

**El intercambio de energía desde la teoría del
coarse-graining
(Máster en física de los sistemas complejos)
22 de febrero de 2012**

Marc Meléndez Schofield
Director: Pep Español

Índice general

Introducción	5
Termodinámica y <i>coarse-graining</i>	5
Las herramientas básicas	6
Capítulo 1. Evolución temporal de las distribuciones de probabilidad	9
1.1. La dinámica de sistemas hamiltonianos clásicos en interacción	9
1.2. Distribuciones de probabilidad en el espacio de macroestados	10
1.3. Operadores de proyección	13
1.4. Ecuación de evolución para $P(e, t)$	14
1.5. Equilibrio macroscópico	16
Capítulo 2. La aproximación markoviana	19
2.1. Aproximación mediante la distribución relevante	20
2.2. Variables lentas	22
2.3. Energías potenciales independientes de los momentos conjugados	25
2.4. Intercambio de energías entre gases ideales	27
Capítulo 3. Procesos no markovianos	33
3.1. El sistema Fermi-Pasta-Ulam	34
3.2. Simular el equilibrio térmico	35
3.3. Intercambio de energía	36
3.4. El trabajo macroscópico	38
Bibliografía	41
Apéndice A. Demostraciones	43
Apéndice B. Código fuente	47
Gas de esferas duras	47
Ecuación de Fokker-Planck	57
Cadenas de Fermi-Pasta-Ulam	59

Introducción



Unas gotas de tinta que caen en un vaso de agua se abren dibujando una figura intrincada a medida que se mezclan¹. Podemos reproducir el efecto en el sentido de que otras gotas de tinta en agua dejarán figuras igual de intrincadas, pero los patrones concretos serán siempre diferentes. Este sencillo experimento es increíblemente sensible a las condiciones iniciales, de modo que hasta las imperceptibles variaciones microscópicas se amplifican, trepando por las escalas, hasta que son evidentes a simple vista. Sin embargo, a pesar de la diversidad de siluetas, el final es siempre el mismo: al cabo de un rato encontramos que la tinta se ha extendido de forma homogénea por el agua y que la superficie está en reposo. El estado final sólo depende de los volúmenes de agua y tinta.

La materia tiene esta propiedad general de tender hacia *estados de equilibrio*, que se pueden caracterizar con unos pocos parámetros completamente independientes de la historia pasada del sistema en cuestión. Ahora bien, el tiempo que tarda un sistema en alcanzar el equilibrio y “olvidar” las peculiaridades de su evolución es muy variable. Dejamos de sentir una corriente de aire en el momento en el que se cierra la puerta de una habitación; la presión parece estabilizarse al instante. En cambio, se dice que los diamantes son para siempre, aunque la disposición de los átomos sería más estable si adoptaran la estructura del grafito. Sea rápidamente o con lentitud geológica, pero inexorable, el olvido se acaba extendiendo por la materia dejando un paisaje inhóspito, homogéneo, apagado y frío.

Termodinámica y *coarse-graining*

En palabras de Herbert B. Callen, la termodinámica clásica se encarga sólo de determinar el estado de equilibrio que resulta finalmente tras la eliminación de ciertas restricciones internas en un sistema compuesto aislado. En el caso al que está dedicado este trabajo, el de subsistemas previamente aislados que se ponen en contacto, predice que finalmente llegarán a un equilibrio mecánico y térmico, es decir, que se equilibrarán las fuerzas que ejercen unos sistemas sobre otros y que alcanzarán las mismas temperaturas. Esta descripción es dinámica sólo en el sentido de que aspira a predecir el estado final de la evolución de un sistema a partir del estado inicial, pero no describe el *proceso* dinámico que los conecta.

¹Fotografía: *Ink swirls*, por Graham Richardson (Algunos derechos reservados, CC-BY-NC 2.0), <http://www.flickr.com/photos/didbygraham/3503800137/in/photostream/>.

En principio, las leyes de la mecánica (clásica o cuántica) determinan completamente el comportamiento temporal de un conjunto de átomos, pero la complejidad de la dinámica y el número de variables involucradas hacen que, en la práctica, sea imposible deducir cómo llegan al equilibrio más que en casos contados. Además, incluso si tuviéramos un listado completo de la dependencia temporal de los valores de cada una de las variables microscópicas, es muy probable que esto no sirviera para iluminar las características esenciales del proceso, porque el equilibrio térmico es un fenómeno *emergente*: un número inmenso de partículas, átomos o moléculas, se reparten de manera equitativa la energía entre todas las formas posibles de almacenarla. Por supuesto, siempre hay cierta probabilidad de que una partícula concreta tenga una energía dada por encima o por debajo de la media, pero esta probabilidad no se mantiene constante al aumentar el sistema, sino que *disminuye* con el incremento del número de partículas. Es decir, la energía se reparte de manera más eficiente cuanto mayor es el número de átomos. Lo que es todavía más sorprendente es que, a pesar de la gran variedad de interacciones entre átomos y moléculas, la tendencia al equilibrio térmico parece una propiedad general de la materia, con lo que, aunque es un fenómeno que surge como consecuencia del intercambio de energía entre partículas, no depende (como fenómeno) de los detalles de este intercambio.

La termodinámica ha demostrado la utilidad de caracterizar los estados de equilibrio, pero frecuentemente estamos más interesados en estudiar los procesos que llevan a ellos (al fin y al cabo, nuestras vidas se desarrollan por completo enmarcadas en estos procesos). Si conocemos de antemano las variables macroscópicas relevantes, la técnica conocida como *coarse-graining* nos puede suministrar las ecuaciones diferenciales para la evolución temporal del sistema considerada desde el punto de vista macroscópico. El formalismo de esta técnica se desarrolla en el primer capítulo para el caso del intercambio de energía entre sistemas previamente aislados que se ponen en contacto. Los siguientes capítulos aplican los resultados teóricos a casos relativamente sencillos. Por lo tanto, el capítulo siguiente estará dedicado a un recorrido a vista de pájaro desde las interacciones microscópicas hasta el concepto macroscópico de equilibrio térmico. Veremos que, si la distribución de probabilidad sobre los microestados se mezcla en el espacio de fases, el sistema, al igual que pasaba con la tinta, acaba olvidando su evolución anterior.

El segundo capítulo parte de la ecuación exacta para el intercambio energético macroscópico y examina el caso en el que el sistema se olvida muy rápidamente de su pasado, de modo que la evolución temporal de las energías depende en cada instante sólo del macroestado presente. El capítulo final examina analíticamente y numéricamente un sistema ideal en el que no se cumple la aproximación considerada en el capítulo que lo precede.

Las herramientas básicas

Ya he mencionado que, en la práctica, es imposible determinar el estado de un sistema que tiene muchos grados de libertad. Para un gas ideal, por ejemplo, implica saber las posiciones y velocidades de cada uno de sus átomos. Sin embargo, en un laboratorio normalmente es posible describir físicamente un gas con los valores de unas pocas funciones del estado: la presión, el volumen, la temperatura y la densidad. Si hemos determinado el valor de todas estas funciones, diremos que conocemos el *macroestado* del gas, mientras que a la teórica especificación de cada una de las posiciones y velocidades le llamaremos el *microestado*. Evidentemente, el conocimiento del macroestado no nos permite inferir el microestado, debido a que en general habrá muchos microestados a los que les corresponde el mismo macroestado.

En los capítulos que siguen, el microestado de nuestro sistema estará representado por una variable z . Si conocemos la probabilidad (o densidad de probabilidad) $\rho(z, t)$ de que el sistema se encuentre en el microestado z en un tiempo t para todos los valores posibles de z , entonces podemos calcular la probabilidad (o densidad de probabilidad) $P(x, t)$ de que cualquier función $X(z)$ tome un valor dado x en el tiempo t . No tenemos más que sumar (o integrar) la función $\rho(z, t)$ sobre todos los estados tales que $X(z) = x$, es decir,

$$P(x, t) = \begin{cases} \sum_{X(z)=x} \rho(z, t), & \text{si } z \text{ es una variable discreta,} \\ \int_S \rho(z, t) ds, & \text{si } z \text{ es una variable continua.} \end{cases}$$

S representa aquí el conjunto de puntos $S = \{z : X(z) = x\}$. Evidentemente, $P(x, t)$ será una densidad o una probabilidad dependiendo de si los valores posibles de x son continuos o discretos. En los capítulos siguientes, escribiremos la integral sobre S de manera más conveniente, como sigue:

$$\int_S \rho(z, t) ds = \int \rho(z, t) \delta(X(z) - x) dz,$$

donde δ es la delta de Dirac y se sobreentiende que la integral se extiende a todos los valores posibles de z .

Si tuviéramos una expresión analítica para $\rho(z, t)$, en principio podríamos calcular las probabilidades $P(x, t)$ llevando a cabo las integrales o sumas pertinentes, pero normalmente sólo disponemos de una ley para la evolución temporal de ρ , como, por ejemplo,

$$\frac{\partial \rho(z, t)}{\partial t} = \mathcal{L}\rho(z, t),$$

donde \mathcal{L} representa algún operador lineal. Dado que sólo estamos interesados en la evolución de los valores de las funciones X , lo mejor sería intentar obtener una ecuación de evolución para $P(x, t)$ a partir de la de la densidad de probabilidad ρ . Derivando la definición de P , y utilizando la ecuación diferencial para ρ , obtenemos

$$\begin{aligned} \frac{\partial P(x, t)}{\partial t} &= \frac{\partial}{\partial t} \int \rho(z, t) \delta(X(z) - x) dz \\ &= \int \frac{\partial \rho(z)}{\partial t} \delta(X(z) - x) dz \\ &= \int \mathcal{L}\rho(z) \delta(X(z) - x) dz. \end{aligned}$$

Si pudiéramos expresar la integral de la derecha en términos de la función P , tendríamos una descripción matemática precisa de la dinámica de las funciones X para la que no sería necesario tener en cuenta todos los grados de libertad microscópicos.

La estrategia expuesta en el primer capítulo se conoce como la técnica de los operadores de proyección. Consiste en obtener una ecuación de evolución utilizando una *distribución relevante* $\bar{\rho}$, que se define como una distribución que es equivalente a ρ desde el punto de vista macroscópico, es decir, para el caso de variables continuas,

$$P(x, t) = \int_S \rho(z, t) ds = \int_S \bar{\rho}(z, t) ds.$$

y en el caso discreto,

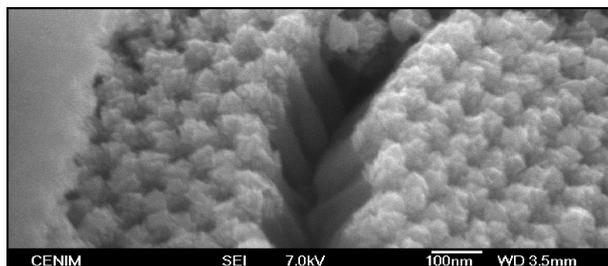
$$P(x, t) = \sum_{X(z)=x} \rho(z, t) = \sum_{X(z)=x} \bar{\rho}(z, t).$$

La ventaja de utilizar una distribución relevante se basa en la posibilidad de definirla como una función de las funciones X que determinan el macroestado,

$\bar{\rho} = \bar{\rho}(X(z))$. Como veremos más adelante, nuestra elección de $\bar{\rho}$ resulta ser una generalización de la colectividad microcanónica a estados fuera del equilibrio.

Si pensamos en el intercambio energético entre sistemas con muchos grados de libertad microscópicos, no suele ser difícil imaginar las características generales del proceso, pues se trata simplemente de una relajación al estado de equilibrio. Cuando mezclamos café caliente con leche fría, nuestro café con leche evoluciona hacia una mezcla homogénea en reposo en la que la temperatura del café se equilibra con la de la leche. Pero, ¿por qué y en qué casos se produce este equilibrio? El capítulo que sigue está dedicado a esbozar una respuesta a estas preguntas.

Evolución temporal de las distribuciones de probabilidad



La termodinámica parte de la idea de que si se ponen en contacto dos sistemas previamente aislados, se desencadena un flujo de energía irreversible desde el cuerpo más caliente al más frío hasta que se igualan las temperaturas. Se dice entonces que los sistemas están en equilibrio térmico. Pero, a nivel microscópico¹, los sistemas están formados por partículas (de las que no se puede decir que estén frías o calientes) que interactúan de acuerdo con leyes reversibles, y las idas y venidas de energía entre los sistemas no se detienen nunca. Este capítulo explica esquemáticamente la relación entre la perspectiva microscópica y la evolución macroscópica de las energías.

1.1. La dinámica de sistemas hamiltonianos clásicos en interacción

Siempre es posible imaginar la interacción entre varios sistemas suponiendo que están englobados en un sistema aislado mayor. En este sistema total, numeraremos los subsistemas de interés con el índice $j = 1, 2, \dots, n$. Diremos que el subsistema j -ésimo tiene N_j grados de libertad. El número de grados de libertad del sistema total será $N = \sum_{j=1}^n N_j$.

Si no existiera interacción entre los subsistemas, sus dinámicas estarían determinadas por sendos hamiltonianos $H^j(q_1^j, p_1^j, \dots, q_{N_j}^j, p_{N_j}^j)$, donde q_i^j representa la coordenada generalizada i -ésima del subsistema j -ésimo, y p_i^j el correspondiente momento conjugado. Al tener en cuenta dicha interacción, se debe añadir a la suma de los hamiltonianos H^j un hamiltoniano de interacción $H^{int.}$ para obtener el hamiltoniano total del sistema

(1.1.1)

$$\begin{aligned} H(z) &= \sum_{j=1}^n H^j(q_1^j, p_1^j, \dots, q_{N_j}^j, p_{N_j}^j) + H^{int.}(q_1^1, p_1^1, \dots, q_i^j, p_i^j, \dots, q_{N_n}^n, p_{N_n}^n) \\ &= E. \end{aligned}$$

¹Fotografía: *Nickel Oxide growth (by Sputtering RF Magnetron) on porous alumina membrane*, por Guillermo Domínguez Cañizares (2011).

La variable z representa el vector de posiciones y momentos

$$z = \left(q_1^1, p_1^1, \dots, q_i^j, p_i^j, \dots, q_{N_n}^n, p_{N_n}^n \right),$$

y E es la energía total del sistema. Nótese que los potenciales que dependen de grados de libertad del mismo subsistema están ya incluidos en los términos H^j , por lo que el término H^{int} añade solamente los potenciales debidos a interacciones entre grados de libertad que pertenecen a subsistemas diferentes.

Cualquier trayectoria del sistema en el espacio de fases,

$$z(t) = \left(q_1^1(t), p_1^1(t), \dots, q_i^j(t), p_i^j(t), \dots, q_{N_n}^n(t), p_{N_n}^n(t) \right),$$

satisface las ecuaciones de Hamilton,

$$(1.1.2) \quad \begin{cases} \dot{q}_i^j &= \frac{\partial H}{\partial p_i^j}, \\ \dot{p}_i^j &= -\frac{\partial H}{\partial q_i^j}. \end{cases}$$

Una distribución de probabilidad ρ definida sobre los puntos z fluye por el espacio de fases sin comprimirse ni expandirse, de modo que $\rho(z(t), t) = \rho(z(0), 0)$. Esto implica que ρ debe cumplir la condición impuesta por la ecuación de Liouville,

$$(1.1.3) \quad \frac{\partial \rho(z, t)}{\partial t} = \{H, \rho\}.$$

Llamaremos *distribución de equilibrio*, ρ^{eq} , a la solución estacionaria de la ecuación de Liouville, es decir,

$$(1.1.4) \quad \frac{\partial \rho^{eq}(z)}{\partial t} = \{H, \rho^{eq}\} = 0.$$

La función $\rho^{eq}(z)$ debe ser siempre mayor o igual que cero,

$$(1.1.5) \quad \rho^{eq}(z) \geq 0,$$

debe estar normalizada²,

$$(1.1.6) \quad \int \rho^{eq}(z) dz = 1,$$

y debe ser igual a cero siempre que $H(z) \neq E$,

$$(1.1.7) \quad H(z) \neq E \implies \rho^{eq}(z) = 0.$$

Se puede comprobar fácilmente que la *distribución microcanónica* satisface las ecuaciones (1.1.4)-(1.1.7),

$$(1.1.8) \quad \rho^{eq}(z) = \frac{\delta(H(z) - E)}{\int \delta(H(z) - E) dz}.$$

1.2. Distribuciones de probabilidad en el espacio de macroestados

En muchos casos, la evolución temporal de la distribución de probabilidad $\rho(z, t)$ en el espacio de fases es muy compleja. Aunque, en principio, podríamos intentar resolver la ecuación de Liouville (1.1.3) y después utilizar la solución para determinar las probabilidades correspondientes a cada uno de los valores de las funciones $H^j(z)$, este método se vuelve inabordable en la práctica cuando z representa un número grande de grados de libertad.

Si quisiéramos resolver numéricamente las ecuaciones (y normalmente es esta la única solución disponible), reduciríamos de manera drástica el tiempo de computación si tuviéramos una ecuación cerrada para la probabilidad de que cada

²Cuando no se indiquen los límites de integración, se supondrá que la integral se extiende a todos los valores de las variables correspondientes. La integral $\int \dots dz$, por ejemplo, se extiende a todo el espacio de fases.

subsistema tenga una energía dada. Desde este punto de vista *macroscópico*, el estado del sistema estará definido por las energías e_j de cada uno de los subsistemas. El espacio de macroestados será el conjunto de vectores

$$e = (e_1, \dots, e_n) = (H^1(z), \dots, H^n(z)),$$

con la restricción adicional

$$\sum_{j=1}^n e_j \leq E,$$

debido a la conservación de la energía del sistema total. La expresión anterior se convierte en una identidad cuando $H^{int.} = 0$.

La densidad de probabilidad $P(e, t)$ para el macroestado e en el tiempo t es igual a la integral de $\rho(z, t)$ extendida a todos los puntos del espacio de fases tales que $H^j(z) = e_j$, es decir

$$(1.2.1) \quad P(e, t) = \int \rho(z, t) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz.$$

Es evidente que el conocimiento de P no nos permite inferir la distribución de probabilidad ρ sobre el espacio de fases, pues habrá muchas distribuciones microscópicas compatibles con los valores de P . Por ejemplo, la función $\tilde{\rho} = \rho + \delta\rho$, donde $\delta\rho$ es cualquier función que cumpla la propiedad

$$\int \delta\rho(z, t) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz = 0,$$

produce la misma distribución de probabilidad macroscópica que ρ . No obstante, dada una distribución $P(e, t)$, es posible determinar cuál de las distribuciones microscópicas compatibles es *menos sesgada*, hallando la que maximiza el funcional de la entropía

$$(1.2.2) \quad S[\rho] \equiv -k_B \int \rho(z, t) \ln \left(\frac{\rho(z, t)}{\rho^{eq.}(z)} \right) dz,$$

restringiéndonos a las funciones *normalizadas* compatibles con la distribución $P(e, t)$,

$$(1.2.3) \quad \int \rho(z, t) dz = 1,$$

$$(1.2.4) \quad \int \rho(z, t) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz = P(e, t).$$

Utilizando el método de los multiplicadores de Lagrange, encontramos que debemos maximizar el funcional

$$I[\rho, \lambda, \mu] = S[\rho] + \int \lambda(e) \left(P(e, t) - \int \rho(z, t) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz \right) de \\ + \mu \left(\int \rho(z, t) dz - 1 \right).$$

Buscamos a continuación los valores mínimos de $I[\rho, \lambda, \mu]$, hallando los casos en los que se anulan las derivadas de I . La derivada respecto de μ y la derivada variacional

respecto de $\lambda(e)$ recuperan las restricciones sobre la distribución de probabilidad,

$$\begin{aligned}\frac{\partial}{\partial \mu} I[\rho, \lambda, \mu] = 0 &\implies \int \rho(z, t) dz = 1, \\ \frac{\delta}{\delta \lambda(e)} I[\rho, \lambda, \mu] = 0 &\implies \int \rho(z, t) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz = P(e, t).\end{aligned}$$

La derivada variacional respecto de ρ produce la ecuación

$$\begin{aligned}\frac{\delta}{\delta \rho} I[\rho, \lambda, \mu] &= -k_B \left(\ln \left(\frac{\rho(z, t)}{\rho^{eq.}(z)} \right) + 1 \right) \\ &\quad + \int \lambda(e) \left(\int \left[\prod_{j=1}^n \delta(H^j(z') - e_j) \right] \delta(z' - z) dz' \right) de + \mu \\ &= -k_B \left(\ln \left(\frac{\rho(z, t)}{\rho^{eq.}(z)} \right) + 1 \right) + \lambda(H^1(z), \dots, H^n(z)) + \mu.\end{aligned}$$

Para que se anule esta expresión, se debe cumplir que

$$(1.2.5) \quad \rho(z, t) = \rho^{eq.}(z) \exp \left\{ \frac{1}{k_B} (\lambda(H^1(z), \dots, H^n(z)) + \mu - k_B) \right\}.$$

Si introducimos (1.2.5) en la condición para $P(e, t)$ (1.2.4),

$$\begin{aligned}P(e, t) &= \int \rho(z, t) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz \\ &= \int \rho^{eq.}(z) \exp \left\{ \frac{1}{k_B} (\lambda(H^1(z), \dots, H^n(z)) + \mu - k_B) \right\} \\ &\quad \times \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz.\end{aligned}$$

Llevando a cabo la integral,

$$(1.2.6) \quad P(e, t) = \exp \left\{ \frac{1}{k_B} (\lambda(e) + \mu - k_B) \right\} \Omega(e).$$

La función $\Omega(e)$ representa la distribución de probabilidad macroscópica de equilibrio

$$(1.2.7) \quad \Omega(e) = \int \rho^{eq.}(z) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz$$

(compárese con la ecuación 1.2.1). Podemos despejar $\lambda(e)$ en la ecuación (1.2.6),

$$\lambda(e) = k_B \ln \left(\frac{P(e, t)}{\Omega(e)} \right) - \mu + k_B,$$

y sustituir el resultado en la ecuación (1.2.5) para obtener la distribución menos sesgada compatible con $P(e, t)$,

$$(1.2.8) \quad \rho(z, t) = \rho^{eq.}(z) \frac{P(H^1(z), \dots, H^n(z), t)}{\Omega(H^1(z), \dots, H^n(z))} \equiv \bar{\rho}(z, t).$$

En adelante nos referiremos a $\bar{\rho}$ como la *distribución relevante*.

