n \rho_k \left[\vec{F}_k - (\vec{\nabla} \mu_k)_T \right] = 0 \quad (\text{C.3})$$

Para conseguirlo hacemos uso de la relación de Gibbs-Duhem:

$$\sum_{k=1}^n \rho_k d\mu_k = -\rho s dT + dp \quad (\text{C.4})$$

Como trabajamos en un sistema isotermo tenemos:

$$\sum_{k=1}^n \rho_k (\vec{\nabla} \mu_k)_T = \vec{\nabla} p \quad (\text{C.5})$$

Como se opera bajo equilibrio mecánico se cumple:

$$\vec{\nabla} p = \sum_{k=1}^n \rho_k \vec{F}_k \quad (\text{C.6})$$

Por lo que sustituyendo las ecuaciones C.5 y C.6 en la expresión C.3 demostramos su valor nulo. De esta forma si cambiamos la variable \vec{v}_{CM} por otra aleatoria \vec{v}_a no cambiará el valor de la fuente de entropía interna.

$$\sigma = \frac{1}{T} \sum_{k=1}^n (\vec{v}_k - \vec{v}_a) \rho_k \left[\vec{F}_k - (\vec{\nabla} \mu_k)_T \right] \quad (C.7)$$

Apéndice D

Coeficientes de la ecuación de balance entrópico

Para nuestro sistema en rotación podemos expresar la fuente de entropía interna σ en función de los flujos de difusión \vec{J}_k y de los gradientes del potencial químico $\vec{\nabla}\mu_k$ de $n - 1$ componentes de la mezcla a partir de las ecuaciones D.1 y D.2:

$$\sigma = \sum_{k=1}^n \vec{J}_k^a \cdot \vec{X}_k^a \quad (\text{D.1})$$

$$\vec{X}_k^a = \frac{1}{T} (1 - \nu_k \rho) \underbrace{(\omega^2 \vec{r} + 2\vec{v}_{CM} \wedge \vec{\omega})}_W - (\vec{\nabla}\mu_k)_{T,p} \quad (\text{D.2})$$

Primero introducimos las ecuaciones D.3, D.4 y D.5 con sus correspondientes demostraciones:

$$\sum_{k=1}^n \frac{a_k}{c_k} \vec{J}_k^a = 0 \quad (\text{D.3})$$

Demostración:

$$\begin{aligned} \sum_{k=1}^n \frac{a_k}{c_k} \vec{J}_k^a &= \sum_{k=1}^n \frac{a_k}{c_k} \rho_k (\vec{v}_k - \vec{v}_a) = \sum_{k=1}^n \rho a_k (\vec{v}_k - \vec{v}_a) = \\ &\rho \underbrace{\sum_{k=1}^n a_k \vec{v}_k}_1 - \rho \vec{v}_a \underbrace{\sum_{k=1}^n a_k}_1 = 0 \\ \sum_{k=1}^n c_k (\vec{\nabla}\mu_k)_{T,p} &= 0 \quad (\text{D.4}) \end{aligned}$$

Para demostrar esta expresión únicamente utilizamos la ecuación C.5 obtenida a partir de la relación de Gibbs-Duhem, como asumimos que la presión es uniforme:

$$\begin{aligned} \sum_{k=1}^n \rho_k (\vec{\nabla}\mu_k)_{T,p} &= \vec{\nabla}p = 0 \Rightarrow \sum_{k=1}^n \frac{\rho_k}{\rho} (\vec{\nabla}\mu_k)_{T,p} = \sum_{k=1}^n c_k (\vec{\nabla}\mu_k)_{T,p} = 0 \\ \sum_{k=1}^n c_k (1 - \rho \nu_k) &= 0 \quad (\text{D.5}) \end{aligned}$$

Para demostrar esta ecuación restamos las dos expresiones siguientes equivalentes a la unidad:

$$\begin{aligned} \sum_{k=1}^n \rho_k \nu_k &= 1 \quad ; \quad \sum_{k=1}^n c_k = 1 \\ \sum_{k=1}^n \rho_k \nu_k - \sum_{k=1}^n c_k &= 0 \Rightarrow \sum_{k=1}^n (\rho c_k \nu_k - c_k) = 0 \\ \sum_{k=1}^n c_k (1 - \rho \nu_k) &= 0 \end{aligned}$$