Si no existe interacción entre los subsistemas ($H^{int.} = 0$), entonces la distribución relevante coincide con la distribución de equilibrio del conjunto de subsistemas

aislados entre sí, puesto que en este caso

$$(1.2.9) \quad P(H^1(z), \dots, H^n(z)) = \prod_{j=1}^n \delta(H^j(z) - e_j),$$

ya que el único macroestado posible es aquel en el que cada sistema tiene su energía de partida, y

$$(1.2.10) \quad \Omega(H^1(z), \dots, H^n(z)) = \rho^{eq.}(z) \left[\prod_{j=1}^n \int \delta(H^j(z) - e_j) dz^j \right],$$

a consecuencia de (1.2.7) y (1.1.4). El vector z^j se refiere a

$$z^j = (q_1^j, p_1^j, \dots, q_{N_j}^j, p_{N_j}^j).$$

Sustituyendo (1.2.9) y (1.2.10) en la ecuación (1.2.8), obtenemos la distribución de equilibrio para subsistemas aislados

$$(1.2.11) \quad \bar{\rho}(z, t) = \frac{\prod_{j=1}^n \delta(H^j(z) - e_j)}{\prod_{j=1}^n \int \delta(H^j(z) - e_j) dz^j},$$

que no es más que el producto de las distribuciones microcanónicas para los subsistemas, como cabía esperar. En las páginas siguientes, supondremos que la distribución inicial $\rho(z, 0)$ tiene la forma (1.2.11). Esto es equivalente a imaginar que la interacción comienza cuando $t = 0$ y se produce entre sistemas que habían estado aislados unos de otros hasta ese instante y que habían alcanzado el equilibrio por separado.

La distribución de equilibrio $\rho^{eq.}(z)$ también es una distribución relevante, porque su distribución macroscópica correspondiente es $\Omega(e)$. Aplicando (1.1.4),

$$(1.2.12) \quad \bar{\rho}^{eq.} = \rho^{eq.}(z) \frac{\Omega(H^1(z), \dots, H^n(z))}{\Omega(H^1(z), \dots, H^n(z))} = \rho^{eq.}.$$

1.3. Operadores de proyección

Si sustituimos la definición de $P(e, t)$ (1.2.1) en la definición de la distribución relevante (1.2.8), obtenemos

$$\bar{\rho}(z, t) = \rho^{eq.}(z) \int \frac{1}{\Omega(e)} \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] \left(\int \rho(z', t) \left[\prod_{j=1}^n \delta(H^j(z') - e_j) \right] dz' \right) de,$$

que puede interpretarse como el resultado de aplicar un operador lineal de proyección, que llamaremos \mathcal{P}^\dagger , sobre la distribución ρ ,

$$\mathcal{P}^\dagger \rho = \bar{\rho}.$$

El operador \mathcal{P}^\dagger quedaría entonces definido como

$$(1.3.1) \quad \mathcal{P}^\dagger \mu(z, t) = \rho^{eq.}(z) \int \frac{1}{\Omega(e)} \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] \int \mu(z', t) \left[\prod_{j=1}^n \delta(H^j(z') - e_j) \right] dz' de.$$

Se dice que \mathcal{P}^\dagger es un proyector porque es idempotente, es decir, $\mathcal{P}^\dagger \mathcal{P}^\dagger = \mathcal{P}^\dagger$.

Llamaremos a $\delta\rho \equiv \rho - \bar{\rho}$ la *parte no relevante* de la distribución. Se comprueba fácilmente que

$$\mathcal{P}^\dagger \delta\rho = 0.$$

Denotaremos mediante el signo \mathcal{Q}^\dagger el operador que proyecta la distribución ρ sobre la parte no relevante,

$$(1.3.2) \quad \mathcal{Q}^\dagger \rho(z, t) = \delta\rho(z, t) = \rho(z, t) - \bar{\rho}(z, t) = (I - \mathcal{P}^\dagger) \rho(z, t),$$

donde I representa el operador identidad

$$I\rho = \rho.$$

El operador \mathcal{Q}^\dagger también es un proyector,

$$\mathcal{Q}^\dagger \mathcal{Q}^\dagger = (I - \mathcal{P}^\dagger)(I - \mathcal{P}^\dagger) = (I - \mathcal{P}^\dagger) = \mathcal{Q}^\dagger.$$

Como $\mathcal{P}^\dagger + \mathcal{Q}^\dagger = I$, siempre podremos escribir una distribución de probabilidad ρ como suma de una distribución relevante más la parte no relevante,

$$\rho = \mathcal{P}^\dagger \rho + \mathcal{Q}^\dagger \rho = \bar{\rho} + \delta\rho.$$

1.4. Ecuación de evolución para $P(e, t)$

Hemos determinado la distribución macroscópica $P(e, t)$ y su relación con las distribuciones microscópicas $\rho(z, t)$ y $\bar{\rho}(z, t)$, pero todavía estamos lejos de nuestro objetivo, que es una ecuación de evolución cerrada para $P(e, t)$. Derivando respecto del tiempo la ecuación (1.2.1),

$$(1.4.1) \quad \frac{\partial P(e, t)}{\partial t} = \int \{H, \rho\} \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz,$$

obtenemos una ecuación de evolución, pero no es cerrada en P , sino que depende de las distribuciones sobre el espacio de fases. Sin embargo, es posible convertir la ecuación anterior en una ecuación cerrada trabajando un poco con los operadores de proyección definidos en el apartado anterior.

Si separamos ρ en suma de la distribución relevante más la parte no relevante, entonces

$$\{H, \rho\} = \{H, \bar{\rho}\} + \{H, \delta\rho\},$$

y la ecuación (1.4.1) se convierte en

$$(1.4.2) \quad \frac{\partial P(e, t)}{\partial t} = \int \{H, \bar{\rho}\} \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz + \int \{H, \delta\rho\} \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz.$$

Introduciendo la definición de $\bar{\rho}$ (1.2.8), podemos simplificar el primer corchete de Poisson,

$$\begin{aligned} \{H, \bar{\rho}\} &= \left\{ H, \rho^{eq} \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\} \\ &= \{H, \rho^{eq}\} \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} + \rho^{eq} \left\{ H, \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\} \\ &= \rho^{eq} \left\{ H, \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\}. \end{aligned}$$

El término en $\{H, \rho^{eq}\}$ se anula (1.1.4). Si ahora utilizamos la definición de nuestro hamiltoniano (1.1.1),

$$(1.4.3) \quad \begin{aligned} \{H, \bar{\rho}\} &= \sum_{j=1}^n \rho^{eq} \left\{ H^j, \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\} + \rho^{eq} \left\{ H^{int}, \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\} \\ &= \rho^{eq} \left\{ H^{int}, \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\}. \end{aligned}$$

Los corchetes con H^j se anulan porque P y Ω son funciones de las variables H^j . Además,

$$(1.4.4) \quad \{H, \bar{\rho}\} = \rho^{eq.} \left\{ H^{int.}, \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\} \\ = \sum_{j=1}^n \rho^{eq.} \{H^{int.}, H^j\} \frac{\partial}{\partial H^j} \left(\frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right),$$

que introducido en el primer sumando de la ecuación de evolución para $P(e, t)$ (1.4.2) nos da

$$(1.4.5) \quad \int \{H, \bar{\rho}\} \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz = \sum_{j=1}^n \frac{\partial}{\partial e_j} \frac{P(e, t)}{\Omega(e)} \\ \times \int \rho^{eq.}(z) \{H^{int.}, H^j\} \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dz \\ = - \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) P(e, t),$$

donde se han utilizado (1.4.4) y la definición

$$(1.4.6) \quad v_j(e) = - \frac{1}{\Omega(e)} \int \rho^{eq.}(z) \{H^{int.}, H^j\} \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dz.$$

Hemos conseguido escribir el primer sumando de (1.4.2) en función de $P(e, t)$. Para transformar el sumando que contiene el corchete $\{H, \delta\rho\}$, recurriremos a una expresión de $\delta\rho$ en función de $\bar{\rho}$. Partiremos de la ecuación de evolución temporal de $\delta\rho$,

$$\frac{\partial \delta\rho(z, t)}{\partial t} = \frac{\partial}{\partial t} \mathcal{Q}^\dagger \rho(z, t).$$

Como los proyectores \mathcal{P}^\dagger y \mathcal{Q}^\dagger son operadores lineales independientes del tiempo,

$$\frac{\partial \delta\rho(z, t)}{\partial t} = \mathcal{Q}^\dagger \{H, \rho\} = \mathcal{Q}^\dagger \{H, \bar{\rho}\} + \mathcal{Q}^\dagger \{H, \delta\rho\}.$$

La solución formal de esta ecuación es

$$\delta\rho(z, t) = \exp\{\mathcal{Q}^\dagger \mathcal{L} t\} \delta\rho(z, 0) + \int_0^t \exp\{\mathcal{Q}^\dagger \mathcal{L}(t-s)\} \mathcal{Q}^\dagger \mathcal{L} \bar{\rho}(z, s) ds,$$

donde \mathcal{L} representa el operador Liouvilliano

$$\mathcal{L}\mu \equiv \{H, \mu\}.$$

Como dijimos en la sección 1.2, página 13, supondremos que los subsistemas estaban aislados y en equilibrio antes de que comenzara la interacción en $t = 0$, por lo que $\rho(z, 0) = \bar{\rho}(z, 0)$ (tomando $t = 0$ en la ecuación (1.2.11)) y $\delta\rho(z, 0) = 0$, luego

$$(1.4.7) \quad \delta\rho(z, t) = \int_0^t \exp\{\mathcal{Q}^\dagger \mathcal{L}(t-s)\} \mathcal{Q}^\dagger \mathcal{L} \bar{\rho}(z, s) ds.$$

Sustituyendo esta solución en la ecuación de evolución para $P(e, t)$ (1.4.2), y escribiendo $\bar{\rho}$ en función de P (1.2.8), obtenemos una ecuación integrodiferencial cerrada

exacta para la densidad de probabilidad macroscópica.

$$(1.4.8) \quad \frac{\partial P(e, t)}{\partial t} = - \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) P(e, t) + \int \{H, \delta\rho(z, t)\} \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dz,$$

con $\delta\rho(z, t)$ definida por la ecuación 1.4.7.

1.5. Equilibrio macroscópico

Ya hemos mencionado que $\Omega(e)$ es la distribución macroscópica (1.2.7) que corresponde a la distribución de equilibrio, $\rho^{eq.}(z)$. No es difícil demostrar que también es la solución estacionaria de la ecuación (1.4.8). En primer lugar,

$$\frac{\partial \Omega(e)}{\partial t} = 0,$$

por lo que el segundo miembro de la ecuación (1.4.8) debe anularse también. Utilizando (1.4.5) y (1.2.12), vemos que

$$- \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) \Omega(e) = \int \{H, \rho^{eq.}\} \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz = 0,$$

pues el corchete de Poisson se anula por (1.1.4). El segundo sumando se anula también, debido a la presencia de $\mathcal{L}\rho^{eq.} = \{H, \rho^{eq.}\}$ en el integrando.

Ahora bien, el hecho de que $\Omega(e)$ sea una solución estacionaria de la ecuación (1.4.8) no garantiza que sea un auténtico estado de equilibrio macroscópico, puesto que no sabemos si la distribución inicial se irá aproximando a la estacionaria al transcurrir el tiempo. Por ejemplo, si no hay interacción entre subsistemas, $H^{int.} = 0$, la distribución es en todo momento igual al producto de distribuciones microcanónicas (1.2.11) y no hay evolución macroscópica. En efecto, si suponemos que $H^{int.} = 0$, entonces

$$\mathcal{L}\bar{\rho} = \{H, \bar{\rho}\} = \rho^{eq.} \left\{ H^{int.}, \frac{P(H^1, \dots, H^n, t)}{\Omega(H^1, \dots, H^n)} \right\} = 0.$$

Pero tanto las funciones $v_j(e)$ (1.4.6) como $\delta\rho$ (1.4.7) incluyen en sus integrandos la función $\mathcal{L}\bar{\rho}$, por lo que la ecuación (1.4.8) se anula cuando no hay interacción,

$$H^{int.} = 0 \implies \frac{\partial P(e, t)}{\partial t} = 0.$$

Se sigue entonces que $H^{int.} \neq 0$ es una *condición necesaria* para que exista la tendencia hacia el equilibrio. Es fácil ver que no es condición suficiente mediante el ejemplo más sencillo posible: dos masas idénticas unidas con un muelle lineal, donde el hamiltoniano es

$$(1.5.1) \quad \begin{aligned} H(z) &= \sum_{j=1}^2 H^j(z) + H^{int.}(z) \\ &= \sum_{j=1}^2 \left(\frac{(p^j)^2}{2m} \right) + \frac{m\omega^2 (q^1 - q^2)^2}{2}. \end{aligned}$$

Las ecuaciones de Hamilton (1.1.2) son entonces

$$\begin{cases} \dot{q}^1 = \frac{p^1}{m}, & \dot{q}^2 = \frac{p^2}{m}, \\ \dot{p}^1 = m\omega^2 (q^1 - q^2), & \dot{p}^2 = -m\omega^2 (q^1 - q^2). \end{cases}$$

Es fácil comprobar que la solución del sistema es

$$\begin{cases} q^1 = A \sin\left(\frac{\omega}{\sqrt{2}}t + \delta\right), & q^2 = -A \sin\left(\frac{\omega}{\sqrt{2}}t + \delta\right), \\ p^1 = Am\omega \cos\left(\frac{\omega}{\sqrt{2}}t + \delta\right), & p^2 = -Am\omega \cos\left(\frac{\omega}{\sqrt{2}}t + \delta\right). \end{cases}$$

Todas las trayectorias del espacio de fases, $z(t)$, tienen periodo $T = \frac{2\pi}{\omega/\sqrt{2}}$. Sea cual sea la distribución de probabilidad inicial, podemos concluir que para la colectividad representada por el hamiltoniano (1.5.1), la distribución de probabilidad vuelve cíclicamente a su valor inicial

$$\rho(z(nT), nT) = \rho(z(0), 0), \quad n = 1, 2, 3, \dots$$

En otras palabras, si la distribución inicial $\rho(z, 0)$ es diferente de la microcanónica, entonces la colectividad no se acercará asintóticamente a la distribución de equilibrio al transcurrir el tiempo.

En realidad, es imposible que una distribución diferente de la de equilibrio y acotada en el espacio de fases tienda a ρ^{eq} , a consecuencia de la ecuación de Liouville (1.1.3). En efecto, si

$$\rho(z(0), 0) \neq \frac{1}{\int \delta(H(z) - E) dz}$$

para un punto tal que $H(z) = E$, entonces

$$\rho(z(t), t) \neq \frac{1}{\int \delta(H(z) - E) dz},$$

para cualquier $t \geq 0$. Ahora bien, puede ocurrir que la distribución ρ sea equivalente a ρ^{eq} cuando se lleva a cabo la integral sobre una región, es decir, puede que los valores esperados de las energías sean los mismos que los que predice (1.1.8),

$$\begin{aligned} \lim_{t \rightarrow \infty} \langle H^j \rangle &= \lim_{t \rightarrow \infty} \frac{\int H^j(z) \rho(z, t) dz}{\int \rho^{eq}(z) dz} \\ &= \frac{\int H^j(z) \rho^{eq}(z) dz}{\int \rho^{eq}(z) dz} \\ &= \frac{\int H^j(z) \delta(H(z) - E) dz}{\int \delta(H(z) - E) dz}. \end{aligned}$$

En este caso, se dice que el sistema se mezcla (la propiedad se conoce como *mixing*). Supongamos entonces que podemos escribir $\rho^{eq}(z)$ en lugar de $\rho(z, t)$ en las integrales de la ecuación (1.4.8) cuando $t \rightarrow \infty$. Entonces,

$$\lim_{t \rightarrow \infty} \delta\rho(z, t) = 0,$$

y los dos miembros de (1.4.8) se anulan. Utilizando (1.4.7), el límite anterior se convierte en

$$\lim_{t \rightarrow \infty} \int_0^t \exp\{Q^\dagger \mathcal{L}(t-s)\} Q^\dagger \mathcal{L} \bar{\rho}(z, s) ds = 0.$$

Está claro que el integrando se anula para $s \rightarrow t$, puesto que en este caso podemos cambiar $\bar{\rho}$ por ρ^{eq} , pero no podemos hacer lo mismo para $s \rightarrow 0$. Por lo tanto, el integrando debe tender a cero para aquellos valores de s para los que no se pueda intercambiar $\bar{\rho}(z, s)$ por $\rho^{eq}(z)$. Esto quiere decir que el sistema acabará “olvidando” su evolución pasada.

Si finalmente se alcanza el estado de equilibrio y podemos despreciar la energía de interacción, la distribución de probabilidad final es

$$\begin{aligned}\Omega(e) &= \int \rho^{eq.}(z) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz \\ &\approx \frac{\delta\left(\sum_{j=1}^n e_j - E\right)}{\int \delta(H(z) - E) dz} \left[\prod_{j=1}^n \Omega^j(e_j) \right] \\ &= \frac{\delta\left(\sum_{j=1}^n e_j - E\right)}{\int \delta(H(z) - E) dz} \exp\left\{ \frac{1}{k_B} \sum_{j=1}^n S^j(e_j) \right\},\end{aligned}$$

donde se han utilizado (1.2.7), (1.1.4), $H^{int.} \approx 0$ y las definiciones

$$\begin{aligned}\Omega^j(e_j) &= \int \delta(H^j(z) - e_j) dz^j, \\ S^j(e_j) &= k_B \ln(\Omega^j(e_j)).\end{aligned}$$

La función $\Omega(e)$ se anula para los estados incompatibles con la conservación de la energía. El valor más probable se puede determinar mediante el método de los multiplicadores de Lagrange, encontrando el máximo de

$$I(e, \lambda) = \frac{1}{\int \delta(H(z) - E) dz} \exp\left\{ \frac{1}{k_B} \sum_{j=1}^n S^j(e_j) \right\} + \lambda \left(\sum_{j=1}^n e_j - E \right).$$

La derivada de I respecto de λ produce la condición de conservación de la energía total,

$$\sum_{j=1}^n e_j = E,$$

y de las derivadas de I respecto de cada una de las variables e_j se obtienen expresiones de la forma

$$\frac{1}{\int \delta(H(z) - E) dz} \exp\left\{ \frac{1}{k_B} \sum_{j=1}^n S^j(e_j) \right\} \frac{1}{k_B} \frac{\partial S^j(e_j)}{\partial e_j} + \lambda = 0.$$

Despejando λ en la expresión anterior, y sustituyendo el resultado en la ecuación para $\frac{\partial I}{\partial e_k}$, obtenemos

$$\frac{1}{\int \delta(H(z) - E) dz} \exp\left\{ \frac{1}{k_B} \sum_{j=1}^n S^j(e_j) \right\} \frac{1}{k_B} \left(\frac{\partial S^j(e_j)}{\partial e_j} - \frac{\partial S^k(e_k)}{\partial e_k} \right) = 0,$$

que se satisface solamente para

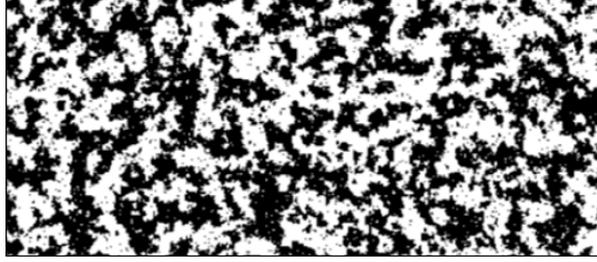
$$(1.5.2) \quad \frac{\partial S^j(e_j)}{\partial e_j} = \frac{\partial S^k(e_k)}{\partial e_k}.$$

Utilizando la definición

$$\frac{1}{T^j} = \frac{\partial S^j(e_j)}{\partial e_j},$$

vemos que (1.5.2) no es más que la igualdad de temperaturas entre los sistemas j -ésimo y k -ésimo, $T^j = T^k$. Aquí termina nuestra excursión desde la dinámica hamiltoniana microscópica hasta el equilibrio térmico.

La aproximación markoviana



En el capítulo anterior, las ecuaciones de la dinámica de varios subsistemas en interacción nos permitieron encontrar una expresión cerrada para la probabilidad $P(e, t)$ de que tuvieran sendas energías e_j en el tiempo t (1.4.8):

$$\frac{\partial P(e, t)}{\partial t} = - \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) P(e, t) + \int \{H, \delta\rho\} \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dz,$$

donde

$$v_j(e) = - \frac{1}{\Omega(e)} \int \rho^{eq.}(z) \{H^{int.}, H^j\} \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dz,$$

$$\delta\rho(z, t) = \int_0^t \exp\{\mathcal{Q}^\dagger \mathcal{L}(t-s)\} \mathcal{Q}^\dagger \mathcal{L} \left(\rho^{eq.}(z) \frac{P(H^1(z), \dots, H^n(z), s)}{\Omega(H^1(z), \dots, H^n(z))} \right) ds,$$

$$\Omega(e) = \int \rho^{eq.}(z) \left[\prod_{j=1}^n \delta(H^j(z) - e_j) \right] dz.$$

Vimos que $\Omega(e)$ era una solución estacionaria de (1.4.8) e indicamos en qué condiciones podía decirse que la distribución $P(e, t)$ se aproximaría a $\Omega(e)$ al transcurrir el tiempo. En los casos en los que puede despreciarse la energía de interacción frente a la energía interna de cada uno de los subsistemas, obtenemos el resultado familiar de la tendencia al equilibrio térmico, puesto que la distribución $\Omega(e)$ tiene un máximo para aquellos valores de las energías que igualan las temperaturas de los subsistemas.

Aunque estos resultados son agradables, todavía estamos muy lejos de nuestro objetivo declarado de reducir el tiempo de computación necesario para determinar la evolución macroscópica del intercambio energético entre sistemas. El problema con la ecuación (1.4.8) consiste en que es difícil tratarla analíticamente y una resolución numérica exigiría evaluar funciones intrincadas en todo el espacio de fases y a lo largo del intervalo $[0, t]$, ¡lo cual es más complicado que computar directamente la solución de la ecuación de Liouville (1.1.3)!

Nos vemos obligados, entonces, a utilizar aproximaciones, que serán más o menos acertadas dependiendo de la precisión buscada y del tiempo de computación que ahorren.

2.1. Aproximación mediante la distribución relevante

El paso más evidente parece ser la suposición de que $\delta\rho \approx 0$, es decir, que la distribución relevante es una buena aproximación de la distribución real, $\rho \approx \bar{\rho}$. Dado que $\bar{\rho}$ es función de las variables macroscópicas (1.2.8), esta aproximación implica que la evolución macroscópica del sistema es *markoviana*¹, es decir, que la derivada temporal de $P(e, t)$ en un instante arbitrario, t_a , depende sólo de $P(e, t_a)$. Nuestra ecuación cerrada (1.4.8) se reduce entonces a

$$(2.1.1) \quad \frac{\partial P(e, t)}{\partial t} = - \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) P(e, t).$$

Veamos un ejemplo sencillo. Utilizaremos como sistemas en interacción dos modelos de Ising unidimensionales con condiciones de contorno periódicas. Cada modelo es una cadena de espines que pueden tomar sólo uno de dos valores, correspondiendo a orientaciones opuestas, $s_i^j = \frac{1}{2}$ o $s_i^j = -\frac{1}{2}$. El hamiltoniano en este caso es

$$\begin{aligned} H &= H^1 + H^2 + H^{int}. \\ &= \left(-\frac{1}{2} \sum_{\langle i, k \rangle} s_i^1 s_k^1 + N_1 \right) + \left(-\frac{1}{2} \sum_{\langle i, k \rangle} s_i^2 s_k^2 + N_2 \right) + H^{int} = E, \end{aligned}$$

donde $\langle i, k \rangle$ indica que los nodos numerados por i y k son contiguos. Los sumandos N_1 y N_2 se introducen sólo para que el estado de mínima energía coincida con el valor cero. Desde el punto de vista energético, el nodo i -ésimo puede estar en uno de tres estados:

1. El estado de mínima energía, si $s_{i-1}^j = s_i^j = s_{i+1}^j$.
2. Un estado intermedio, si $s_{i-1}^j \neq s_{i+1}^j$.
3. Un estado de máxima energía, si $s_{i-1}^j = s_{i+1}^j \neq s_i^j$.

La interacción entre las cadenas viene determinada por el siguiente algoritmo sencillo. Pasado un tiempo característico Δt , se escoge un nodo i al azar entre los de ambas cadenas. Si el nodo está en el estado de mínima energía, se deja como está. Si está en el estado intermedio, entonces se voltea ese espín. Por último, si nos encontramos con el caso de máxima energía, también se voltea el espín, pero en este caso la conservación de la energía obliga a voltear un espín que se encuentre en el estado de mínima energía, y éste también se escoge al azar.

Como estamos tratando con dos sistemas, sólo necesitamos una variable macroscópica de energía, e , que corresponderá a la energía del primer sistema. La energía de la segunda cadena se puede obtener restando e de la energía total E . Dada la naturaleza discreta de esta descripción, deberíamos convertir (2.1.1) en la siguiente ecuación en diferencias,

$$(2.1.2) \quad \frac{P(e, t + \Delta t) - P(e, t)}{\Delta t} = \frac{v(e + \Delta e) P(e + \Delta e, t) - v(e) P(e, t)}{\Delta e}.$$

¹La imagen que encabeza este capítulo es una representación de un patrón emergente en los valores del momento magnético de un modelo de Ising bidimensional sencillo. El negro y blanco indican, respectivamente, momentos negativos y positivos. El modelo de Ising evoluciona según un algoritmo estocástico markoviano, es decir, sin memoria.

Desafortunadamente, el valor de $v(e + \Delta e) P(e + \Delta e, t)$ generalmente depende de si Δe se escoge positivo o negativo, y $v(e^+) \neq v(e^-)$. Si definimos las funciones

$$\begin{aligned} B^-(e) &= \frac{v(e^-)}{2\Delta e}, \\ B^+(e) &= \frac{v(e^+)}{2\Delta e}, \end{aligned}$$

podemos escribir la ecuación en diferencias de modo que quede reducida a (2.1.2) cuando $v(e + \Delta e) P(e + \Delta e, t)$ no dependa del signo de Δe , y $v(e^+) = v(e^-)$,

$$\begin{aligned} \frac{P(e, t + \Delta t) - P(e, t)}{\Delta t} &= B^-(e + |\Delta e|) P(e + |\Delta e|, t) \\ &\quad + B^+(e - |\Delta e|) P(e - |\Delta e|, t) \\ &\quad - (B^+(e) + B^-(e)) P(e, t). \end{aligned}$$

Durante el tiempo Δt , las cadenas pueden intercambiar (o no) dos unidades de energía. Escogiendo en consecuencia unidades tales que $\Delta t = 1$ y $\Delta e = 2$, y despejando $P(e, t + \Delta t)$, obtenemos la ecuación de evolución para $P(e, t)$,

$$(2.1.3) \quad \begin{aligned} P(e, t + 1) &= B^-(e + 2) P(e + 2, t) + B^+(e - 2) P(e - 2, t) \\ &\quad + (1 - (B^+(e) - B^-(e))) P(e, t). \end{aligned}$$

Escrita así, la ecuación permite descifrar el significado de las funciones B^- y B^+ : $B^-(e)$ no es más que la probabilidad de que la primera cadena ceda dos unidades de energía, mientras que $B^+(e)$ es la probabilidad de que las absorba. Como hemos supuesto que $\rho \approx \bar{\rho}$, podemos calcular directamente el valor de $B^-(e)$ suponiendo que todas las configuraciones con una energía dada tienen la misma probabilidad:

$$\begin{aligned} B^-(e) &= \frac{e}{N_1} \cdot \frac{(e-1)}{N_1-1} \cdot \frac{N_2 - (E - e + 1)}{N_1 - (e + 1) + N_2 - (E - e + 1)} \\ &= \frac{e(e-1)(N_2 + e - E - 1)}{N_1(N_1-1)(N_1 + N_2 - E - 2)}. \end{aligned}$$

La probabilidad $B^+(e)$ de que la cadena absorba dos unidades de energía es igual a la probabilidad de que las ceda la segunda cadena, por lo que

$$B^+(e) = \frac{(E - e)(E - e - 1)(N_1 - e - 1)}{N_1(N_1 - 1)(N_1 + N_2 - E - 2)}.$$

La figura 2.1.1 compara el comportamiento real de la probabilidad $P(e, t)$ calculado mediante simulaciones con la predicción de la ecuación (2.1.3). La aproximación funciona bastante bien en términos cualitativos, pero parece que sobreestima la velocidad a la que el sistema se aproxima al equilibrio.

¿De dónde viene la diferencia entre la predicción y la simulación? En el cálculo de los coeficientes B^- y B^+ , hicimos uso de la hipótesis de que las configuraciones microscópicas de la misma energía son equiprobables. En realidad, las configuraciones que permiten el intercambio de energía son algo menos probables. Para ver por qué, imaginemos un segmento de cadena con la siguiente configuración:

$$\dots, -\frac{1}{2}, -\frac{1}{2}, +\frac{1}{2}, -\frac{1}{2}, -\frac{1}{2}, \dots$$

La probabilidad de que se voltee el espín positivo es igual a la de que se voltee uno de los espines contiguos. Sin embargo, si ha ocurrido esto último, este segmento de cadena ya no puede ceder energía directamente. Además, la probabilidad de volver a la configuración con un único espín positivo es menor que la de que se voltee otro espín negativo contiguo. En otras palabras, existe una cierta tendencia a que se formen “parches” de espines orientados en la misma dirección, y esto hace que la

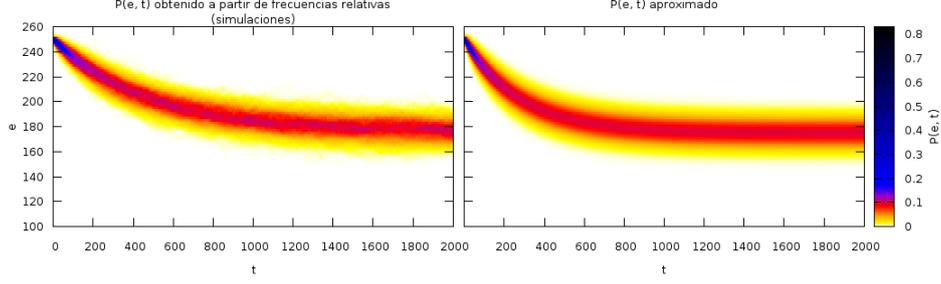


FIGURA 2.1.1. Comparación de las probabilidades $P(e, t)$ calculadas mediante simulaciones (*izquierda*) y la fórmula aproximada (2.1.3) (*derecha*). Se utilizaron dos modelos de Ising de $N_1 = N_2 = 500$ espines, con energías iniciales $e = 250$ y $E - e = 100$ unidades, respectivamente.

probabilidad de que se produzca un intercambio de energía sea menor que la que habíamos supuesto en el cálculo aproximado de $P(e, t)$.

2.2. Variables lentas

El ejemplo anterior ha demostrado la importancia de tener en cuenta la función $\delta\rho(z, t)$ y, como consecuencia, el segundo sumando en la ecuación (1.4.8). En algunos casos, este es el único sumando que importa ya que, como veremos más adelante, para una clase muy amplia de sistemas, $v_j(e) = 0$.

Debemos enfrentarnos, por tanto, a la integral de $\mathcal{L}\delta\rho = \{H, \delta\rho\}$ extendida a todos los microestados compatibles con las condiciones $H^j(e) = e_j$. Si la función $\delta\rho$ es tal que permite intercambiar el operador de Liouville con el signo integral, entonces

(2.2.1)

$$\int \{H, \delta\rho\} \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dz = \int_0^t \int \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] \mathcal{L} \exp \{ \mathcal{Q}^\dagger \mathcal{L}(t-s) \} \mathcal{Q}^\dagger \mathcal{L} \bar{\rho}(z, s) dz ds.$$

A continuación, utilizaremos la propiedad

$$\int f(z) dz = \int \int f(z) \left[\prod_{k=1}^n \delta(H^k(z) - e'_k) \right] dz de',$$

pero escribiremos el producto de funciones delta de manera más abreviada como

$$\Psi_e(z) = \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right].$$

Esto convierte la integral (2.2.1) en

$$\int \{H, \delta\rho\} \Psi_e(z) dz = \int_0^t \int \int \Psi_e(z) \mathcal{L} \exp \{ \mathcal{Q}^\dagger \mathcal{L}(t-s) \} \mathcal{Q}^\dagger \mathcal{L} \bar{\rho}(z, s) \Psi_{e'}(z) dz de' ds.$$

Si ahora utilizamos la definición de $\bar{\rho}$ (1.2.8), obtenemos una expresión más compacta,

$$\int \{H, \delta\rho\} \Psi_e(z) dz = \int_0^t \int \mathcal{D}(e, e', t-s) \frac{P(e', s)}{\Omega(e')} de' ds,$$

donde se ha introducido el operador \mathcal{D} , cuya definición es

$$(2.2.2) \quad \mathcal{D}(e, e', \tau) \equiv \int \Psi_e(z) \mathcal{L} \exp \{ \mathcal{Q}^\dagger \mathcal{L} \tau \} \mathcal{Q}^\dagger \mathcal{L} \rho^{eq.}(z) \Psi_{e'}(z) dz.$$

Este operador no depende de la distribución de probabilidad ρ , sino de la dinámica microscópica. Si pudiéramos llevar a cabo explícitamente la integral, tendríamos la siguiente ecuación cerrada exacta para la evolución de $P(e, t)$:

$$\frac{\partial P(e, t)}{\partial t} = - \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) P(e, t) + \int_0^t \int \mathcal{D}(e, e', t-s) \frac{P(e', s)}{\Omega(e')} de' ds.$$

Desgraciadamente, si nos vemos obligados a calcular numéricamente $\mathcal{D}(e, e', t)$, entonces no hemos avanzado demasiado.

Hacia el final del primer capítulo se mencionó que la tendencia al equilibrio térmico implica que

$$\lim_{t \rightarrow \infty} \delta \rho(z, t) = 0.$$

Ahora bien, si el proceso macroscópico parece markoviano, esto tiene que ser debido a que la función $\delta \rho$ tiende a anularse muy rápidamente, comparada con los tiempos característicos de variación de la probabilidad $P(e, t)$. Este hecho sugiere que introduzcamos la *hipótesis de variables lentas*: si la variación temporal de e es muy lenta comparada con el tiempo que tarda en anularse $\delta \rho$, entonces

$$\int_0^t \int \mathcal{D}(e, e', t-s) \frac{P(e', s)}{\Omega(e')} de' ds \approx \int \left(\int_0^t \mathcal{D}(e, e', t-s) ds \right) \frac{P(e', s)}{\Omega(e')} de'.$$

Pero como estamos suponiendo que $\mathcal{D}(e, e', t-s)$ es nula a partir de cierto valor (pequeño) de s , podemos extender indefinidamente la integral temporal

$$\int_0^t \mathcal{D}(e, e', t-s) ds = \int_0^\infty \mathcal{D}(e, e', t-s) ds.$$

Antes de hacer uso de esta hipótesis, encontraremos una expresión más conveniente para \mathcal{D} , utilizando los siguientes cuatro teoremas, cuyas demostraciones se han incluido en el apéndice A.

$$(2.2.3) \quad A(z) \mathcal{L}(\rho^{eq.}(z) B(z)) = \rho^{eq.}(z) A(z) \mathcal{L} B(z),$$

$$(2.2.4) \quad \int A(z) \mathcal{Q}^\dagger(\rho^{eq.}(z) B(z)) dz = \int \rho^{eq.}(z) A(z) \mathcal{Q} B(z) dz,$$

$$(2.2.5) \quad \int A(z) \exp \{ \mathcal{Q}^\dagger \mathcal{L} \tau \} (\rho^{eq.}(z) B(z)) dz = \int \rho^{eq.}(z) A(z) \exp \{ \mathcal{Q} \mathcal{L} \tau \} B(z) dz,$$

$$(2.2.6) \quad \int \rho^{eq.}(z) A(z) \mathcal{L} B(z) dz = - \int \rho^{eq.}(z) B(z) \mathcal{L} A(z) dz,$$

donde el proyector \mathcal{Q} se define como $\mathcal{Q} = I - \mathcal{P}$, y el operador \mathcal{P} se define como

$$(2.2.7) \quad \mathcal{P} A(z) = \int \frac{1}{\Omega(e)} \left(\int \rho^{eq.}(z') A(z') \Psi_e(z') dz' \right) \Psi_e(z) de.$$

\mathcal{P} y \mathcal{Q} son operadores adjuntos de \mathcal{P}^\dagger y \mathcal{Q}^\dagger , respectivamente, en el producto escalar

$$\langle A, B \rangle = \int A(z) B(z) dz.$$

Aplicando (2.2.3), (2.2.4), (2.2.5) y (2.2.6) a la definición de \mathcal{D} (2.2.2), obtenemos

$$(2.2.8) \quad \begin{aligned} \mathcal{D}(e, e', \tau) &= - \int \rho^{eq.}(z) (\exp\{\mathcal{QL}\tau\} \mathcal{QL}\Psi_{e'}(z)) (\mathcal{L}\Psi_e(z)) dz \\ &= - \int \rho^{eq.}(z) (\mathcal{QL}\Psi_{e'}(z)) \exp\{\mathcal{QL}\tau\} (\mathcal{L}\Psi_e(z)) dz. \end{aligned}$$

Las propiedades de los corchetes de Poisson nos permiten afirmar que

$$\mathcal{L}\Psi_e = \{H, \Psi_e\} = \sum_{j=1}^n \{H^{int.}, H^j\} \frac{\partial \Psi_e}{\partial H^j},$$

y, teniendo en cuenta que

$$\int A(z) \frac{\partial \Psi_e(z)}{\partial H^j} dz = \frac{\partial}{\partial e_j} \int A(z) \Psi_e(z) dz,$$

llegamos a la conclusión de que podemos escribir (2.2.8) como

$$\mathcal{D}(e, e', \tau) = - \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial e_j} \frac{\partial}{\partial e'_k} \Omega(e') D_{jk}(e, e', \tau),$$

siendo

$$(2.2.9) \quad \begin{aligned} D_{jk}(e, e', \tau) &= \frac{1}{\Omega(e')} \int \rho^{eq.}(z) (\mathcal{Q}\{H^{int.}, H^j\} \Psi_{e'}) \\ &\quad \times \exp\{\mathcal{QL}\tau\} (\mathcal{Q}\{H^{int.}, H^k\} \Psi_e) dz. \end{aligned}$$