Con las ecuaciones D.3, D.4 y D.5 podemos determinar los siguientes siguientes parámetros de la n-ésima componente:

$$\begin{aligned} \vec{J}_n^a &= -\frac{c_n}{a_n} \sum_{k=1}^{n-1} \frac{a_k}{c_k} \vec{J}_k^a \\ (\vec{\nabla} \mu_n)_{T,p} &= -\frac{1}{c_n} \sum_{k=1}^{n-1} c_k (\vec{\nabla} \mu_k)_{T,p} \\ 1 - \rho \nu_n &= -\frac{1}{c_n} \sum_{k=1}^{n-1} c_k (1 - \rho \nu_k) \end{aligned}$$

Con todo esto sustituimos en la ecuación D.1:

$$\begin{aligned} \sigma &= \sum_{k=1}^{n-1} \vec{J}_k^a \cdot \vec{X}_k^a + \vec{J}_n^a \cdot \vec{X}_n^a \\ \sigma &= \sum_{k=1}^{n-1} \vec{J}_k^a \cdot \vec{X}_k^a - \frac{c_n}{a_n} \left[\sum_{k=1}^{n-1} \frac{a_k}{c_k} \vec{J}_k^a \right] \frac{1}{T} \left[-\frac{1}{c_n} \sum_{j=1}^{n-1} c_j (1 - \rho \nu_j) W + \frac{1}{c_n} \sum_{j=1}^{n-1} c_j (\vec{\nabla} \mu_j)_{T,p} \right] \\ \sigma &= \sum_{k=1}^{n-1} \vec{J}_k^a \cdot \vec{X}_k^a + \frac{1}{a_n} \left[\sum_{k=1}^{n-1} \frac{a_k}{c_k} \vec{J}_k^a \right] \frac{1}{T} \left[\sum_{j=1}^{n-1} c_j \left((1 - \rho \nu_j) W - (\vec{\nabla} \mu_j)_{T,p} \right) \right] \\ \sigma &= \sum_{k=1}^{n-1} \vec{J}_k^a \cdot \vec{X}_k^a + \left[\sum_{k=1}^{n-1} \sum_{j=1}^{n-1} \frac{a_k}{a_n} \frac{c_j}{c_k} \vec{J}_k^a \underbrace{\left((1 - \rho \nu_j) W - (\vec{\nabla} \mu_j)_{T,p} \right)}_{\vec{X}_k^a} \frac{1}{T} \right] \\ \sigma &= \sum_{k=1}^{n-1} \left[\vec{J}_k^a \cdot \vec{X}_k^a + \sum_{j=1}^{n-1} \frac{a_k}{a_n} \frac{c_j}{c_k} \vec{J}_k^a \cdot \vec{X}_k^a \right] \end{aligned}$$

Así, podemos escribir la fuente de entropía interna de la siguiente manera:

$$\sigma = \sum_{k=1}^{n-1} \vec{J}_k^a \cdot \vec{X}_k^a \quad (D.6)$$

$$\vec{X}_k^a = \sum_{j=1}^{n-1} \frac{Z_{kj}^a}{T} \left[(1 - \nu_j \rho) (\omega^2 \vec{r} + 2 \vec{v}_{CM} \wedge \vec{\omega}) - (\vec{\nabla} \mu_j)_{T,p} \right] \quad (D.7)$$

$$Z_{kj}^a = \delta_{kj} + \frac{a_k c_j}{a_n c_k} \quad (D.8)$$

Apéndice E

Relación entre el coeficiente de difusión y el coeficiente de linealidad

Como ya se dicho los flujos de los parámetros extensivos pueden escribirse en función de las fuerzas termodinámicas que los provocan multiplicados por unas constantes llamadas coeficientes de linealidad L_{ik} .

$$\underbrace{\vec{J}_i^a}_{\text{notación vectorial}} = \underbrace{\sum_k L_{ik} \vec{X}_k^a}_{\text{notación vectorial}} \quad \underbrace{\vec{J}^a}_{\text{notación tensorial}} = \underbrace{\bar{L}^a \cdot \vec{X}^a}_{\text{notación tensorial}} \quad (\text{E.1})$$