Introduciendo $\mathcal{D}(e, e', \tau)$ en la ecuación de evolución para $P(e, t)$ e integrando por partes respecto de e'_k , obtenemos

$$\begin{aligned} \frac{\partial P(e, t)}{\partial t} &= - \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) P(e, t) \\ &\quad + \sum_{j=1}^n \sum_{k=1}^n \int_0^t \int \frac{\partial}{\partial e_j} \Omega(e') D_{jk}(e, e', \tau) \frac{\partial}{\partial e'_k} \frac{P(e', s)}{\Omega(e')} de' ds. \end{aligned}$$

Podemos escribir Ψ_e y $\Psi_{e'}$ a la izquierda del operador \mathcal{Q} , puesto que

$$\begin{aligned} \mathcal{Q}\Psi_e \mathcal{L}H^j &= (I - \mathcal{P}) \Psi_e \mathcal{L}H^j \\ &= \Psi_e \mathcal{L}H^j - \int \Psi_{e'} \Psi_e \mathcal{L}H^j dz \\ &= \Psi_e \mathcal{L}H^j - \Psi_e \int \Psi_{e'} \mathcal{L}H^j dz \\ &= \Psi_e (I - \mathcal{P}) \mathcal{L}H^j \\ &= \Psi_e \mathcal{Q}\mathcal{L}H^j. \end{aligned}$$

Si, además, utilizamos el hecho de que

$$\begin{aligned} \Psi_e \Psi_{e'} &= \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] \left[\prod_{k=1}^n \delta(H^k(z) - e'_k) \right] \\ &= \left[\prod_{k=1}^n \delta(e_k - e'_k) \right] \left[\prod_{k=1}^n \delta(H^k(z) - e'_k) \right], \end{aligned}$$

entonces podemos reescribir D_{jk} como

$$\begin{aligned} D_{jk}(e, e', \tau) &\approx \left[\prod_{k=1}^n \delta(e_k - e'_k) \right] \frac{1}{\Omega(e')} \int \rho^{eq.}(z) \Psi_{e'}(z) (\mathcal{Q}\{H^{int.}, H^j\}) \\ &\quad \times \exp\{\mathcal{QL}\tau\} (\mathcal{Q}\{H^{int.}, H^k\}) dz. \end{aligned}$$

Al llevar a cabo la integración temporal, usando la hipótesis de variables lentas, obtenemos una *ecuación de Fokker-Planck*,

$$(2.2.10) \quad \frac{\partial P(e, t)}{\partial t} = - \sum_{j=1}^n \frac{\partial}{\partial e_j} v_j(e) P(e, t) + \sum_{j=1}^n \sum_{k=1}^n \frac{\partial}{\partial e_j} \Omega(e) \mathcal{D}_{jk}(e) \frac{\partial}{\partial e_k} \frac{P(e, s)}{\Omega(e)},$$

donde el llamado *tensor de difusión* \mathcal{D}_{jk} es

$$(2.2.11) \quad \mathcal{D}_{jk}(e) = \int_0^\infty \frac{1}{\Omega(e')} \int \rho^{eq.}(z) \Psi_{e'}(z) (\mathcal{Q}\{H^{int.}, H^j\}) \times \exp\{\mathcal{Q}\mathcal{L}\tau\} (\mathcal{Q}\{H^{int.}, H^k\}) dz d\tau.$$

Desde el punto de vista macroscópico, la ecuación de Fokker-Planck (2.2.10) da lugar a una evolución markoviana de la probabilidad $P(e, t)$. Los coeficientes $v_j(e)$ y $\mathcal{D}_{jk}(e)$ dependen sólo de la dinámica microscópica, y no de la distribución de probabilidad. Para que la hipótesis de variables lentas sea efectiva, debe haber una separación clara de escalas temporales. En otras palabras, los tiempos característicos de variación de la variable z deben ser mucho menores que los de e .

2.3. Energías potenciales independientes de los momentos conjugados

Muchos sistemas clásicos poseen un hamiltoniano de la forma

$$H = \sum_i \frac{(p_i)^2}{2m_i} + V(q_1, \dots, q_N),$$

en la que la energía potencial V es independiente de los valores de los momentos conjugados p_i . Si nuestro hamiltoniano (1.1.1) tuviera esta propiedad, podríamos escribirlo en la forma

$$(2.3.1) \quad H = \sum_{j=1}^n \sum_{i=1}^{N_j} \left[\frac{(p_i^j)^2}{2m_i^j} + V^j(q_1^j, \dots, q_{N_j}^j) \right] + H^{int.}(q_1^1, \dots, q_i^j, \dots, q_{N_n}^n).$$

Como consecuencia,

$$\frac{\partial H^{int.}}{\partial p_i^j} = 0,$$

y el corchete de Poisson $\mathcal{L}H^j$ se convertiría en

$$(2.3.2) \quad \mathcal{L}H^j = \{H^{int.}, H^j\} = \sum_{i=1}^{N_j} \frac{\partial H^{int.}}{\partial q_i^j} \cdot \frac{p_i^j}{m_i^j}.$$

De ahora en adelante, utilizaremos la notación de los *promedios restringidos*. El promedio de $A(z)$ restringido al macroestado e se define de la forma siguiente:

$$\langle A(z) \rangle^e = \frac{1}{\Omega(e)} \int \rho^{eq.}(z) A(z) \Psi_e(z) dz.$$

Si introducimos el resultado (2.3.2) en la ecuación para $v_j(e)$ (1.4.6), comprobamos que

$$v_j(e) = \langle \mathcal{L}H^j \rangle^e = 0,$$

debido a que la integral es antisimétrica en p_i^j . Esto nos deja sólo con el segundo sumando de la ecuación de Fokker-Planck (2.2.10). Utilizando de nuevo la notación

en términos de promedios restringidos, el tensor de difusión es

$$\mathcal{D}_{jk}(e) = \int_0^\infty \left\langle \left(\mathcal{L}H^j - \langle \mathcal{L}H^j \rangle^e \right) \exp\{\mathcal{Q}\mathcal{L}t'\} \left(\mathcal{L}H^k - \langle \mathcal{L}H^k \rangle^e \right) \right\rangle^e dt'.$$

Utilizaremos la aproximación

$$\mathcal{D}_{jk}(e) \approx \int_0^\infty e^{-t'/\tau} \left\langle \left(\mathcal{L}H^j - \langle \mathcal{L}H^j \rangle^e \right) \left(\mathcal{L}H^k - \langle \mathcal{L}H^k \rangle^e \right) \right\rangle^e dt',$$

para modelar el rápido decaimiento (exponencial) del integrando. Ambas expresiones del integrando coinciden para $t' = 0$, pero serán los experimentos o las simulaciones los que determinen si se parecen al transcurrir el tiempo.

Anteriormente, vimos que $v_j(e) = \langle \mathcal{L}H^j \rangle^e = 0$ para los hamiltonianos de la forma (2.3.1), luego

$$\mathcal{D}_{jk} \approx \int_0^\infty e^{-t'/\tau} \langle (\mathcal{L}H^j) (\mathcal{L}H^k) \rangle^e dt' = \tau \langle (\mathcal{L}H^j) (\mathcal{L}H^k) \rangle^e.$$

El promedio restringido del integrando se anula para $j \neq k$, debido a que las integrales son antisimétricas en p_i^j . Entonces, sólo hay que calcular los términos

$$(2.3.3) \quad \mathcal{D}_{jj} \approx \tau \langle (\mathcal{L}H^j)^2 \rangle^e.$$

El factor τ viene de realizar la integral sobre t' .

La expresión contenida en el promedio restringido es

$$(\mathcal{L}H^j)^2 = \sum_{i=1}^{N_j} \sum_{l=1}^{N_j} \frac{p_i^j}{m_i^j} \frac{p_l^j}{m_l^j} \frac{\partial H^{int.}}{\partial q_i^j} \frac{\partial H^{int.}}{\partial q_l^j}.$$

La mayoría de los términos en la suma se anulan al realizar la integral, debido a la asimetría en los términos p_i^j , lo cual nos deja sólo los términos en los que $i = l$,

$$\langle (\mathcal{L}H^j)^2 \rangle^e = \sum_{i=1}^{N_j} \left\langle \left(\frac{p_i^j}{m_i^j} \frac{\partial H^{int.}}{\partial q_i^j} \right)^2 \right\rangle^e.$$

Explícitamente,

$$(2.3.4) \quad \langle (\mathcal{L}H^j)^2 \rangle^e = \sum_{i=1}^{N_j} \frac{1}{\Omega(e)} \int \rho^{eq.}(z) \left(\frac{p_i^j}{m_i^j} \frac{\partial H^{int.}}{\partial q_i^j} \right)^2 \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dp dq.$$

Si despreciamos la energía de interacción frente a la de los subsistemas, $H^{int} \ll H^j$, entonces podemos escribir la distribución de equilibrio en la forma

$$\rho^{eq.}(z) = \frac{\delta(\sum_{k=1}^n H^k(z) - E)}{\Omega(H^1(z), \dots, H^n(z))}.$$

Esto nos permitiría integrar la distribución de equilibrio en (2.3.4),

$$\langle (\mathcal{L}H^j)^2 \rangle^e = \sum_{i=1}^{N_j} \frac{\delta(\sum_{k=1}^n e_k - E)}{(\Omega(e))^2} \int \left(\frac{p_i^j}{m_i^j} \frac{\partial H^{int.}}{\partial q_i^j} \right)^2 \left[\prod_{k=1}^n \delta(H^k(z) - e_k) \right] dp dq.$$

Si todas las partículas del subsistema j tienen la misma masa, escribimos $m_i^j = m^j$ y podemos hacer uso de la integral

$$\begin{aligned} \int \frac{(p_i^j)^2}{2m_i^j} \delta(H^j(z) - e_k) dp^j &= \frac{1}{N_j} \int \left(\sum_{i=1}^{N_j} \frac{(p_i^j)^2}{2m^j} \right) \delta \left(\sum_{i=1}^{N_j} \frac{(p_i^j)^2}{2m^j} + V^j(q^j) - e_j \right) dp^j \\ &= \frac{1}{N_j} (e_j - V^j(q^j)) \int \delta \left(\sum_{i=1}^{N_j} \frac{(p_i^j)^2}{2m^j} + V^j(q^j) - e_j \right) dp^j \\ &= \frac{\pi^{N_j/2} \sqrt{2m^j}}{N_j \Gamma(N_j/2)} (e_j - V^j(q^j))^{N_j/2}. \end{aligned}$$

Aplicando estas dos expresiones en la del tensor de difusión (2.3.3), obtenemos

$$\begin{aligned} \mathcal{D}_{jj}(e) = \tau \langle (\mathcal{L}H^j)^2 \rangle^e &= \tau \sum_{i=1}^{N_j} \frac{\delta(\sum_{k=1}^n e_k - E)}{(\Omega(e))^2} \frac{\pi^{N_j/2} \sqrt{2m^j}}{N_j \Gamma(N_j/2)} \\ &\quad \times \int \left(\frac{\partial H^{int.}}{\partial q_i^j} \right)^2 (e_j - V^j(q^j))^{N_j/2} dq^j \\ &\quad \times \left[\prod_{k \neq j} \Omega^k(e_k) \right], \end{aligned}$$

donde se han introducido las funciones $\Omega^k(e_k)$,

$$\Omega^k(e_k) = \int \delta(H^k(z) - e_k) dz^k.$$

Este es un buen momento para detenernos y recapitular. Hemos visto que, si el hamiltoniano de nuestro sistema tiene la forma (2.3.1) y se cumple la hipótesis de variables lentas, entonces se anulan las funciones $v_j(e)$ en la ecuación de Fokker-Planck (2.2.10), y el tensor de difusión \mathcal{D}_{jk} queda reducido a forma diagonal. Además, hemos encontrado una expresión analítica para $\mathcal{D}_{jj}(e)$ que, al menos en teoría, es integrable para ciertos potenciales sencillos. Por ejemplo, si las partículas interactúan con potenciales armónicos, la anterior integral respecto de las variables q^j resulta ser un polinomio. Desgraciadamente, en los casos que verdaderamente nos interesan, con números tan elevados de grados de libertad, un polinomio así es inmanejable en la práctica.

2.4. Intercambio de energías entre gases ideales

No hay por qué sentirse descorazonado. Al fin y al cabo, hemos conseguido expresar $\mathcal{D}_{jj}(e)$ en términos de una autocorrelación de la variación de la energía del sistema j -ésimo para un valor fijo de e (2.3.3). Estos valores se pueden calcular con relativa facilidad a partir de una simulación numérica de dinámica molecular.

Podría parecer que no hemos ganado nada si ahora tenemos que recurrir a la dinámica molecular para calcular el tensor de difusión. La clave está en darse cuenta de que el promedio que aparece en la expresión (2.3.3) es un promedio *restringido*, es decir, que depende solo de las propiedades de equilibrio de los sistemas por separado. Basta calcular *una vez* los valores de $\langle (\mathcal{L}H^j)^2 \rangle^e$ para un rango de energías mediante dinámica molecular, y después podremos usar la ecuación de Fokker-Planck (2.2.10) para simular *cualquier* proceso de intercambio de energía en dicho rango.

Veamos lo que ocurre en el caso de una mezcla homogénea de dos gases ideales a temperaturas iniciales diferentes. Si está en equilibrio, un gas ideal está formado

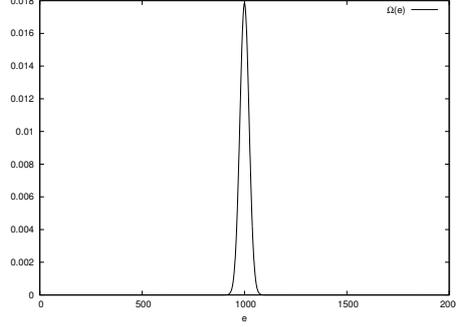


FIGURA 2.4.1. Distribución de probabilidad en el equilibrio de los valores de la energía e de un gas ideal en una mezcla homogénea de dos gases idénticos de 1000 partículas cada uno y con energía total $E = 2000$.

por partículas con energías cinéticas que siguen una distribución exponencial. Como conocemos la probabilidad de encontrar una partícula de energía ε ,

$$(2.4.1) \quad P(\varepsilon) = \frac{1}{k_B T} \exp \left\{ -\frac{\varepsilon}{k_B T} \right\},$$

de acuerdo con la distribución de Maxwell-Boltzmann, podemos crear artificialmente un microestado de equilibrio, escogiendo aleatoriamente la posición y la dirección de las partículas (según una distribución uniforme) y asignando una velocidad de módulo

$$v = \sqrt{\frac{2\varepsilon}{m}},$$

donde ε se escoge de acuerdo con (2.4.1).

Ciertamente, si los gases fueran ideales *de verdad*, en el sentido de que sus partículas no interaccionaran en absoluto, nunca podrían llegar a equilibrarse. Supondremos, por tanto, que existe una pequeña energía de interacción. Para la simulación se utilizará un gas de esferas duras (ver apéndice B, p. 47). Aunque el potencial en este caso no es estrictamente independiente de los momentos, imaginaremos que en realidad es una aproximación de un potencial de tipo Lennard-Jones, que sí lo es.

Como en el ejemplo de las cadenas de Ising, nuestra ecuación de evolución de P tendrá sólo una variable relevante, correspondiente a la energía e de uno de los gases,

$$\frac{\partial P(e, t)}{\partial t} = \frac{\partial}{\partial e} \Omega(e) \mathcal{D}(e) \frac{\partial P(e, t)}{\partial e} \frac{1}{\Omega(e)}.$$

La función $\Omega(e)$ representa la distribución de equilibrio de los valores de e , y se puede hallar llevando a cabo la integral (1.2.7) (ver figura 2.4.1). Sin embargo, es más sencillo utilizar la definición

$$\frac{1}{T^j} = \frac{\partial}{\partial e_j} \ln(\Omega_j(e_j))$$

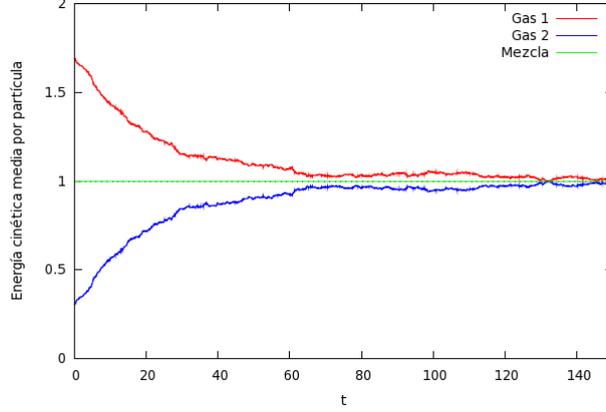


FIGURA 2.4.2. Evolución temporal de la energía media por partícula de una mezcla homogénea de dos gases ideales idénticos ($N_1 = N_2 = 1000$). En rojo, se representa la energía correspondiente al gas que inicialmente tenía una temperatura mayor (energía inicial $e_1 = 1700$). Las líneas azul y verde son, respectivamente, para el gas que empezó más frío ($e_2 = 300$) y la energía media por partícula del sistema total ($E = \frac{1}{2}(e_1 + e_2)$). La línea de puntos indica la energía asociada con la temperatura de equilibrio.

para reescribir la ecuación de Fokker-Planck, llevando a cabo la derivada de $\frac{P(e, t)}{\Omega(e)}$,

(2.4.2)

$$\begin{aligned} \frac{\partial P(e, t)}{\partial t} &= \frac{\partial}{\partial e} \left[\mathcal{D}(e) \left(\frac{\partial P(e, t)}{\partial e} - P(e, t) \frac{\partial}{\partial e} \ln(\Omega(e)) \right) \right] \\ &= \frac{\partial}{\partial e} \left[\mathcal{D}(e) \left(\frac{\partial P(e, t)}{\partial e} - \left(\frac{1}{T^1(e)} - \frac{1}{T^2(E-e)} \right) P(e, t) \right) \right]. \end{aligned}$$

Las funciones $T^j(e)$ deben interpretarse como la temperatura que tendría el gas j , aislado y en equilibrio, si su energía interna fuera e . De acuerdo con el teorema de equipartición,

$$T^1(e) = \frac{2}{3} \frac{e}{N_1 k_B}, \quad T^2(E-e) = \frac{2}{3} \frac{(E-e)}{N_2 k_B}.$$

La ecuación (2.4.2) tiene dos rasgos muy satisfactorios. Primero, tiene incorporado un arrastre que tiende a igualar las temperaturas T^1 y T^2 . Segundo, cuando las temperaturas son iguales, la ecuación corresponde simplemente a un fenómeno de difusión.

Las simulaciones de dinámica molecular confirman el resultado esperado: al poner en contacto los gases, el más caliente cede energía al más frío hasta que se alcanza la temperatura de equilibrio (ver figura 2.4.2).

Veamos ahora qué predice la ecuación de Fokker-Planck. Para poder llevar a cabo la integración numérica de (2.4.2), primero hay que determinar la función $\mathcal{D}(e)$ a partir de simulaciones de dinámica molecular. En el apartado 2.3, llegamos a la conclusión de que podía escribirse como una autocorrelación multiplicada por un tiempo característico. Para nuestra mezcla de gases ideales,

$$\mathcal{D}(e) \approx \tau \left\langle (\mathcal{L}H^1)^2 \right\rangle^e,$$

siendo τ un tiempo que, en principio, podría depender de la energía. Debemos calcular, entonces, el promedio de la función $(\mathcal{L}H^1)^2$ sobre todos los puntos del

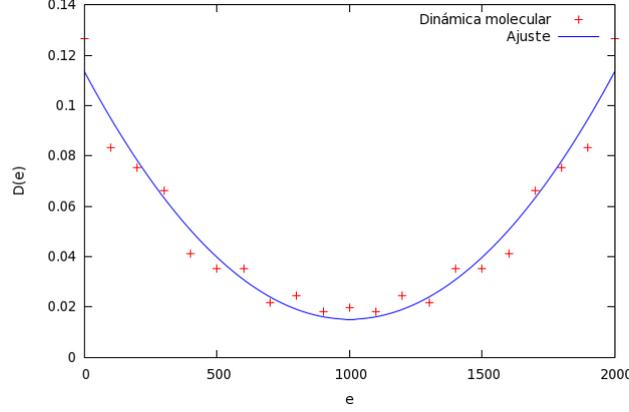


FIGURA 2.4.3. Tensor de difusión $\mathcal{D}(e)$ frente a la energía de un gas en una mezcla homogénea de dos gases ideales. Los resultados de las simulaciones de dinámica molecular se indican con cruces rojas. En azul, se ha representado una curva de ajuste, utilizada posteriormente en la integración numérica de la ecuación de Fokker-Planck (2.4.2).

espacio de fases tales que $H^1(z) = e$. Pero esto podemos hacerlo suponiendo simplemente que no hay interacción entre los gases 1 y 2, y hallando el promedio temporal $\langle (\mathcal{L}H^1)^2 \rangle_t$, de acuerdo con la hipótesis de que la dinámica de equilibrio será ergódica. Para determinar τ , buscaremos el valor a partir del cual se anula la autocorrelación $\langle (\mathcal{L}H^1(t)) (\mathcal{L}H^1(t-\tau)) \rangle$. Los resultados de estos cálculos se presentan en la figura 2.4.3. Los valores obtenidos mediante las simulaciones se ajustaron a una curva suave que después se utilizó para integrar la ecuación (2.4.2).

La integración numérica de una ecuación diferencial en derivadas parciales como (2.4.2) presenta muchos problemas técnicos. En general, en estos problemas se suele utilizar la ecuación diferencial estocástica asociada para simular un gran número de procesos estocásticos y construir a partir de ellos un histograma que sirva de aproximación a la distribución P . En este caso, sin embargo, se integró directamente la ecuación de Fokker-Planck utilizando el método de Crank-Nicolson (ver apéndice B, p. 57), que eliminaba gran parte de las inestabilidades numéricas (aunque no todas). La evolución temporal resultante de la distribución de probabilidad para e se presenta en la figura 2.4.4, comparada con una realización de $e(t)$ calculada a partir de las simulaciones de dinámica molecular.

Los sistemas macroscópicos suelen tener distribuciones de probabilidad muy concentradas alrededor del valor esperado de las variables relevantes. En estos casos, se pueden despreciar las fluctuaciones. Si aproximamos la función \mathcal{D} tratándola como una constante, podemos hallar fácilmente una ecuación para el valor esperado de la energía e . Multiplicando la ecuación de Fokker-Planck (2.4.2) por e , e integrando por partes respecto de e , obtenemos

$$\frac{\partial \langle e \rangle}{\partial t} = \mathcal{D} \int_0^\infty \left(\frac{1}{T^1(e)} - \frac{1}{T^2(E-e)} \right) P(e, t) de.$$

Si P tiene un único pico muy pronunciado y centrado en $\langle e \rangle$, entonces podemos utilizar la aproximación $P(e, t) \approx \delta(e - \langle e \rangle)$, que nos deja con la ecuación

$$(2.4.3) \quad \frac{\partial \langle e \rangle}{\partial t} = \mathcal{D} \left(\frac{1}{T^1(\langle e \rangle)} - \frac{1}{T^2(E - \langle e \rangle)} \right).$$

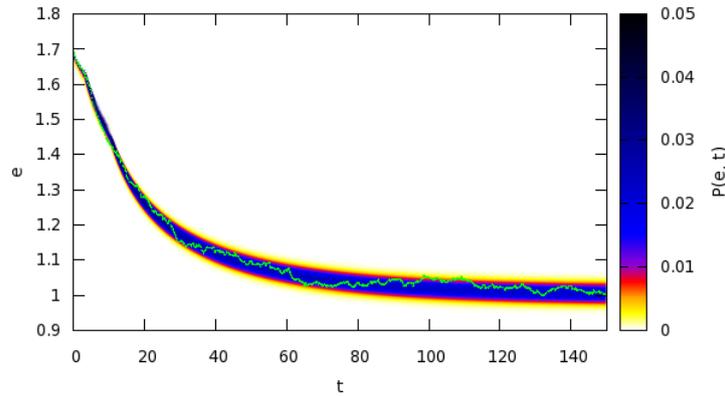


FIGURA 2.4.4. Evolución temporal de la distribución de probabilidad $P(e, t)$ para la energía media por partícula de un gas ideal en una mezcla homogénea de dos gases idénticos. El valor inicial de la energía media por partícula es $e = 1,7$. En el equilibrio, la distribución queda centrada en la energía correspondiente a la temperatura de equilibrio. Se ha representado en verde una realización de $e(t)$ calculada a partir de una simulación de dinámica molecular.

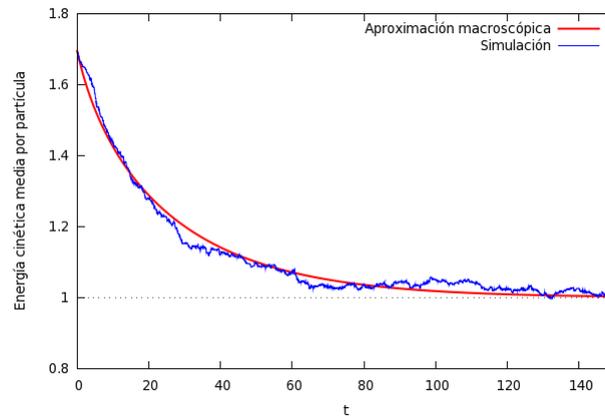


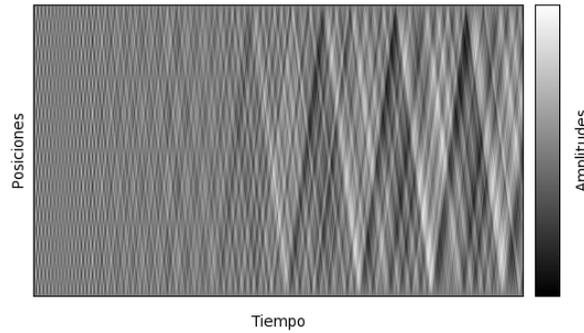
FIGURA 2.4.5. Evolución del valor esperado de la energía media por partícula (*en rojo*) según la aproximación (2.4.3) comparada con una realización de $e(t)$ calculada mediante dinámica molecular (*en azul*).

La integración numérica de la ecuación anterior para nuestro sistema de gases produce la curva que se muestra en la figura 2.4.5, que resulta ser una aproximación razonable de la dinámica del intercambio de energías.

La técnica de los operadores de proyección ha tenido bastante éxito en el tratamiento de la dinámica de la mezcla de gases ideales, pero no hay duda de que este es un problema académico de aplicación relativamente limitada. Para conseguir una mezcla homogénea de gases ideales a temperaturas diferentes, necesitamos gases lo suficientemente diluidos como para que el recorrido libre medio de las partículas sea comparable con la longitud característica del volumen que las contiene. En cualquier otro caso, habrá que tener en cuenta la energía que se transfiere entre ellos mientras se mezclan. Sin embargo, el método ilustrado en este apartado

puede aplicarse en principio a otros sistemas más complejos, suponiendo que uno tenga la suerte de encontrar un sistema que verdaderamente tenga una dinámica markoviana.

Procesos no markovianos



La dinámica del flujo de calor entre dos sistemas gaseosos que cerraba el capítulo anterior no nos ha dejado ningún resultado sorprendente. Podríamos haber anticipado las características generales del proceso sin habernos embarcado en la travesía que llevó a la deducción de la ecuación de Fokker-Planck (2.4.2). El gas más caliente cede energía al frío, y tiene lugar una aproximación monótona a la temperatura de equilibrio. No hay duda de que es valiosa la capacidad que poseemos ahora para predecir el tiempo que tardará el sistema en equilibrarse y la amplitud de las fluctuaciones, pero la tendencia general del proceso no nos ha dejado ningún rasgo intrigante.

La técnica esbozada en el capítulo anterior puede aplicarse en principio a un rango muy amplio de sistemas, aunque es posible que haya que refinar alguna de las aproximaciones (como en el cálculo del tensor de difusión, por ejemplo). Sin embargo, para poder utilizar la ecuación de Fokker-Planck (2.2.10), es imprescindible que se cumpla la hipótesis de variables lentas. Conviene, por tanto, reflexionar sobre las limitaciones de esta hipótesis.

La aproximación markoviana consistía en reconocer que $\delta\rho$, la parte no relevante de la distribución microscópica, tiende a anularse en tiempos muy cortos comparados con los tiempos característicos de evolución de las energías macroscópicas. Recordemos que $\delta\rho$ representa en cuánto se aparta la distribución real ρ de la distribución relevante $\bar{\rho}$,

$$\delta\rho = \rho - \bar{\rho},$$

lo que significa que la hipótesis de variables lentas dejará de ser válida cuando ρ se aleje apreciablemente de $\bar{\rho}$ durante tiempos comparables a los tiempos de evolución de las energías macroscópicas.

Por ejemplo, si tenemos dos esferas metálicas aisladas a diferentes temperaturas y las ponemos en contacto por medio de un hilo metálico conductor de masa despreciable, se desencadenará un flujo de calor entre ellas, pero las esferas no tendrán una temperatura uniforme durante el proceso, sino que aparecerá un gradiente térmico entre los puntos más cercanos al hilo y los más alejados. Si las variables

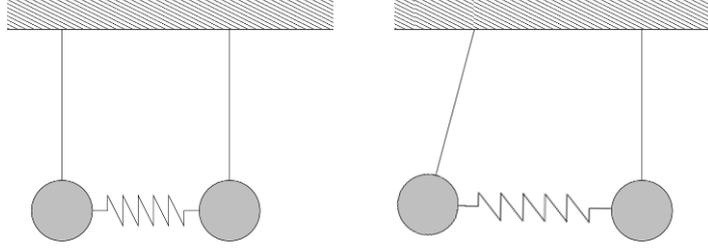


FIGURA 3.0.1. Sistema aislado de dos esferas metálicas unidas mediante un muelle lineal débil metálico, en la posición de equilibrio mecánico (izquierda) y con una masa desplazada respecto de su posición de equilibrio (derecha).

macroscópicas fueran las energías almacenadas en las esferas, la distribución relevante $\bar{\rho}$ correspondería a un reparto uniforme de cada energía en todo el volumen de la esfera correspondiente, en el sentido de que es muy poco probable que el sistema se encuentre en un microestado que se aleje apreciablemente de este reparto uniforme. Por lo tanto, el proceso de equilibrado real pasará la mayor parte del tiempo lejos de la distribución relevante.

Pero supongamos que la transferencia de energía de una esfera a otra es lenta comparada con el tiempo que tarda en equilibrarse la temperatura en el seno de cada esfera. Todavía hay un efecto que puede evitar que la evolución sea markoviana: el trabajo macroscópico. Supongamos que el contacto metálico es un muelle lineal débil (ver figura 3.0.1). Si apartamos una de las esferas de su posición de equilibrio, observaremos que la energía mecánica se traslada de una esfera a otra, y de vuelta a la primera, sin que importe si el flujo está dirigido hacia el cuerpo más caliente o el más frío. Evidentemente, el estado final será también de equilibrio mecánico y térmico, pero la evolución temporal de las energías será muy diferente de la aproximación monótona al equilibrio que vimos en el capítulo anterior.

En ocasiones, el trabajo macroscópico no es tan conspicuo como en el ejemplo anterior. Este capítulo trata de una de esas ocasiones. Narra la historia de cómo fracasó con estrépito nuestro intento de aplicar la aproximación markoviana al equilibrado entre cadenas de Fermi-Pasta-Ulam¹.

3.1. El sistema Fermi-Pasta-Ulam

En el ejemplo de la mezcla de gases, se mencionó que una ausencia completa de interacción entre las partículas impediría que se alcanzase el equilibrio. Cuando se utiliza una red de osciladores lineales para representar un sólido cristalino ocurre algo parecido. Los modos normales de vibración están completamente desacoplados y el sistema se comporta como un “gas ideal de modos”.

A mediados de los años cincuenta, Fermi, Pasta y Ulam se propusieron simular el equilibrado de una cadena de osciladores unidimensional con extremos fijos. Para ello, introdujeron una pequeña corrección anarmónica en las energías potenciales para cada masa,

$$V_i(q) = \frac{1}{2}\kappa(2q_i - q_{i-1} - q_{i+1})^2 + \frac{1}{4}\lambda \left[(q_i - q_{i+1})^4 - (q_{i-1} - q_i)^4 \right],$$

con $x_0 = x_{N+1} = 0$ (el parámetro λ regula la intensidad de la contribución anarmónica). Esperaban que, después de haber excitado el modo fundamental, la

¹La imagen al comienzo del capítulo representa la evolución temporal de una cadena de Fermi-Pasta-Ulam de 256 masas. Puede observarse claramente el nacimiento de un solitón que recorre la cadena de lado a lado, reflejándose en los extremos.

energía migrara gradualmente a otros modos hasta distribuirse equitativamente. Al principio las simulaciones arrojaban los datos esperados pero, para sorpresa de los investigadores, transcurrido un tiempo muy largo (de más de 1000 periodos de oscilación) casi toda la energía regresaba al modo fundamental. Este comportamiento inesperado, conocido como la paradoja de Fermi-Pasta-Ulam, inspiró toda una serie de estudios en campos que más tarde se denominarían caos dinámico, física computacional y sistemas complejos, revelando fenómenos muy interesantes. Entre ellos, se observó la formación de solitones en el límite continuo.

Los estudios subsiguientes demostraron que existe un umbral a partir del cual la perturbación anarmónica era suficientemente intensa como para que se instalara un régimen ergódico, que repartía la energía entre los diferentes modos normales (definidos en ausencia de perturbación).

3.2. Simular el equilibrio térmico

Para el diseño de nuestros experimentos numéricos debemos recordar la siguiente lección: es necesario encontrar rangos de valores de los parámetros donde la dinámica sea ergódica, puesto que queremos sustituir promedios sobre el espacio de fases por promedios temporales. Debemos asegurar entonces que la energía se ha repartido de manera más o menos equitativa entre los diferentes modos normales antes de poner dos cadenas en contacto entre sí.

La primera dificultad consiste en encontrar la amplitud que debemos asignar a un modo de vibración para que la energía total almacenada en la cadena sea igual a un valor previamente escogido e . El hamiltoniano de una cadena unidimensional de osciladores anarmónicos de Fermi-Pasta-Ulam es

$$H(q, p) = \sum_{i=1}^N \left(\frac{p_i^2}{2m} + \frac{1}{2}\kappa (q_i - q_{i-1})^2 + \frac{1}{4}\lambda (q_i - q_{i-1})^4 \right).$$

Cuando las masas se encuentran en el estado de máxima elongación, se anula el término de energía cinética y tenemos

$$\begin{aligned} e &= \sum_{i=1}^N \left(\frac{1}{2}\kappa (q_i - q_{i-1})^2 + \frac{1}{4}\lambda (q_i - q_{i-1})^4 \right) \\ &= \frac{\kappa}{2} \sum_{i=1}^N (q_i - q_{i-1})^2 + \frac{\lambda}{4} \sum_{i=1}^N (q_i - q_{i-1})^4. \end{aligned}$$

El vector de posiciones del modo normal n -ésimo se define como

$$q_i = A \sin \left(2\pi n \frac{i}{N} \right),$$

así que

$$\begin{aligned} e &= \frac{\kappa}{2} A^2 \sum_{i=1}^N \left(\sin \left(2\pi n \frac{i}{N} \right) - \sin \left(2\pi n \frac{(i-1)}{N} \right) \right)^2 \\ &\quad + \frac{\lambda}{4} A^4 \sum_{i=1}^N \left(\sin \left(2\pi n \frac{i}{N} \right) - \sin \left(2\pi n \frac{(i-1)}{N} \right) \right)^4. \end{aligned}$$

Para valores suficientemente grandes de N , se puede aproximar la diferencia entre senos por medio de la expresión

$$\frac{\sin \left(2\pi n \frac{i}{N} \right) - \sin \left(2\pi n \frac{(i-1)}{N} \right)}{\frac{1}{N}} \approx 2\pi n \cos \left(2\pi n \frac{i}{N} \right).$$

Esto nos deja con la siguiente suma para la energía e :

$$\begin{aligned} e &= \frac{k}{2} A^2 \sum_{i=1}^N \left(\frac{2\pi n}{N} \cos \left(2\pi n \frac{i}{N} \right) \right)^2 + \frac{\lambda}{4} A^4 \sum_{i=1}^N \left(\frac{2\pi n}{N} \cos \left(2\pi n \frac{i}{N} \right) \right)^4 \\ &= \frac{k}{2} A^2 \frac{(2\pi n)^2}{N} \sum_{i=1}^N \cos^2 \left(2\pi n \frac{i}{N} \right) \frac{1}{N} + \frac{\lambda}{4} A^4 \frac{(2\pi n)^4}{N^3} \sum_{i=1}^N \cos^4 \left(2\pi n \frac{i}{N} \right) \frac{1}{N}. \end{aligned}$$

Ahora podemos aproximar las sumas usando la idea

$$\sum_{i=1}^N f \left(\frac{i}{N} \right) \frac{1}{N} \approx \int_0^1 f(x) dx.$$

Si calculamos las integrales para valores enteros de n ,

$$\int_0^1 \cos^2(2\pi n x) dx = \frac{1}{2},$$

$$\int_0^1 \cos^4(2\pi n x) dx = \frac{3}{8},$$

obtenemos una ecuación bicuadrática para A ,

$$\frac{\lambda}{4} A^4 \frac{(2\pi n)^4}{N^3} \cdot \frac{3}{8} + \frac{\kappa}{2} A^2 \frac{(2\pi n)^2}{N} \cdot \frac{1}{2} - e = 0,$$

que se puede resolver fácilmente,

$$(3.2.1) \quad A(e) = \frac{N}{\pi n} \sqrt{\frac{-1 + \sqrt{\kappa^2 + \frac{6\lambda e}{N}}}{3\lambda}}.$$

La fórmula anterior nos permite introducir en un modo normal arbitrario la energía e que deseemos, pero debemos asegurarnos después de que la dinámica es ergódica para esa energía. Este problema se puede resolver calculando los coeficientes de una serie de Fourier discreta para las posiciones de la cadena con el objetivo de determinar la amplitud de cada uno de los modos normales. Los detalles se pueden consultar en el código fuente del programa de simulación (página 59). Estas amplitudes nos permiten calcular la energía potencial almacenada en un modo normal de vibración concreto, y podemos hacer un seguimiento del reparto de la energía entre los diferentes modos a lo largo de toda la simulación. En el rango de energías observado, tras haber excitado el modo fundamental ($n = 1$), la energía se repartía de manera aproximadamente equilibrada entre todos los modos superado un cierto tiempo t_{eq} . Se permitió que las cadenas de osciladores evolucionaran aisladamente durante este tiempo, de modo que quedarán en un microestado típico del equilibrio (ver figura 3.2.1). Para $t > t_{eq}$, se pusieron en contacto las cadenas y se observó la evolución posterior de las energías.

Para hacerse una idea del comportamiento dinámico del estado de equilibrio final, basta ejecutar la simulación con la temperatura de los dos sistemas igual a la temperatura final. Esto permite estimar el tamaño de las fluctuaciones en la energía de cada subsistema.