Vamos a suponer que en el sistema la única fuerza termodinámica existente es la provocada por el gradiente del potencial químico $\vec{\nabla}\bar{\mu}$, por lo que en notación tensorial tendríamos:

$$\vec{X}^a = -\bar{Z}^a \frac{\vec{\nabla}\bar{\mu}}{T} \quad (\text{E.2})$$

De forma que la entropía se determina mediante la ecuación E.3. Para las ecuaciones E.1, E.2 y E.3 \vec{J}^a se corresponde con un vector formado por las componentes \vec{J}_i^a con $1 \leq i \leq n-1$, \vec{X} se corresponde con un vector formado por las componentes \vec{X}_i^a con $1 \leq i \leq n-1$, $\vec{\nabla}\bar{\mu}$ es otro vector formado por las componentes $\vec{\nabla}\bar{\mu}_i$ con $1 \leq i \leq n-1$, \bar{Z}^a es una matriz formada por los coeficientes expresados en la ecuación D.8 (aunque aquí no tengamos en cuenta la fuerza centrífuga los coeficientes son iguales) y \bar{L}^a es otra matriz formada por otros coeficientes no conocidos. Estas dos últimas matrices son de dimensión $(n-1) \times (n-1)$. De forma que la entropía interna queda:

$$\sigma = \vec{J}^a \cdot \vec{X}^a \quad (\text{E.3})$$

Si consideramos los flujos de difusión con respecto a velocidades medias de diferentes parámetros a y b , la relación que mantienen entre ellos se expresan en la ecuación E.4:

$$\vec{J}^a = \bar{Q}^{ab} \cdot \vec{J}^b \quad (\text{E.4})$$

$$\bar{Q}_{kj}^{ab} = \delta_{kj} + \frac{c_k}{c_j} \left[a_n \frac{b_j}{b_n} - a_j \right] \quad (\text{E.5})$$

Siendo \bar{Q}^{ab} una matriz de dimensión $(n-1) \times (n-1)$ cuyas componentes son las expresadas en la ecuación E.5, se trata de una matriz igual a su recíproca, $\bar{Q}^{ab} = \bar{Q}^{ba}$.

Acudiendo a la primera ley de Fick, introducimos el coeficiente de difusión en forma matricial \bar{D}^{ax} y \bar{D}^{bx} mediante las expresiones E.6 y E.7, cada una relativa a la velocidad media de un parámetro a y otro b respectivamente, donde x es un parámetro que determina la composición de la disolución (por ejemplo la concentración).

$$\vec{J}^a = -\bar{D}^{ax} \cdot \vec{\nabla} \vec{x} \quad (\text{E.6})$$

$$\vec{J}^b = -\bar{D}^{bx} \cdot \vec{\nabla} \vec{x} \quad (\text{E.7})$$

Comparando estas dos últimas expresiones con la ecuación E.4 llegamos a:

$$\bar{D}^{ax} = \bar{Q}^{ab} \cdot \bar{D}^{bx} \quad (\text{E.8})$$

Ahora comparando las expresiones E.6 y E.1 y sustituyendo \bar{D}^{ax} gracias a la ecuación E.8:

$$\bar{L}^a \cdot \vec{X}^a = -\bar{D}^{ax} \cdot \vec{\nabla} \vec{x} = -\bar{Q}^{ab} \cdot \bar{D}^{bx} \cdot \vec{\nabla} \vec{x} \quad (\text{E.9})$$

Finalmente sustituimos \vec{X}^a mediante la expresión E.2 tenemos:

$$\begin{aligned} \bar{L}^a \cdot \vec{X}^a &= -\bar{Q}^{ab} \cdot \bar{D}^{bx} \cdot \vec{\nabla} \vec{x} \\ \bar{L}^a \cdot \left(-\bar{Z}^a \frac{\vec{\nabla} \bar{\mu}}{T} \right) &= -\bar{Q}^{ab} \cdot \bar{D}^{bx} \cdot \vec{\nabla} \vec{x} \\ \bar{D}^{bx} &= \frac{1}{T} \bar{Q}^{ba} \cdot \bar{L}^a \cdot \bar{Z}^a \frac{\vec{\nabla} \bar{\mu}}{\vec{\nabla} \vec{x}} = \frac{1}{T} \bar{Q}^{ba} \cdot \bar{L}^a \cdot \bar{Z}^a \frac{\partial \bar{\mu}}{\partial \vec{x}} \vec{\nabla} \vec{x} \frac{1}{\vec{\nabla} \vec{x}} \end{aligned}$$

$$\bar{D}^{bx} = \frac{1}{T} \bar{Q}^{ba} \cdot \bar{L}^a \cdot \bar{Z}^a \cdot \bar{\mu}^x \quad (\text{E.10})$$

$$\mu_{jm}^x = \left[\frac{\partial \mu_j}{\partial x_m} \right]_{p,T,x_i} \quad i \neq m \quad (\text{E.11})$$

Donde $\bar{\mu}^x$ es una matriz de dimensión $(n-1) \times (n-1)$ cuyos elementos son los indicados en la expresión E.11. Legamos así a la ecuación E.10, la cual nos relaciona directamente los coeficientes linealidad con el coeficiente de difusión.

Apéndice F

Diseño del programa

Este apartado se va a dedicar a explicar tanto la discretización de la ecuación de Lamm como el programa utilizado para la simulación de nuestro sistema el cual se va a llevar a cabo en lenguaje *C++* por medio del método de Runge-Kutta de cuarto orden. El código constará de dos subalgoritmos esenciales fuera del cuerpo principal del programa, uno destinado a calcular la derivada temporal de la concentración en cada instante t y otro dedicado a determinar la concentración en el siguiente instante $t + \delta t$ (siendo δt el paso temporal) usando el método de Runge-Kutta en base al subalgoritmo anterior.

F.1. Discretización de la ecuación de Lamm

Nuestro modelo de discretización se basa en un arco de circunferencia de ángulo α , longitud $r_b + L$ y altura z en el cual se dividen n fragmentos de volúmen entre las posiciones radiales $r = r_b$ y $r = L$. Cada fragmento de volúmen está delimitado por las posiciones radiales r_{i-1} y r_i por lo que tendrán una longitud radial $a = \frac{L}{n}$ con una concentración c_i , $0 \leq i \leq n - 1$, con la excepción del primer fragmento con concentración c_0 que estará delimitado por r_b y r_0 tal y como se representa en la figura F.1. De esta forma garantizamos que se mantenga la simetría cilíndrica.

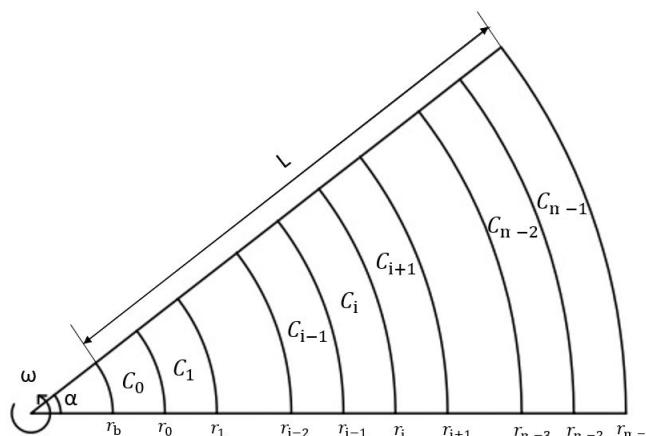


Figura F.1: Representación de la discretización de la ecuación de Lamm

Cada fragmento tiene un volúmen V_i y unas superficies internas y externas S_i^i y S_i^e respectivamente cuyos valores son los siguientes:

$$V_i = \frac{\alpha z}{2}(r_i^2 - r_{i-1}^2) = \frac{\alpha z}{2} \overbrace{(r_i - r_{i-1})}^a (r_i + r_{i-1}) \Rightarrow V_i = \frac{\alpha z a}{2}(r_i + r_{i-1}) \quad (\text{F.1})$$

$$S_i^i = \alpha z r_{i-1} \quad (\text{F.2})$$

$$S_i^e = \alpha z r_i \quad (\text{F.3})$$

Para calcular la derivada temporal de la concentración en un cierto instante se determinará el flujo de masa resultante en cada fragmento de volumen, el cual se compone de las siguientes contribuciones:

- Flujo debido a la fuerza centrífuga $\vec{J}_{FC} = s\omega^2 c \vec{r}$. Este siempre apuntará en dirección de la posición radial \hat{r} en sentido positivo, por lo que la variación de masa que experimentará un fragmento de volumen i será el flujo entrante desde el fragmento anterior a través de su superficie interna menos el flujo saliente hacia el fragmento posterior a través de su superficie externa:

$$\frac{\partial m}{\partial t}_{FC} = J_{FC,i-1} S_i^i - J_{FC,i} S_i^e = s\omega^2 c_{i-1} r_{i-1} \alpha z r_{i-1} - s\omega^2 c_i r_i \alpha z r_i$$

$$\frac{\partial m}{\partial t}_{FC} = s\omega^2 \alpha z (c_{i-1} r_{i-1}^2 - c_i r_i^2) \quad (\text{F.4})$$