3.3. Intercambio de energía

Si pretendemos utilizar la aproximación markoviana, no podemos poner en contacto las cadenas de osciladores simplemente uniendo los extremos, debido a que se formarán los gradientes mencionados en el ejemplo de las esferas al principio de este capítulo. El acoplamiento entre sistemas debe ser, en algún sentido, global, como en las simulaciones de gases de esferas duras. El esquema escogido finalmente se ilustra en la figura 3.3.1. Los osciladores se emparejaban con un oscilador de

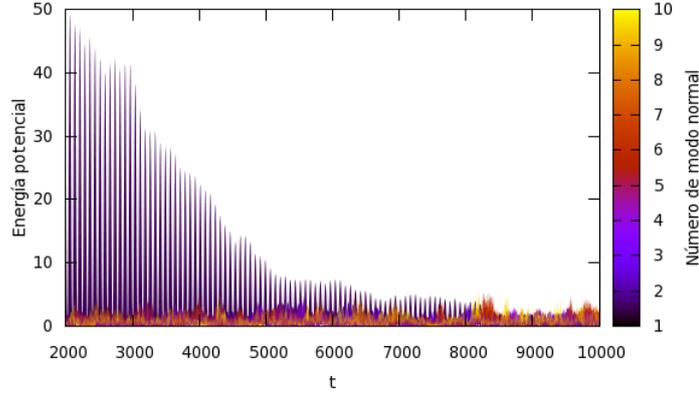


FIGURA 3.2.1. Energía potencial almacenada en los primeros diez modos normales de vibración en función del tiempo, en una cadena de Fermi-Pasta-Ulam de 256 osciladores con condiciones de contorno periódicas. La energía inicial, $e = 512$, se utilizó para excitar únicamente el modo fundamental ($n = 1$). La evolución dinámica del sistema distribuye esta energía globalmente entre los modos.

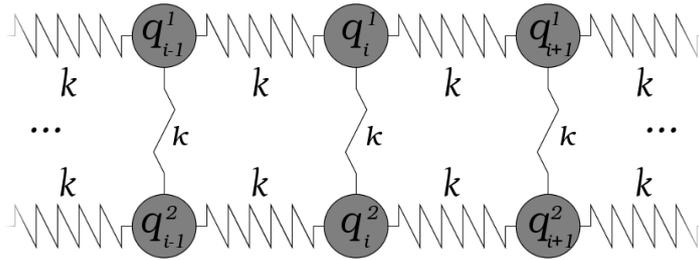


FIGURA 3.3.1. Dos cadenas de Fermi-Pasta-Ulam en contacto. Se supone que las masas de todos los osciladores son iguales a m . La posición del oscilador i -ésimo de la cadena j (1 o 2) se indica con la variable q_i^j . Cada oscilador de la primera cadena está en contacto con un oscilador de la segunda cadena por medio de un muelle lineal débil. Para las simulaciones, se escogieron condiciones de contorno periódicas, luego $q_0 = q_N$ y $p_0 = p_N$.

la otra cadena por medio de muelles lineales débiles (de constante $k = 10^{-3}\kappa$) de modo que pudiera despreciarse la energía de interacción frente a las energías de cada una de las cadenas.

La evolución de las energías en las simulaciones fue de todo menos monótona. La figura 3.3.2 muestra dos ejemplos. Evidentemente, existe una tendencia a alcanzar el equilibrio pasado un tiempo muy largo, pero el proceso que lleva a este estado es muy diferente del que encontramos en el ejemplo de la mezcla de gases, al final del capítulo anterior.

El cálculo de los promedios restringidos $\langle \mathcal{L}H^1(t) \cdot \mathcal{L}H^1(t-t') \rangle$ confirmó la sospecha de que las energías de las cadenas de Fermi-Pasta-Ulam no podían considerarse lentas comparadas con las frecuencias características de vibración de las masas (ver figura 3.3.3). El pico que se ve a la izquierda de la figura coincide aproximadamente con el periodo de oscilación fundamental del sistema armónico.

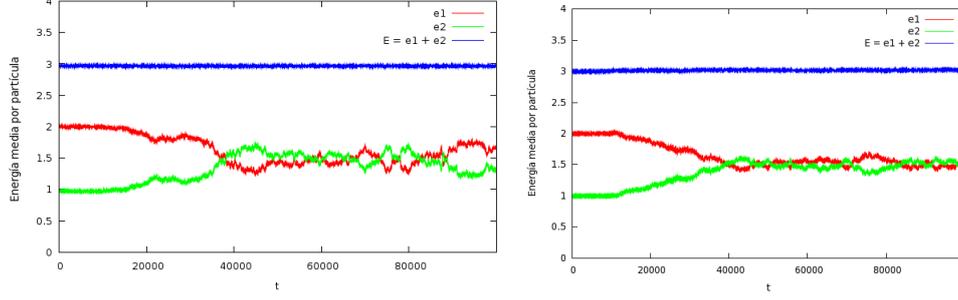


FIGURA 3.3.2. Evolución temporal de las energías almacenadas en dos cadenas de Fermi-Pasta-Ulam puestas en contacto mediante muelles débiles a partir de $t_{eq.} = 10000$. A la izquierda, se muestra el sistema para 65 masas, a la derecha para 256.

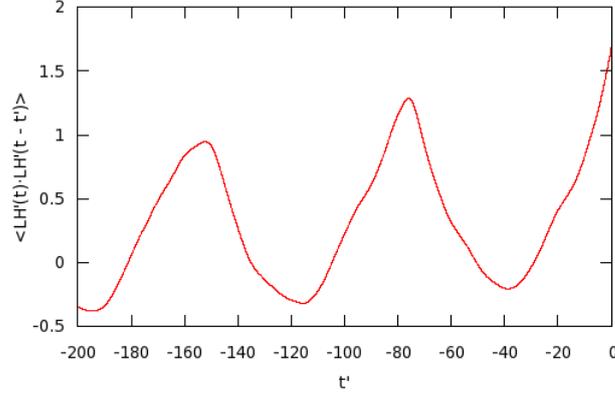


FIGURA 3.3.3. Representación del promedio restringido $\langle \mathcal{LH}^1(t) \cdot \mathcal{LH}^1(t - t') \rangle^e$ en función de t' para $e_1 = 512$ y $e_2 = 256$. Se aprecia muy claramente que no hay un decaimiento rápido a cero, por lo que la hipótesis de variables lentas resulta no ser válida. Nótese que la representación habitual de estos promedios es una imagen especular de la que se ve aquí.

3.4. El trabajo macroscópico

Tal y como anticipaba el principio de este capítulo, el equilibrado no se produce a la manera de la aproximación markoviana debido a la presencia de trabajo macroscópico. Dejemos de lado por un momento la perturbación anarmónica en el potencial de los osciladores. Las cadenas, ahora armónicas, unidas por muelles lineales débiles se comportan en cada modo normal como péndulos acoplados (ver figura 3.4.1). Las ecuaciones del movimiento para la pareja de masas i -ésima son

$$\begin{cases} \ddot{q}_i^1 = -\omega_n^2 q_i^1 - \frac{k}{m} (q_i^1 - q_i^2), \\ \ddot{q}_i^2 = -\omega_n^2 q_i^2 - \frac{k}{m} (q_i^2 - q_i^1), \end{cases}$$

siendo ω_n la frecuencia característica del modo n -ésimo. Utilizando el cambio de variables $x_1 = q_i^1 + q_i^2$, $x_2 = q_i^1 - q_i^2$, obtenemos las nuevas ecuaciones

$$\begin{cases} \ddot{x}_1 = -\omega_n^2 x_1, \\ \ddot{x}_2 = -\left(\omega_n^2 + 2\frac{k}{m}\right) x_2. \end{cases}$$

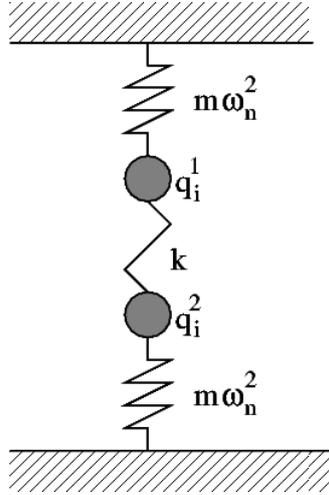


FIGURA 3.4.1. La interacción entre cadenas de osciladores armónicos se comporta en cada modo normal como una pareja de osciladores acoplados mediante un muelle lineal débil de constante k . Cada pareja de masas en contacto tiene una frecuencia característica de vibración ω_n , que puede representarse mediante un muelle lineal de constante $m\omega_n^2$.

Este sistema tiene una solución sencilla,

$$\begin{cases} x_1 = A \exp \{i\omega_n t\}, \\ x_2 = B \exp \left\{ i\sqrt{\omega_n^2 + 2\frac{k}{m}} t \right\}. \end{cases}$$

Deshaciendo el cambio de variables,

$$\begin{cases} q_i^1 = \frac{x_1 + x_2}{2} = \frac{A}{2} \exp \{i\omega_n t\} + \frac{B}{2} \exp \left\{ i\sqrt{\omega_n^2 + 2\frac{k}{m}} t \right\}, \\ q_i^2 = \frac{x_1 - x_2}{2} = \frac{A}{2} \exp \{i\omega_n t\} - \frac{B}{2} \exp \left\{ i\sqrt{\omega_n^2 + 2\frac{k}{m}} t \right\}. \end{cases}$$

Podemos sacar factor común en la expresión anterior,

$$\begin{cases} q_i^1 = \exp \{i\omega_n t\} \left(\frac{A}{2} + \frac{B}{2} \exp \left\{ i\sqrt{1 + 2\frac{\omega_k^2}{\omega_n^2}} t \right\} \right), \\ q_i^2 = \exp \{i\omega_n t\} \left(\frac{A}{2} - \frac{B}{2} \exp \left\{ i\sqrt{1 + 2\frac{\omega_k^2}{\omega_n^2}} t \right\} \right), \end{cases}$$

donde $\omega_k^2 = \frac{k}{m}$, para ver que las coordenadas tienen un movimiento oscilatorio de frecuencia ω_n , pero cuya amplitud oscila con frecuencia $\sqrt{\omega_n^2 + 2\omega_k^2}$.

Los modos normales más lentos son más efectivos en la transferencia de energía de una cadena a otra, debido a que, para una misma cantidad de energía, las amplitudes son mayores (3.2.1), y esto implica que se tensan más los muelles de contacto entre cadenas. Las idas y venidas de energía entre cadenas en los modos bajos produce cambios apreciables en la energía total de cada cadena. Si ha de cumplirse la hipótesis de variables lentas, los periodos asociados a estos intercambios de energía deben ser mucho menores que el tiempo característico τ que tarda el sistema no lineal en distribuir la energía de un modo normal entre otros. Es decir,

la aproximación markoviana será inviable a no ser que

$$\frac{2\pi}{\sqrt{\omega_n^2 + 2\omega_k^2}} \ll \tau,$$

que implica

$$\omega_n^2 + 2\omega_k^2 \ll \frac{4\pi^2}{\tau^2}.$$

Evidentemente, esta condición no se cumplía en las simulaciones que se llevaron a cabo. Sin embargo, aunque es relativamente fácil hacer más pequeño el miembro de la izquierda en la inecuación, aumentando el número de masas (para hacer que las frecuencias de los modos más bajos sean más lentas) y disminuyendo el valor de la constante k , el principal problema consiste en que el valor de τ es elevado (en las simulaciones era del orden de 10^3), y que aumenta con el número de masas. Si quisiéramos forzar el comportamiento markoviano, nos veríamos obligados entonces a reducir drásticamente el valor de τ aumentando la intensidad de la contribución anarmónica en el potencial y esto implicaría a su vez simular un sistema completamente diferente del que se pretendía simular (en el que la “pequeña” contribución anarmónica servía sólo para garantizar la ergodicidad de la dinámica). Otra posibilidad sería la de utilizar muelles no lineales en el contacto entre sistemas, pero en este caso tendríamos un problema parecido. El desarrollo teórico se llevó a cabo suponiendo que podía despreciarse la energía almacenada en estos muelles. Si esto es así, entonces los muelles de contacto se encontrarán siempre en el régimen lineal y la contribución no lineal será inapreciable.

Nos vemos obligados a concluir, por tanto, que el sistema que pretendíamos simular simplemente no se comporta de manera markoviana y que si queremos describirlo con variables macroscópicas y leyes markovianas, en forma de ecuaciones diferenciales, tendremos que introducir variables macroscópicas que representen el trabajo entre sistemas.

En el segundo capítulo encontramos una ley para describir la evolución dinámica del intercambio energético entre sistemas, y una técnica para determinar los valores del tensor de difusión a partir de la dinámica microscópica. En este sentido, hemos hallado verdaderas *leyes dinámicas* de la evolución hacia el equilibrio. Sin embargo, como hemos visto en estas páginas finales, estas leyes son aplicables sólo a sistemas en los que se cumpla la hipótesis de variables lentas. En éstos, los subsistemas que intercambian energía están siempre en un estado parecido al que tendrían si se encontraran aislados entre sí y en equilibrio con esas energías internas. En otras palabras, las ecuaciones de evolución para la probabilidad $P(e, t)$ pueden usarse sólo cuando cada subsistema se olvida muy rápidamente de su evolución anterior. A nosotros, que sí tenemos memoria, nos queda la esperanza de que, aunque a todos nos espere el mismo estado final inevitable de olvido, disolución y equilibrio térmico con el ambiente, el camino no sea un progreso monótono hacia ese final, sino el continuo florecer de bellos patrones emergentes. El estudio de estas formas complejas está, naturalmente, más allá de los objetivos de este trabajo.

Bibliografía

- [1] CALLEN, H. B.: *Thermodynamics and an introduction to thermostatistics (second edition)*. Wiley (1985).

Esta obra clásica es el origen de la introducción y los primeros apartados de este trabajo. Callen presenta de manera clara el problema general de la termodinámica y cómo surgen las leyes macroscópicas de simetrías en la dinámica microscópica.

- [2] ESPAÑOL GARRIGÓS, P.: *Mecánica estadística de fluidos complejos fuera de equilibrio*. Libro de texto de la asignatura del mismo nombre del máster de física de los sistemas complejos impartido en la UNED.

La técnica de operadores de proyección del primer capítulo sigue la exposición de Español con algunas pequeñas modificaciones en el enfoque. La naturaleza del problema permitía además una simplificación apreciable al presentar los cálculos en términos de corchetes de Poisson. Esta estrategia concreta de deducir una ecuación de Fokker-Planck macroscópica a partir de la dinámica microscópica mediante operadores de proyección se debe a los trabajos de Zwanzig.

- [3] IZRAILEV, F. M., BERMAN, G. P.: *The Fermi-Pasta-Ulam problem: 50 years of progress*, arXiv.nlin/0411062v3 [nlin.CD] 8 Mar 2005.

Este artículo repasa los principales descubrimientos en el estudio de las cadenas de Fermi, Pasta y Ulam desde el artículo original de 1955 hasta 2005.

Apéndice A. Demostraciones

En las demostraciones siguientes, utilizaremos una notación más compacta para las integrales sobre el espacio de fases y las funciones delta:

$$\begin{aligned}\mathrm{tr}[A] &= \int A(z) dz, \\ \Psi_e(z) &= \prod_{k=1}^n \delta(H^k(z) - e_k).\end{aligned}$$

TEOREMA 1. *El operador \mathcal{P}^\dagger es adjunto de \mathcal{P} , es decir,*

$$(3.4.1) \quad \mathrm{tr}[B\mathcal{P}^\dagger A] = \mathrm{tr}[APB].$$

DEMOSTRACIÓN. Utilizando la definición del operador \mathcal{P}^\dagger (1.3.1) e intercambiando el orden de las integrales,

$$\begin{aligned}\mathrm{tr}[B\mathcal{P}A] &= \mathrm{tr} \left[\rho^{eq} \cdot B \left(\int \Psi_e \frac{1}{\Omega(e)} \left(\int A(z') \Psi_e(z') dz' \right) de \right) \right] \\ &= \mathrm{tr} \left[A \left(\Psi_e \int \frac{1}{\Omega(e)} \left(\int \rho^{eq} \cdot (z) B(z) \Psi_e(z) dz \right) de \right) \right] \\ &= \mathrm{tr}[APB],\end{aligned}$$

de acuerdo con la definición del operador \mathcal{P} (2.2.7). □

COROLARIO 1. *El operador \mathcal{Q}^\dagger es adjunto de \mathcal{Q} .*

DEMOSTRACIÓN. Utilizando la definición de \mathcal{Q}^\dagger (1.3.2) y la propiedad de linealidad de la integral,

$$\begin{aligned}\mathrm{tr}[B\mathcal{Q}^\dagger A] &= \mathrm{tr}[BA] - \mathrm{tr}[B\mathcal{P}^\dagger A] \\ &= \mathrm{tr}[AB] - \mathrm{tr}[APB] \\ &= \mathrm{tr}[A(I - \mathcal{P})B] \\ &= \mathrm{tr}[BQA].\end{aligned}$$

□

TEOREMA 2. *El operador de Liouville deja pasar la colectividad de equilibrio de derecha a izquierda, es decir,*

$$A\mathcal{L}(\rho^{eq} \cdot B) = \rho^{eq} \cdot A\mathcal{L}B$$

DEMOSTRACIÓN. Sustituyendo la expresión del operador de Liouville en términos de derivadas y desarrollando las derivadas de $\rho^{eq} \cdot B$,

$$\begin{aligned}
A\mathcal{L}(\rho^{eq} \cdot B) &= A(z) \sum_{j=1}^n \sum_{i=1}^{N_j} \frac{\partial H(z)}{\partial p_i^j} \frac{\partial}{\partial q_i^j} (\rho^{eq} \cdot (z) B(z)) \\
&\quad - A(z) \sum_{j=1}^n \sum_{i=1}^{N_j} \frac{\partial H(z)}{\partial q_i^j} \frac{\partial}{\partial p_i^j} (\rho^{eq} \cdot (z) B(z)) \\
&= \rho^{eq} \cdot (z) A(z) \sum_{j=1}^n \sum_{i=1}^{N_j} \frac{\partial H(z)}{\partial p_i^j} \frac{\partial}{\partial q_i^j} B(z) \\
&\quad - \rho^{eq} \cdot (z) A(z) \sum_{j=1}^n \sum_{i=1}^{N_j} \frac{\partial H(z)}{\partial q_i^j} \frac{\partial}{\partial p_i^j} B(z) \\
&\quad + A(z) B(z) \sum_{j=1}^n \sum_{i=1}^{N_j} \frac{\partial H(z)}{\partial p_i^j} \frac{\partial}{\partial q_i^j} \rho^{eq} \cdot (z) \\
&\quad - A(z) B(z) \sum_{j=1}^n \sum_{i=1}^{N_j} \frac{\partial H(z)}{\partial q_i^j} \frac{\partial}{\partial p_i^j} \rho^{eq} \cdot (z).
\end{aligned}$$

Teniendo en cuenta la definición del operador de Liouville, vemos que esta expresión se puede simplificar apreciablemente,

$$A\mathcal{L}(\rho^{eq} \cdot B) = \rho^{eq} \cdot A\mathcal{L}B + AB\mathcal{L}\rho^{eq}.$$

Pero como $\mathcal{L}\rho^{eq} = 0$ (1.1.4),

$$A\mathcal{L}(\rho^{eq} \cdot B) = \rho^{eq} \cdot A\mathcal{L}B.$$

□

TEOREMA 3. *El operador \mathcal{P} es autoadjunto respecto de la medida de equilibrio, es decir,*

$$\text{tr}[\rho^{eq} \cdot A\mathcal{P}B] = \text{tr}[\rho^{eq} \cdot B\mathcal{P}A].$$

DEMOSTRACIÓN. Partiendo del hecho de que \mathcal{P} y \mathcal{P}^\dagger son adjuntos, y utilizando las definiciones de \mathcal{P}^\dagger y \mathcal{P} ,

$$\begin{aligned}
\text{tr}[\rho^{eq} \cdot A\mathcal{P}B] &= \text{tr}[B\mathcal{P}^\dagger(\rho^{eq} \cdot A)] \\
&= \text{tr}\left[B\rho^{eq} \int \frac{1}{\Omega(e)} \Psi_e \int \rho^{eq} \cdot (z') A(z') \Psi_e(z') dz de\right] \\
&= \text{tr}[\rho^{eq} \cdot B\mathcal{P}A].
\end{aligned}$$

□

TEOREMA 4. *El operador \mathcal{Q}^\dagger se convierte en el operador \mathcal{Q} cuando la colectividad de equilibrio pasa de derecha a izquierda, es decir,*

$$\text{tr}[A\mathcal{Q}^\dagger(\rho^{eq} \cdot B)] = \text{tr}[\rho^{eq} \cdot A\mathcal{Q}B]$$

DEMOSTRACIÓN. La linealidad de la integral nos permite escribir

$$\begin{aligned}
\text{tr}[A\mathcal{Q}^\dagger(\rho^{eq} \cdot B)] &= \text{tr}[A(I - \mathcal{P}^\dagger)(\rho^{eq} \cdot B)] \\
&= \text{tr}[A\rho^{eq} \cdot B] - \text{tr}[A\mathcal{P}^\dagger(\rho^{eq} \cdot B)] \\
&= \text{tr}[A\rho^{eq} \cdot B] - \text{tr}[\rho^{eq} \cdot B\mathcal{P}A].
\end{aligned}$$

Utilizando el hecho de que el operador \mathcal{P} es autoadjunto respecto de la medida de equilibrio,

$$\begin{aligned} \text{tr} [AQ^\dagger (\rho^{eq} B)] &= \text{tr} [A\rho^{eq} B] - \text{tr} [\rho^{eq} A\mathcal{P}B] \\ &= \text{tr} [(\rho^{eq} B)(I - \mathcal{P})A] \\ &= \text{tr} [\rho^{eq} AQB]. \end{aligned}$$

□

TEOREMA 5. $\text{tr} [A \exp \{Q^\dagger \mathcal{L}\tau\} B \rho^{eq}] = \text{tr} [\rho^{eq} A \exp \{Q\mathcal{L}\tau\} B]$.

DEMOSTRACIÓN. Introduciendo la definición de $\exp \{Q^\dagger \mathcal{L}\tau\}$ y aplicando las propiedades de linealidad de la traza,

$$\begin{aligned} \text{tr} [A \exp \{Q^\dagger \mathcal{L}\tau\} B \rho^{eq}] &= \text{tr} \left[A \left(\sum_{n=0}^{\infty} \frac{1}{n!} (Q^\dagger \mathcal{L}\tau)^n B \rho^{eq} \right) \right] \\ &= \sum_{n=0}^{\infty} \frac{1}{n!} \text{tr} [A (Q^\dagger \mathcal{L}\tau)^n B \rho^{eq}]. \end{aligned}$$

Hemos visto en los teoremas precedentes que \mathcal{L} deja pasar la colectividad de equilibrio de derecha a izquierda, y que Q^\dagger se convierte en Q , luego podemos concluir que

$$\text{tr} [AQ^\dagger (\mathcal{L}\tau) B \rho^{eq}] = \text{tr} [\rho^{eq} A Q (\mathcal{L}\tau) B].$$

Tomando esta identidad como caso base, podemos proceder por inducción completa para convencernos de que

$$\begin{aligned} \text{tr} [AQ^\dagger (\mathcal{L}\tau)^n B \rho^{eq}] &= \text{tr} [AQ^\dagger \mathcal{L} (\mathcal{L}\tau)^{n-1} \tau^n B \rho^{eq}] \\ &= \text{tr} [\rho^{eq} A Q \mathcal{L} (\mathcal{L}\tau)^{n-1} \tau^n B] \\ &= \text{tr} [\rho^{eq} A Q (\mathcal{L}\tau)^n B], \end{aligned}$$

de lo que se sigue la ecuación que queríamos probar

$$\begin{aligned} \text{tr} [A \exp \{Q^\dagger \mathcal{L}\tau\} B \rho^{eq}] &= \sum_{n=0}^{\infty} \frac{1}{n!} \text{tr} [\rho^{eq} A Q (\mathcal{L}\tau)^n B] \\ &= \text{tr} \left[\rho^{eq} A \sum_{n=0}^{\infty} \frac{1}{n!} (Q\mathcal{L}\tau)^n B \right] \\ &= \text{tr} [\rho^{eq} A \exp \{Q\mathcal{L}\tau\} B]. \end{aligned}$$

□

TEOREMA 6. *El operador de Liouville es hermítico respecto de la medida de equilibrio,*

$$\text{tr} [\rho^{eq} A\mathcal{L}B] = -\text{tr} [\rho^{eq} B\mathcal{L}A].$$

DEMOSTRACIÓN. En primer lugar, calcularemos las dos derivadas siguientes (en seguida se verá por qué),

$$\begin{aligned} \frac{\partial}{\partial q_i^j} \left(\rho^{eq} \frac{\partial H}{\partial p_i^j} AB \right) &= \frac{\partial \rho^{eq}}{\partial q_i^j} \frac{\partial H}{\partial p_i^j} AB + \rho^{eq} \frac{\partial H}{\partial q_i^j \partial p_i^j} AB \\ &\quad + \rho^{eq} \frac{\partial H}{\partial p_i^j} \frac{\partial A}{\partial q_i^j} B + \rho^{eq} \frac{\partial H}{\partial p_i^j} A \frac{\partial B}{\partial q_i^j}, \\ \frac{\partial}{\partial p_i^j} \left(\rho^{eq} \frac{\partial H}{\partial q_i^j} AB \right) &= \frac{\partial \rho^{eq}}{\partial p_i^j} \frac{\partial H}{\partial q_i^j} AB + \rho^{eq} \frac{\partial H}{\partial p_i^j \partial q_i^j} AB \\ &\quad + \rho^{eq} \frac{\partial H}{\partial q_i^j} \frac{\partial A}{\partial p_i^j} B + \rho^{eq} \frac{\partial H}{\partial q_i^j} A \frac{\partial B}{\partial p_i^j}. \end{aligned}$$

Integramos la primera ecuación respecto de q_i^j y la segunda respecto de p_i^j , extendiendo la integral a todo el espacio de fases. Los términos de la izquierda de las dos ecuaciones se anulan, dado que, al estar normalizada la distribución ρ^{eq} , $\lim_{\|z\| \rightarrow \infty} \rho^{eq}(z) = 0$.

$$\begin{aligned} \int_{-\infty}^{\infty} \frac{\partial \rho^{eq}}{\partial q_i^j} \frac{\partial H}{\partial p_i^j} AB + \rho^{eq} \frac{\partial H}{\partial q_i^j \partial p_i^j} AB + \rho^{eq} \frac{\partial H}{\partial p_i^j} \frac{\partial A}{\partial q_i^j} B + \rho^{eq} \frac{\partial H}{\partial p_i^j} A \frac{\partial B}{\partial q_i^j} dq_i^j &= 0, \\ \int_{-\infty}^{\infty} \frac{\partial \rho^{eq}}{\partial p_i^j} \frac{\partial H}{\partial q_i^j} AB + \rho^{eq} \frac{\partial H}{\partial p_i^j \partial q_i^j} AB + \rho^{eq} \frac{\partial H}{\partial q_i^j} \frac{\partial A}{\partial p_i^j} B + \rho^{eq} \frac{\partial H}{\partial q_i^j} A \frac{\partial B}{\partial p_i^j} dp_i^j &= 0. \end{aligned}$$

Integrando ahora sobre p_i^j en la primera ecuación y sobre q_i^j en la segunda, y sumando sobre todos los valores de i y j , obtenemos

$$\begin{aligned} \sum_{j=1}^n \sum_{i=1}^{N_j} \int \frac{\partial \rho^{eq}}{\partial q_i^j} \frac{\partial H}{\partial p_i^j} AB + \rho^{eq} \frac{\partial H}{\partial q_i^j \partial p_i^j} AB + \rho^{eq} \frac{\partial H}{\partial p_i^j} \frac{\partial A}{\partial q_i^j} B + \rho^{eq} \frac{\partial H}{\partial p_i^j} A \frac{\partial B}{\partial q_i^j} dz &= 0, \\ \sum_{j=1}^n \sum_{i=1}^{N_j} \int \frac{\partial \rho^{eq}}{\partial p_i^j} \frac{\partial H}{\partial q_i^j} AB + \rho^{eq} \frac{\partial H}{\partial p_i^j \partial q_i^j} AB + \rho^{eq} \frac{\partial H}{\partial q_i^j} \frac{\partial A}{\partial p_i^j} B + \rho^{eq} \frac{\partial H}{\partial q_i^j} A \frac{\partial B}{\partial p_i^j} dz &= 0. \end{aligned}$$

Restando la segunda ecuación de la primera, e introduciendo la definición de \mathcal{L} obtenemos (intercambiando el signo integral con el de sumatorio)

$$(3.4.2) \quad \int (AB\mathcal{L}\rho^{eq} + \rho^{eq}B\mathcal{L}A - \rho^{eq}A\mathcal{L}B) dz = 0.$$

Ahora bien, según la ecuación de Liouville (1.1.4),

$$AB\mathcal{L}\rho^{eq} = 0,$$

y separando la integral (3.4.2) en dos y despejando la primera, obtenemos

$$\text{tr}[\rho^{eq}A\mathcal{L}B] = -\text{tr}[\rho^{eq}B\mathcal{L}A].$$

□

Apéndice B. Código fuente

Gas de esferas duras

El programa siguiente calcula mediante dinámica molecular la evolución de las energías internas de dos gases de esferas duras mezclados homogéneamente, pero cuyas temperaturas iniciales son diferentes. El microestado inicial se escoge asignando posiciones y velocidades aleatorias a cada partícula de gas. Las posiciones se escogen utilizando una distribución homogénea (evitando que haya superposiciones de esferas) y las velocidades siguen distribuciones de Maxwell-Boltzmann para sendas temperaturas iniciales.

Durante la simulación, se calcula el tiempo de colisión entre cada pareja de partículas y se selecciona la colisión siguiente. Se avanza hasta ese instante y se calculan las nuevas velocidades de las partículas suponiendo que tiene lugar un choque perfectamente elástico. Este procedimiento se repite hasta concluir el tiempo designado para la simulación (indicado por el parámetro SIMULATION_TIME).

Activando la opción -D, el programa calcula las autocorrelaciones necesarias para determinar el tensor de difusión

$$\mathcal{D} \approx \tau \langle (\mathcal{L}H^j(t)) (\mathcal{L}H^j(t-t')) \rangle^e$$

y el tiempo libre medio del sistema τ , es decir, el valor medio del tiempo entre dos colisiones sucesivas.

```
/******                hsg.c                *****  
  
Molecular dynamics simulation of a uniform  
mixture of two hard-sphere gases. The  
program calculates the collision times of  
pairs of particles and steps forward to  
the next collision, calculating the new  
velocities assuming a perfectly elastic  
collision.  
  
When the -D option is activated, the program  
calculates the self-correlations of the  
variations of energy at a given pair of  
energies for the gases. */  
  
/***** Library calls *****/  
  
#include <stdio.h>  
#include <math.h>  
#include <stdlib.h>  
#include <time.h>  
#include <string.h>
```

```

/***** Simulation parameters *****/

#define NR 1000 /* Number of red particles */
#define NB 1000 /* Number of blue particles */
#define TR 2.0f /* Initial temperature of system 1 */
#define TB 0.1f /* Initial temperature of system 2 */
#define MASS 1.0f /* Default mass of gas particles */
#define RADIUS 1.0f /* Default radius of gas particles */
#define SIMULATION_TIME 100 /* Total simulation time */
#define BOX_L 100.0f /* Linear size of system */
#define TAU 0.1f /* Maximum time step in simulation */

#define MEMORY 100 /* Estimated memory of the dynamics */
#define S_CORR_STEP 0.001f /* Length of self-correlation time step */
#define S_CORR_TIME 0.2f /* Time range in the self-correlation graph */

#define PI 3.1415927f

/***** Auxiliary functions (declaration) *****/

/* Random value from a uniform distribution */
float uniform(float min, float max);

/* Random number from an exponential distribution */
float exponential(float lambda);

/* Determine the collision time for particles 1
   and 2 collide during the time interval (0, tau) */
double collisiont(double x1[], double v1[], double r1,
                 double x2[], double v2[], double r2,
                 float tau);

/***** Main program *****/

int main(int argc, char * argv[])
{
    /* Output data files */
    FILE * param;
    FILE * meso;
    FILE * scorr;

    char output[40], filename[40]; /* Filenames */

    /* Simulation variables */
    double x[NR + NB][3]; /* Positions */
    double v[NR + NB][3]; /* Velocities */
    double m[NR + NB]; /* Masses */
    double r[NR + NB]; /* Radii */
    int N = NR + NB; /* Total number of particles */
    /* Temp variables: speeds */
    double speed, old_v1, old_v2, new_v1, new_v2, tmp_v[3];
    double theta, phi; /* Temp variables: reference frame angles */
    double tmp_x; /* Temp variable: position */
    double v_cm[3]; /* centre of mass velocity */

```

```

/* Time, time step and mean free time of the system */
double t, dt, mft;
double tmp_dt; /* Temp variable: time increment */
int p1 = 0, p2 = 0; /* Particle indices */
int i, j, time_counter; /* Counters */
int test; /* Flag */
double e1, e2; /* Energies */
double Le[MEMORY]; /* Energy variations... */
double coll_dt[MEMORY]; /* ... and corresponding collision dt */
/* Self correlation function */
double sc[(int) (S_CORR_TIME/S_CORR_STEP)];

/* Help message */
if(argc < 2)
{
    printf("Use: %s [-D] <output_label>\n", argv[0]);
    printf("Use the -D option to calculate self-correlations_
of the energy variations.\n");
    return 0;
}
else if(argc < 3)
    strcpy(output, argv[1]);
else
{
    if(strcmp(argv[1], "-D"))
    {
        printf("Unrecognized option: %s\n", argv[1]);
        return -1;
    }
    else
        strcpy(output, argv[2]);
}

/* Open data files */
param = fopen(strcat(strcpy(filename, output),
                    ".param.dat"), "w");
meso = fopen(strcat(strcpy(filename, output),
                    ".energy.dat"), "w");

/* Parameter info */
fprintf(param, "#----- File %s.param.dat -----#\n", output);
fprintf(param, "#\n# Simulation of a mixture of two hard-sphere gases.\n");
fprintf(param, "# Data stored in %s.energy.dat.\n#\n"
        "#-- Simulation parameters --#\n#\n", output);
fprintf(param, "# NR = %d (number of red particles).\n", NR);
fprintf(param, "# NB = %d (number of blue particles).\n", NB);
fprintf(param, "# TR = %f (Initial temperature of the red gas).\n", TR);
fprintf(param, "# TB = %f (Initial temperature of the blue gas).\n", TB);
fprintf(param, "# N = %d (total number of particles).\n", N);
fprintf(param, "# MASS = %f (default mass).\n", MASS);
fprintf(param, "# RADIUS = %f (default radius).\n", RADIUS);
fprintf(param, "# SIMULATION_TIME = %d (total simulation time).\n",
        SIMULATION_TIME);
fprintf(param, "# BOX_L = %f (linear size of system).\n", BOX_L);
fprintf(param, "# TAU = %f (maximum time step in simulation).\n", TAU);
fprintf(param, "# MEMORY = %d (estimated memory of the dynamics"
        " measured in mean free time units).\n", MEMORY);

```

```

/* Gnuplot settings */
fprintf(param, "#\n#\n#-----_GNUPLOT_SETTINGS_-----#\n\n");

if(strcmp(argv[1], "-D"))
{
    fprintf(param, "set term png\nset xlabel \"t\"\n");
    fprintf(param, "\n#_Mean_kinetic_energy\n");
    fprintf(param, "set title \"Mean_kinetic_energy_vs._time_\"
        \"in_a_mixture_of_two\"");
    fprintf(param, "\\n_hard-sphere_gases\\n\nset ylabel \"
        \"Mean_kinetic_energy\"\n");
    fprintf(param, "set output \"%s.energy.png\"\n", output);
    fprintf(param, "plot [0:%d][0:%f] \"%s.energy.dat\"_u_1:($2/%d)_
        \"w_l_l_c_1_title_\"Red_gas\",,
        SIMULATION_TIME, TR/.75, output, NR);
    fprintf(param, "\" %s.energy.dat\"_u_1:($3/%d)_w_l_l_c_3_
        \"title_\"Blue_gas\"\n", output, NB);
    printf("Simulation_in_progress...\n");
}
else
{
    fprintf(param, "set term png\nset xlabel \"-t'\n");
    fprintf(param, "\n#_Self-correlation_of_the_variation_\"
        \"of_energy_vs._time\n");
    fprintf(param, "set title \"Self-correlation_functions_for_\"
        \"the_variation_of\"");
    fprintf(param, "\\n_energy_vs._time\\n\nset ylabel \"
        \"<Le(t)·Le(t-t')>\"\n");
    fprintf(param, "set output \"%s.scorr.png\"\n", output);
    printf("Calculating_self-correlation_functions...\n");
}

/* Data header */
fprintf(meso, "#-----_File:%s.energy.dat_-----#\n", output);
fprintf(meso, "#\n#\n_Simulation_data_for_a_mixture_of_two_\"
    \"hard-sphere_gases:\n#\n");
fprintf(meso, "#_t_._time\n#_e1_._energy_of_system_1\n\"
    \"#_e2_._energy_of_system_2\n");
fprintf(meso, "#_E_._total_energy_(e1+_e2)\n#\n");
fprintf(meso, "#_t_\\t_._e1_\\t_._e2_\\t_._E_#\n\"
    \"#_\\t_\\t_-----\\t_\\t_-----\\t_\\t_-----#\n");

/***** Set up simulation *****/

/* Seed random number generator */
srand(time(NULL));

/* Set clock to zero */
t = 0; time_counter = 0; mft = 0;

/* Clear Le and coll_dt arrays */
for(j = 0; j < MEMORY; j++)
    Le[j] = coll_dt[j] = 0;

```

```

/* Define positions, speeds, radii, masses and tau */

/* Red particles */
for(i = 0; i < NR; i++)
{
  m[i] = MASS;
  r[i] = RADIUS;

  /* The positions are random (uniform distribution)*/
  do
  {
    test = 0;

    x[i][0] = uniform(0, BOX_L);
    x[i][1] = uniform(0, BOX_L);
    x[i][2] = uniform(0, BOX_L);

    /* Particles should not overlap */
    for(j = 0; j < i; j++)
      if(sqrt( (x[i][0] - x[j][0])*(x[i][0] - x[j][0])\
              + (x[i][1] - x[j][1])*(x[i][1] - x[j][1])\
              + (x[i][2] - x[j][2])*(x[i][2] - x[j][2]))\
              < r[i] + r[j])
          test = 1;
    } while (test);

    /* The velocities are random (exponential distribution) */
    speed = sqrt(2*exponential(1/TR)/m[i]);
    theta = uniform(0, PI);
    phi = uniform(0, 2*PI);

    v[i][0] = speed*cos(theta)*cos(2*PI);
    v[i][1] = speed*cos(theta)*sin(2*PI);
    v[i][2] = speed*sin(theta);
  }

/* Blue particles */
for(i = NR; i < N; i++)
{
  m[i] = MASS;
  r[i] = RADIUS;

  /* The positions are random (uniform distribution)*/
  do
  {
    test = 0;

    x[i][0] = uniform(0, BOX_L);
    x[i][1] = uniform(0, BOX_L);
    x[i][2] = uniform(0, BOX_L);

    /* Particles should not overlap */
    for(j = 0; j < i; j++)
      if(sqrt( (x[i][0] - x[j][0])*(x[i][0] - x[j][0])\
              + (x[i][1] - x[j][1])*(x[i][1] - x[j][1])\
              + (x[i][2] - x[j][2])*(x[i][2] - x[j][2]))\
              < r[i] + r[j])

```

```

        test = 1;
    } while (test);

    /* The velocities are random (exponential distribution) */
    speed = sqrt(2*exponential(1/TB)/m[i]);
    theta = uniform(0, PI);
    phi = uniform(0, 2*PI);

    v[i][0] = speed*cos(theta)*cos(2*PI);
    v[i][1] = speed*cos(theta)*sin(2*PI);
    v[i][2] = speed*sin(theta);
}

/* Output initial energies */
e1 = 0; e2 = 0;

for(i = 0; i < NR; i++)
    e1 += m[i]*(v[i][0]*v[i][0] + v[i][1]*v[i][1] + v[i][2]*v[i][2])/2;

for(i = NR; i < N; i++)
    e2 += m[i]*(v[i][0]*v[i][0] + v[i][1]*v[i][1] + v[i][2]*v[i][2])/2;

fprintf(meso, "0\t\t%f\t%f\t%f\n", e1, e2, e1 + e2);

/**** Begin simulation ****/

while(t <= SIMULATION_TIME)
{
    /* Set the time step to TAU, in case there are no collisions */
    dt = TAU;

    /* Find the next collision time in the interval (0, TAU) */
    for(i = 0; i < N - 1; i++)
    {
        for(j = i + 1; j < N; j++)
        {
            tmp_dt = collisiont(x[i], v[i], r[i], x[j], v[j], r[j], TAU);

            /* Set dt to the time of the earliest future collision */
            if(tmp_dt < dt && tmp_dt > 0)
            {
                p1 = i;
                p2 = j;
                dt = tmp_dt;
            }
        }
    }

    /* Increase time in dt */
    t += dt;
    time_counter++;

    /* Calculate new positions of the particles */
    for(i = 0; i < N; i++)
    {
        for(j = 0; j < 3; j++)
        {