- Flujo de difusión $\vec{J}_D = -D \vec{\nabla} c$. La dirección de este flujo dependerá del gradiente de concentración $\vec{\nabla} c = \frac{c_i - c_{i-1}}{a}$. Para calcular la variación de masa procedemos de la misma forma que en el caso anterior:

$$\frac{\partial m}{\partial t}_D = J_{D,i-1} S_i^i - J_{D,i} S_i^e = -D \frac{c_i - c_{i-1}}{a} \alpha z r_{i-1} + D \frac{c_{i+1} - c_i}{a} \alpha z r_i$$

$$\frac{\partial m}{\partial t}_D = -D \alpha z \left[\frac{c_i - c_{i-1}}{a} r_{i-1} - \frac{c_{i+1} - c_i}{a} r_i \right] \quad (\text{F.5})$$

De esta forma la variación de masa total será:

$$\frac{\partial m}{\partial t} = s\omega^2 \alpha z (c_{i-1} r_{i-1}^2 - c_i r_i^2) - D \alpha z \left[\frac{c_i - c_{i-1}}{a} r_{i-1} - \frac{c_{i+1} - c_i}{a} r_i \right] \quad (\text{F.6})$$

Finalmente para hallar la variación de concentración dividimos la expresión F.6 por el volumen V_i introduciendo la ecuación F.1:

$$\frac{\partial c}{\partial t} = \frac{s\omega^2 (c_{i-1} r_{i-1}^2 - c_i r_i^2)}{\frac{a}{2}(r_i + r_{i-1})} - \frac{D \left[\frac{c_i - c_{i-1}}{a} r_{i-1} - \frac{c_{i+1} - c_i}{a} r_i \right]}{\frac{a}{2}(r_i + r_{i-1})} \quad (\text{F.7})$$

Así, hemos conseguido una ecuación que expresa la variación temporal de la concentración discretizada para cada fragmento de volumen en los que se divide el sistema. También podemos escribir la ecuación F.7 de la siguiente manera:

$$\frac{\partial c}{\partial t} = \frac{s\omega^2}{a} \frac{(c_{i-1}r_{i-1}^2 - c_ir_i^2)}{r'_i} - \frac{D}{a^2} \frac{[(c_i - c_{i-1})r_{i-1} - (c_{i+1} - c_i)r_i]}{r'_i} \quad (\text{F.8})$$

$$r'_i = \frac{r_i + r_{i-1}}{2} \quad (\text{F.9})$$

F.2. Estructura del programa

En el cuerpo principal del programa se definirán los vectores (Arrays) $c[n]$ y $r[n]$ formados por n componentes como la concentración y la posición respectivamente, de tal forma que $c[i]$ se corresponde con la concentración del fragmento de volumen delimitado por las posiciones $r[i-1]$ y $r[i]$ con $0 \leq i \leq n - 1$, además de todos los parámetros necesarios para la simulación, algunos de los cuales se aportarán posteriormente a los subalgoritmos, estos son los siguientes:

- Posición inicial del tubo r_b
- Longitud del tubo L
- Velocidad angular del tubo ω
- Coeficiente de sedimentación s
- Coeficiente de difusión D
- Número de componentes del vector concentración n
- Concentración inicial c_0 tal que $c[i] = c_0$ con $0 \leq i \leq n - 1$
- Paso espacial a
- Paso temporal δT
- Tiempo real de simulación $tMedida$

Se definirán una serie de subalgoritmos, dos de ellos son fundamentales para el desarrollo del código, estos son el subalgoritmo "derivada", que nos calculará la derivada de la concentración para cada una de sus componentes, y el otro es el subalgoritmo "rungeKutta", este utilizará al primero y a otros subalgoritmos secundarios para calcular los valores de la concentración en el instante siguiente por el método de Runge-Kutta. En el algoritmo principal nos limitaremos a definir y dar valores a los distintos parámetros e imprimirlas en pantalla, y a escribir el bucle principal del programa en el cual basaremos la evolución temporal de los valores de la concentración. Tanto los subalgoritmos como el algoritmo principal se explican por separado en los apartados siguientes.

F.2.1. Subalgoritmos secundarios

Antes de empezar a desarrollar los dos subalgoritmos que conforman el grueso del programa se han definido una serie de algoritmos que nos facilitan la escritura del código, son los siguientes:

- "muestraVector", tiene como parámetros de entrada un vector $c[n]$ y un entero n que determina la longitud del vector c , este subalgoritmo se encarga de escribir en pantalla el valor de los parámetros de c .

```

void muestraVector (const double * c, int n){
    for (int i=0; i<n; i++) printf ("%g ", c[i]);
}

```

Figura F.2: Subalgoritmo "muestraVector"

- "producto", tiene como parámetros de entrada un número real x , un vector $v[n]$ y un entero n que determina la longitud del vector v , lo que hace este subalgoritmo es multiplicar los valores del vector v por el número x .

```

void producto (double x, double *v, int n){
    for (int i=0; i< n; i++) v[i]*=x;
}

```

Figura F.3: Subalgoritmo "producto"

- "suma", tiene como parámetros de entrada tres vectores; $v[n]$, $w[n]$ y $result[n]$; y un entero n que determina la longitud de los tres vectores, el subalgoritmo otorga al vector $result$ la suma de los vectores v y w tal que $result[i] = v[i] + w[i]$ con $0 \leq i \leq n - 1$.

```

void suma (const double *v, const double *w, int n, double *result) {
    for (int i=0; i<n; i++) result[i] = v[i] + w[i];
}

```

Figura F.4: Subalgoritmo "suma"

- "sumaProd", tiene como parámetros de entrada tres vectores; $v[n]$, $w[n]$ y $result[n]$; un número real x y un entero n que determina la longitud de los vectores, el subalgoritmo otorga al vector $result$ el producto de w y x mas el vector v tal que $result[i] = v[i] + xw[i]$ con $0 \leq i \leq n - 1$.
- "inicializacion", recibe como parámetros de entrada la concentración inicial c_0 (real), la posición inicial del tubo r_b (real), la longitud del tubo L (real), el número de componentes del vector concentración n , la dirección de la variable a (puntero) y los vectores concentración y posición $c[n]$ y $r[n]$ respectivamente. Lo que hace este subalgoritmo es otorgar al vector c un mismo valor c_0 (concentración inicial, constante) para todas sus componentes y otorgar al vector r los valores de la posición radial de la superficie externa de cada fragmento de volúmen.

```

void sumaProd (const double *v, const double *w, int n, double x, double *result) {
    for (int i=0; i<n; i++) result[i] = v[i] + x*w[i];
}

```

Figura F.5: Subalgoritmo "sumaProd"

```

void inicializacion (double c0, double rb, double L, double n, double s, double *c, double *r) {
    a=L/n;
    for (int i=0; i< n; i++) {
        c[i]=c0;
        r[i] = rb + (i+1)*a;
    }
}

```

Figura F.6: Subalgoritmo "inicializacion"

F.2.2. Subalgoritmo "derivada"

Este subalgoritmo se encarga de calcular un vector $\Delta C[n]$ que contiene los valores de la derivada de la concentración para cada fragmento de volúmen en un instante t . Los parámetros de entrada que recibe son los siguientes:

- Vector concentración en un instante t , $c[n]$
- Vector posición $r[n]$
- Número de componentes del vector concentración n
- Paso espacial a
- Coeficiente de sedimentación s
- Coeficiente de difusión D
- Velocidad angular del tubo ω
- Posición inicial del tubo r_b
- Vector derivada de la concentración $\Delta C[n]$

Los valores que posea este último vector antes de invocar el subalgoritmo son irrelevantes ya que serán calculados nuevamente. Primero se definen dentro del algoritmo un vector $j[n]$ cuyas componentes $j[i] = s\omega^2 c[i] r[i] - D(c[i+1] - c[i])/a$, se corresponden con el flujo que atraviesa la superficie externa del fragmento de volúmen i además de la variable r'_i definida en la ecuación (9). Ahora simplemente se calculan todas las componentes de $\Delta C[n]$ por igual excepto para los extremos de la siguiente manera:

- Extremo inicial $i = 0$, solo existe flujo en la superficie externa:

$$r'_i = \frac{r[0] + r_b}{2} \quad (F.10)$$

$$\Delta C[0] = -\frac{j[0]r[0]}{ar'_i} \quad (F.11)$$

- Fragmentos intermedios $1 \leq i \leq n - 2$, existe flujo en las superficies externa e interna:

$$r'_i = \frac{r[i] + r[i - 1]}{2} \quad (\text{F.12})$$

$$\Delta C[0] = -\frac{j[i - 1]r[i - 1] - j[i]r[i]}{ar'_i} \quad (\text{F.13})$$

- Extremo final $i = n - 1$, solo existe flujo en la superficie interna:

$$r'_i = \frac{r[n - 1] + r[n - 2]}{2} \quad (\text{F.14})$$

$$\Delta C[0] = -\frac{j[n - 2]r[n - 2]}{ar'_i} \quad (\text{F.15})$$

De esta forma ya quedan determinados los n valores del vector ΔC , los cuales se emplearán para determinar la concentración en el instante siguiente mediante otro subalgoritmo.

```

void derivada (const double *c, const double *r, int n,
               double a, double D, double s, double w, double rb,
               double *DeltaC) {
    double j[n-1], rPrima;
    for (int i=0; i< n-1; i++) {
        //**Flujo de difusión en r[i]*/
        j[i] = s*w*w*r[i]* c[i] - D * (c[i+1]-c[i])/a;
    }
    //printf ("\t\t j: "); muestraVector (j, n-1); printf ("\n");

    rPrima = (rb+r[0])/2;
    DeltaC[0] = -j[0]*r[0]/rPrima /a;

    for (int i=1; i<n-1;i++) {
        rPrima = (r[i]+r[i-1])/2;
        DeltaC[i] = (j[i-1]*r[i-1] - j[i]*r[i])/rPrima/a;
    }

    rPrima = (r[n-1]+r[n-2])/2;
    DeltaC[n-1] = j[n-2]*r[n-2]/rPrima/a;
}

```

Figura F.7: Subalgoritmo "derivada"

F.2.3. Subalgoritmo "rungeKutta"

Este subalgoritmo cambia los valores del vector concentración $c[n]$ en un instante t por la concentración en el instante posterior usando el método de Runge-Kutta de cuarto orden. Primero explicaremos en qué se basa este método. Definimos una función $y' = f(x_i, y_i)$ con $y(x_0) = y_0$, siendo $y(x)$ la función que queremos resolver numéricamente tenemos:

$$y(x + h) = \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (\text{F.16})$$

$$\begin{aligned}
k_1 &= f(x, y) \\
k_2 &= f\left(x + \frac{h}{2}, y + \frac{k_1 h}{2}\right) \\
k_3 &= f\left(x + \frac{h}{2}, y + \frac{k_2 h}{2}\right) \\
k_4 &= f(x + h, y + k_3 h)
\end{aligned}$$

Siendo h la longitud del intervalo que aumenta la variable x . Este método tiene como ventaja principal su error, que es de cuarto orden $O(h^4)$ (muy pequeño), lo que nos permitirá aportar valores muy reducidos a h para obtener una simulación satisfactoria.

Para nuestro caso identificaremos la concentración c con la función y , dependiente del tiempo t ($x = t$), de forma que y' será la derivada temporal de la concentración y $h = deltaT$ el paso temporal; $y' = f(x, y) = \frac{\partial c}{\partial t}(t, c)$.

Volviendo al subalgoritmo, sus parámetros de entrada son los siguientes:

- Vector concentración en un instante t , $c[n]$
- Vector posición $r[n]$
- Número de componentes del vector concentración n
- Paso espacial a
- Coeficiente de sedimentación s
- Coeficiente de difusión D
- Velocidad angular del tubo ω
- Posición inicial del tubo r_b
- Paso temporal $deltaT$

Dentro del subalgoritmo se definen 5 vectores $k_1[n]$, $k_2[n]$ $k_3[n]$ $k_4[n]$ y $aux[n]$, los 4 primeros servirán para almacenar los valores las k 's definidas en el método de Runge-Kutta definido anteriormente para cada componente de la concentración c , el vector aux se utilizará como variable auxiliar para calcular las distintas k 's.

Primero invocamos el subalgoritmo "derivada" para calcular $k_1[n]$ con el vector c como entrada, a continuación invocamos los subalgoritmos "producto" para multiplicar $k_1[n]$ por $deltaT$ y "sumaProd" para otorgar a $aux[n]$ los valores de $c + k_1/2$ (habiéndose multiplicado ya k_1 por $deltaT$) de tal forma que volvemos a invocar "derivada" con el vector aux como entrada para la concentración y calcular así k_2 . Repetimos este proceso para k_3 y para k_4 (cada una calculada con la k anterior) con la única diferencia de que al invocar "sumaProd" otorgamos a aux los valores $c + k_2/2$ y $c + k_3$ respectivamente.

Una vez calculadas los cuatro vectores de k_1 , k_2 , k_3 y k_4 únicamente falta calcular la ecuación (16) (para todas las componentes de c), para ello invocamos el subalgoritmo "sumaProd" para otorgar a aux los valores de $c + k_1/6$ (recordamos que todos los valores de las k 's ya han sido

multiplicados por $deltaT$), volvemos a invocar este subalgoritmo tres veces mas para sumarle al vector aux los valores $k_2/3$, $k_3/3$ y $k_4/6$. La última vez que invocamos `sumaProd` lo hacemos con el vector $c[n]$ como salida.

De esta forma el subalgoritmo devuelve los valores de la concentración para el instante siguiente en el mismo vector de la concentración en el instante anterior.

```

void rungeKutta (double *c, const double *r, int n,
                 double a, double D, double s, double w, double rb,
                 double deltaT){
    double k1[n], k2[n], k3[n], k4[n], aux[n];

    derivada (c, r, n, a, D, s, w, rb, k1);
    producto (deltaT, k1, n);
    //printf ("\t\t k1: "); muestraVector (k1, n); printf ("\n");

    sumaProd(c, k1, n, 0.5, aux);
    derivada (aux, r, n, a, D, s, w, rb, k2);
    producto (deltaT, k2, n);

    sumaProd(c, k2, n, 0.5, aux);
    derivada (aux, r, n, a, D, s, w, rb, k3);
    producto (deltaT, k3, n);

    sumaProd(c, k3, n, 1, aux);
    derivada (aux, r, n, a, D, s, w, rb, k4);
    producto (deltaT, k4, n);

    sumaProd (c, k1, n, 1.0/6, aux);
    sumaProd (aux, k2, n, 1.0/3, aux);
    sumaProd (aux, k3, n, 1.0/3, aux);
    sumaProd (aux, k4, n, 1.0/6, c);
}

```

Figura F.8: Subalgoritmo rungeKutta”

F.2.4. Algoritmo principal

Como ya se ha mencionado anteriormente en el cuerpo principal se definirán todas las variables que usarán para invocar los subalgoritmos. Invocando el subalgoritmo ”inicializacion” proporcionamos al vector concentración c el valor c_0 a todas sus componentes y también determinamos los valores del vector posición r en función de L , r_b y n . Además se imprimirá en pantalla los valores escogidos para la longitud del tubo L , la velocidad angular ω , el coeficiente de sedimentación s y el coeficiente de difusión D , junto con el paso espacial calculado a . También se imprimirá en pantalla el tiempo que se quiere simular $tMedida$, el paso espacial $deltaT$ y el número de pasos temporales que ejecutará el programa.

Para que el programa calcule la evolución temporal de la concentración escribimos un bucle en el cual avanza el tiempo t , en cada ejecución del bucle t suma $deltaT$ hasta llegar a $tMedida$. En el cuerpo del bucle únicamente se invoca el subalgoritmo "rungeKutta" que proporcionará la concentración en el instante siguiente para sus n componentes.

Finalmente se imprimirá en pantalla los los valores de la concentración obtenidos en el instante final con sus respectivas posiciones radiales. También se escribirán en un fichero para su posterior estudio.

```

int main() {
    FILE *f;
    f = fopen("C:/Users/jonas/Documents/Fisica/Fisica 5/TFG/concenk.txt", "w");
    double c0, rb=0.0608, L=0.1085-0.0608, a;
    int n=1000;
    double c[n], r[n];
    inicializacion (l, rb, L, n, a, c, r);
    //printf ("Concentraciones iniciales\t"); muestraVector (c, n); printf ("\n");
    //printf ("r:\t"); muestraVector (r, n); printf ("\n");

    double w = 55000 *2*M_PI / 60, s=1E-12, D=1E-9;
    printf ("L=%g m\t a=%g m \t w=%g rad s^-1 \t s=%g \t D=%g\n", L, a, w, s, D);

    double deltaT = 0.01;
    double t;
    double tMedida=3600;
    //double tMedida=log(0.09658/rb)/w/w/s;

    printf ("%g %g %g\n", tMedida, deltaT, tMedida/deltaT);
    printf ("t \t r \t c \n");
    for (t=0; t<(tMedida); t+= deltaT){
        rungeKutta(c, r, n, a, D, s, w, rb, deltaT);
        //printf ("%g\t",t); muestraVector (c, n); printf ("\n");
    }
    for (int i=0; i< n; i++) printf ("%g\t %g %g\n", t, r[i], c[i]);
    for (int i=0; i< n; i++) fprintf (f,"%g\t%g\n", r[i], c[i]);
    //fprintf(f,"%f\t%f",deltaT,c[14]);

    return 0;
}

```

Figura F.9: Algoritmo principal