```

```

x[i][j] += v[i][j]*dt;

/* Periodic boundary conditions */
while(x[i][j] >= BOX_L)
    x[i][j] -= BOX_L;
while(x[i][j] < 0)
    x[i][j] += BOX_L;
}
}

/* If there was a collision, calculate the new speeds */
if(dt < TAU)
{
    /* Calculate new velocities of the particles after the collision */

    /* Velocity of the centre of mass of the two colliding particles */
    for(i = 0; i < 3; i++)
        v_cm[i] = (v[p1][i]*m[p1] + v[p2][i]*m[p2])/(m[p1] + m[p2]);

    /* Speed of particles in the reference frame of the centre of mass */
    old_v1 = sqrt( (v[p1][0]-v_cm[0])*(v[p1][0]-v_cm[0])\
        + (v[p1][1]-v_cm[1])*(v[p1][1]-v_cm[1])\
        + (v[p1][2]-v_cm[2])*(v[p1][2]-v_cm[2]));

    old_v2 = -sqrt( (v[p2][0]-v_cm[0])*(v[p2][0]-v_cm[0])\
        + (v[p2][1]-v_cm[1])*(v[p2][1]-v_cm[1])\
        + (v[p2][2]-v_cm[2])*(v[p2][2]-v_cm[2]));

    /* New velocities in the new reference frame */
    new_v1 = (2*m[p2]*old_v2 - old_v1*(m[p2] - m[p1]))/(m[p1] + m[p2]);
    new_v2 = (2*m[p1]*old_v1 - old_v2*(m[p1] - m[p2]))/(m[p1] + m[p2]);

    /* Angles of rotation between the new reference frame and the old one */
    if(sqrt( (x[p2][0]-x[p1][0])*(x[p2][0]-x[p1][0])\
        + (x[p2][1]-x[p1][1])*(x[p2][1]-x[p1][1])) == 0)
        theta = PI/2;
    else
        theta = atan((x[p2][2] - x[p1][2])\
            /sqrt( (x[p2][0]-x[p1][0])*(x[p2][0]-x[p1][0])\
                + (x[p2][1]-x[p1][1])*(x[p2][1]-x[p1][1])));

    phi = atan2(x[p2][1] - x[p1][1], x[p2][0] - x[p1][0]);

    /* If the "-D" option is not activated or p1 and p2 belong to the
       same system... */
    if(strcmp(argv[1], "-D") || !(p1 < NR && p2 >= NR))
    {
        /* ... then undo the change of reference frame, ... */
        v[p1][0] = new_v1*cos(theta)*cos(phi) + v_cm[0];
        v[p1][1] = new_v1*cos(theta)*sin(phi) + v_cm[1];
        v[p1][2] = new_v1*sin(theta) + v_cm[2];

        v[p2][0] = new_v2*cos(theta)*cos(phi) + v_cm[0];
        v[p2][1] = new_v2*cos(theta)*sin(phi) + v_cm[1];
        v[p2][2] = new_v2*sin(theta) + v_cm[2];
    }
    else /* ... otherwise, swap their positions. */

```

```

{
  /* (This allows each subsystem to keep its energy without
     changing the macrostate). */
  for(j = 0; j < 3; j++)
  {
    tmp_x = x[p1][j];
    x[p1][j] = x[p2][j];
    x[p2][j] = tmp_x;
  }
}

/* Calculate self-correlation functions ("-D" activated) */
if(!strcmp(argv[1], "-D"))
{
  /* Move Le(t-t') back in time */
  for(i = MEMORY - 1; i > 0; i--)
    Le[i] = Le[i - 1];

  /* Keep track of the dt that separates each collision
     from the next one */
  for(i = MEMORY - 1; i > 0; i--)
    coll_dt[i] = coll_dt[i - 1] + dt;

  /* Calculate the energy variation for each subsystem */
  tmp_v[0] = new_v1*cos(theta)*cos(phi) + v_cm[0];
  tmp_v[1] = new_v1*cos(theta)*sin(phi) + v_cm[1];
  tmp_v[2] = new_v1*sin(theta) + v_cm[2];

  Le[0] = (m[p1]/2)*( tmp_v[0]*tmp_v[0] + tmp_v[1]*tmp_v[1]\
                      + tmp_v[2]*tmp_v[2]\
                      - v[p1][0]*v[p1][0] - v[p1][1]*v[p1][1]\
                      - v[p1][2]*v[p1][2]);

  /* If the time counter is greater than MEMORY, then
     calculate the self-correlations of the Le array. */
  if(time_counter > MEMORY)
  {
    for(i = 0; i < (S_CORR_TIME/S_CORR_STEP); i++)
    {
      sc[i] = sc[i]*(1 - 1/(S_CORR_TIME/S_CORR_STEP));
    }

    for(i = 0; i < MEMORY; i++)
    {
      j = (int) (coll_dt[i]/S_CORR_STEP);

      if(j < (int) (S_CORR_TIME/S_CORR_STEP))
        sc[j] += Le[0]*Le[i] / (S_CORR_TIME/S_CORR_STEP);
    }
  }
}

/* Update the estimation of the mean free time of the system */
mft = ((time_counter - 1)*mft + dt)/time_counter;
}

if(strcmp(argv[1], "-D"))

```

```

{
  /* Output energies */
  e1 = 0; e2 = 0;

  for(i = 0; i < NR; i++)
    e1 += m[i]*(v[i][0]*v[i][0] + v[i][1]*v[i][1] + v[i][2]*v[i][2])/2;

  for(i = NR; i < N; i++)
    e2 += m[i]*(v[i][0]*v[i][0] + v[i][1]*v[i][1] + v[i][2]*v[i][2])/2;

  fprintf(meso, "%f\t%f\t%f\n", t, e1, e2, e1 + e2);
}

printf("[-- %.1f %--]\r", 100*t/SIMULATION_TIME);
}

/* If "-D" was activated, output self-correlation functions */
if(!strcmp(argv[1], "-D"))
{
  scorr = fopen(strcat(strcpy(filename, output), ".scorr.dat"), "a");

  fprintf(scorr, "#----- File: %s.scorr.dat -----#\n#\n", output);
  fprintf(scorr, "# Self-correlation functions for the "
    "variations in energy\n");
  fprintf(scorr, "# for a mixture of two hard sphere gases.\n#\n");
  fprintf(scorr, "# Mean free time of the system = %f\n", mft);
  fprintf(scorr, "# e1 = Energy of system 1 (red gas) = %f.\n", NR*TR);
  fprintf(scorr, "# t' = time.\n#\n");
  fprintf(scorr, "# t' \t \t <Le1(t)·Le1(t-t')>#\n");
  fprintf(scorr, "# -----\t\t-----#\n");

  for(i = 0; i < (int) (S_CORR_TIME/S_CORR_STEP); i++)
    fprintf(scorr, "%f\t%f\n", -i*S_CORR_STEP, sc[i]);

  fprintf(param, "plot [-%f:0][ ] %s.scorr.dat \t u_1:2_w_1_l_c_1"
    " title \"Red gas\"\n", S_CORR_TIME, output);

  fclose(scorr);
}

printf("\nDone.\n");

/* Close data files */
fclose(meso);
fclose(param);

return 0;
}

/***** Auxiliary functions *****/

/* Random value from a uniform distribution */
float uniform(float min, float max)
{
  return min + (max - min)*random()/RAND_MAX;
}

```

```

/* Random value from an exponential distribution */
float exponential(float lambda)
{
    return -log(1 - uniform(0, .999999))/lambda;
}

/* Determine whether particles 1 and 2 collide during the
time interval (0, tau) (the function returns the time
of collision if there is one, and -1 otherwise).

The basic idea is that the collision occurs at a time t
such that


$$\|(x1 + v1 \cdot t) - (x2 + v2 \cdot t)\| = (r1 + r2)$$

*/

double collisiont(double x1[], double v1[], double r1,
                 double x2[], double v2[], double r2,
                 float tau)
{
    double a, b, c, d, t1, t2;

    /* The time of collision is the solution of the equation


$$a \cdot t^2 + b \cdot t + c = 0,$$


where


$$a = \|v1 - v2\|^2,$$


$$b = 2 \cdot (x1 - x2) \cdot (v1 - v2),$$


$$c = \|x1 - x2\|^2 - (r1 + r2)^2;$$

*/

    /* Coefficients a, b and c */
    a = (v1[0] - v2[0])*(v1[0] - v2[0])\
        + (v1[1] - v2[1])*(v1[1] - v2[1])\
        + (v1[2] - v2[2])*(v1[2] - v2[2]);
    b = 2*( (x1[0] - x2[0])*(v1[0] - v2[0])\
            + (x1[1] - x2[1])*(v1[1] - v2[1])\
            + (x1[2] - x2[2])*(v1[2] - v2[2]));
    c = (x1[0] - x2[0])*(x1[0] - x2[0])\
        + (x1[1] - x2[1])*(x1[1] - x2[1])\
        + (x1[2] - x2[2])*(x1[2] - x2[2])\
        - (r1 + r2)*(r1 + r2);

    /* If particles are moving away from each other, return -1 */
    if(b >= 0)
        return -1;

    /* Square of the discriminant */
    d = b*b - 4*a*c;

    if (a > 0 && d >= 0)
    {
        /* Solutions of the quadratic equation */

```

```

t1 = (-b - sqrt(d))/(2*a);
t2 = (-b + sqrt(d))/(2*a);

/* Return the smallest solution in the interval (0, tau) */
if(t1 > 0 && t1 < tau)
    return t1;
else if (t2 > 0 && t2 < tau)
    return t2;
else
    return -1.0;
}
else
    return -1.0;
}

```

Ecuación de Fokker-Planck

Integración numérica de la ecuación de Fokker-Planck (2.4.2) mediante el método de Crank-Nicolson.

```

/*****          cg.c          *****/

Numerical integration of the following Fokker-Planck
equation for the time evolution of P(e, t),

dP(e, t)/dt = d/de [D(e) d/de (P(e, t)/Omega(e))].

This equation was rewritten in the following form:

dP(e, t)/dt = -a(e) P(e, t) + b(e) dP(e, t)/de
              + D(e) (d2/de2)P(e, t).

This expression was then discretized using the
Crank-Nicolson method.

/***** Parameters *****/

#define NR 1000 /* Number of red particles */
#define NB 1000 /* Number of blue particles */
#define TR 1.9f /* Initial temperature of red gas */
#define TB 0.1f /* Initial temperature of blue gas */
#define E 2000 /* Total energy of the system = NR*TR + NB*TB */
#define SIMULATION_TIME 150
#define TAU 0.1f

/***** Library calls *****/

#include <stdio.h>
#include <math.h>

/***** Auxiliary functions *****/

/* time-dependent time step */
double dt(double t);

```

```

/* Functions in the partial differential equation */
double D(double e);
double deltabeta(double e);
double a(double e);
double b(double e);

/***** Main program *****/

int main(int argc, char * argv[])
{
    int e;
    double t;
    double P[E + 1];
    double diag[E + 1], supdiag[E + 1], subdiag[E + 1], ind[E + 1], m;

    /* Initial distribution */
    for(e = 0; e <= E; e++)
        P[e] = 0;
    for(e = 1696; e <= 1705; e++)
        P[e] = .1;

    /* Numerical integration (Crank-Nicolson method) */
    for(t = 0; t <= SIMULATION_TIME; t += dt(t))
    {
        diag[0] = 1;
        supdiag[0] = subdiag[0] = ind[0] = 0;
        for(e = 1; e < E; e++)
        {
            subdiag[e] = TAU*(b(e)/4 - D(e)/2);
            diag[e] = 1 + TAU*(a(e)/2 + D(e));
            supdiag[e] = -TAU*(b(e)/4 + D(e)/2);
            ind[e] = (1 - TAU*(a(e)/2 + D(e)))*P[e]\
                + TAU*(D(e)/2 + b(e)/4)*P[e + 1]\
                + TAU*(D(e)/2 - b(e)/4)*P[e - 1];
        }

        diag[E] = 1;
        supdiag[E] = subdiag[E] = ind[E] = 0;

        /* Solve the tridiagonal linear system */
        for(e = 1; e <= E; e++)
        {
            m = subdiag[e]/diag[e - 1];
            diag[e] -= m*supdiag[e - 1];
            ind[e] -= m*ind[e - 1];
        }

        P[E] = ind[E]/diag[E];

        for(e = E - 1; e >= 0; e--)
            P[e] = (ind[e] - supdiag[e]*P[e + 1])/diag[e];

        /* Output results */
        for(e = 0; e <= E; e++)
            printf("%f\t%f\t%f\n", t, e, P[e]);
        printf("\n");
    }
}

```

```

    }

    return 0;
}

/***** Auxiliary function definitions *****/

/* time-dependent time step */
double dt(double t)
{
    if(t < 3)
        return .5;
    else if (t < 10)
        return .2;
    else
        return TAU;
}

/* Functions in the partial differential equation */
double D(double e)
{
    return 0.00001*e*e - e*.002 + 1.13462;
}

double deltabeta(double e)
{
    return 3*(NR/e - NB/(E - e))/2;
}

double a(double e)
{
    return (0.00002*e - .002)*deltabeta(e)\
        - D(e)*3*(NR/(e*e) + NB/(E*e + e*e - 2*E*e))/2;
}

double b(double e)
{
    return (0.00002*e - .002) - D(e)*deltabeta(e);
}

```

Cadenas de Fermi-Pasta-Ulam

El código que se presenta a continuación simula la evolución temporal de dos cadenas de Fermi-Pasta-Ulam con condiciones de contorno periódicas. Las masas de la primera cadena están conectadas mediante muelles lineales débiles con sendas masas en la segunda. Antes de poner en contacto las cadenas, se deja que evolucionen aisladamente durante un tiempo designado como EQUILIBRIUM_TIME para garantizar que se encuentran inicialmente en un microestado típico del equilibrio.

Cuando el código se ejecuta con la opción `-D`, el programa calcula las autocorrelaciones necesarias para determinar el tensor de difusión en la aproximación markoviana

$$\mathcal{D} \approx \tau \langle (\mathcal{L}H^j(t)) (\mathcal{L}H^j(t-t')) \rangle^e.$$

El análisis de estos datos demuestra que en este caso es incorrecto hacer uso de la hipótesis de variables lentas.

```

/*****                               fpu.c                               *****/

    This program simulates the time evolution of two
    Fermi-Pasta-Ulam chains with periodic boundary
    conditions. Each mass in the first chain is
    connected to a corresponding mass on the second
    chain by means of a linear spring. The chains are
    first allowed to evolve isolated from each other
    to guarantee that they have reached thermal
    equilibrium when they are brought into contact.

    When the "-D" option is active, the program
    calculates the self-correlations of the variations
    of the energy of each chain. */

/***** Library calls *****/

#include <stdio.h>
#include <string.h>
#include <math.h>

/***** Simulation parameters *****/

#define N 256 /* Number of particles per chain */
#define EQUILIBRIUM_TIME 10000
#define SIMULATION_TIME 100000
#define DIFF_TIME 20000 /* Simulation time for the
                           calculation of D(e), (> MEMORY) */
#define MEASUREMENT 400 /* Number of time steps from one
                           measurement to the next (usually 1000) */
#define PI 3.1415927f
#define TAU 0.01f /* Simulation time step */
#define MASS 1.0f /* Default mass */
#define KAPPA 1.0f /* Spring constant */
#define LAMBDA 1.0f /* Fermi-Pasta-Ulam phi-4 lambda parameter */
#define CONTACT_K 0.001f /* Contact spring constant */
#define ENERGY 3.0f /* Total energy per mass */
#define E1 2.0f /* Initial energy per mass in the first subchain */
#define MEMORY 10000 /* Estimated memory of the dynamics */
#define E_STEPS 20 /* Number of energy steps for the calculation
                     of D(e) */

/***** Auxiliary function declarations *****/

/* Output parameter info file */
int parameters(char filename[]);

/* Assign initial positions and speeds to the
   Fermi-Pasta-Ulam chain with periodic boundary conditions */
int initial_conditions(double positions[], double velocities[],
                      double accelerations[], double energy,
                      int mode_n, int delta_n);

/* Velocity-Verlet integration */

```

```

int velocity_verlet(double x1[], double v1[], double a1[],
                   double x2[], double v2[], double a2[], double k);

/* Calculate the energy of a Fermi-Pasta-Ulam chain with periodic
   boundary conditions */
double energy(double positions[], double speeds[]);

/* Calculate the temperature of a Fermi-Pasta-Ulam chain */
double temperature(double v[]);

/* Fourier transform of signal x, stored in g */
int fourier(double x[], double g[]);

/***** Main program *****/

int main(int argc, char * argv[])
{
    /* Data files */
    FILE * micro;
    FILE * meso;
    FILE * macro;
    FILE * diff;
    char output[40], filename[50]; /* Output label and filenames */

    /* Simulation variables */
    double x1[N], x2[N]; /* Displacement from equilibrium positions */
    double v1[N], v2[N]; /* Velocities */
    double a1[N], a2[N]; /* Accelerations */
    double g1[N], g2[N]; /* Fourier transforms */
    double Vn1[N], Vn2[N]; /* Potential energies per mode */
    double D1[MEMORY], D2[MEMORY]; /* Diffusion coefficients */
    double k; /* contact spring constants */
    double t, dt; /* Time variables */
    double e1, e2, delta_e; /* Energies */
    double Le1[MEMORY], Le2[MEMORY]; /* Time derivatives of e1 and e2 */
    int i, j; /* Counters */

    /* Help messages */
    if(argc < 2)
    {
        printf("Use: %s [-D] <output_label>\n", argv[0]);
        printf("When executed with the -D option, the program calculates the diffusion coefficients.\n");
        return 0;
    }

    if(argc == 2)
        strcpy(output, argv[1]);
    else
    {
        if(strcmp(argv[1], "-D"))
        {
            printf("Unrecognized option: %s\n", argv[1]);
            return 0;
        }
        else

```

```

    strcpy(output, argv[2]);
}

**** Initialize simulation data ****

printf("Setting up the simulation...\n");

/* Output simulation info */
printf("Storing the simulation parameters in %s.param.dat...\n",
       output);

i = parameters(output);
if(i == -1)
{
    printf("Error opening parameter file.\n");
    return -1;
}

/* Set clock to zero */
t = 0;

/* Set time-steps counter to zero */
j = 0;

/* Set the time step to TAU */
dt = TAU;

/* Assign initial positions, speeds and accelerations*/
/* Subsystem 1 */
initial_conditions(x1, v1, a1, N*E1, 1, 1);
/* Subsystem 2 */
initial_conditions(x2, v2, a2, N*(ENERGY - E1), 5, 1);

/* Open data files */
printf("Opening data files: %s.micro.dat", output);
micro = fopen(strcat(strcpy(filename, output), ".micro.dat"),
              "w");
printf("%s.meso.dat and", output);
meso = fopen(strcat(strcpy(filename, output), ".meso.dat"),
              "w");
printf("%s.macro.dat.\n", output);
macro = fopen(strcat(strcpy(filename, output), ".macro.dat"),
              "w");

if(micro == NULL || meso == NULL || macro == NULL)
{
    printf("Error: Unable to open data file.\n");
    return -1;
}

/* Labels for simulation data output */
fprintf(micro, "#----- File: %s.micro.dat -----#\n#\n", output);
fprintf(micro, "# Microscopic variables:\n");
fprintf(micro, "# t - time\n# i - particle number\n# xj [i]"
          "\n - position of particle i (subsystem number j)\n");
fprintf(micro, "# Aj[n] - Amplitude of mode n (subsystem number");

```



```

                                energy(x2, v2));
printf("Simulation_in_progress..._[%0.1f%%]\r",
       100*t/SIMULATION_TIME);
}

j++;
t += dt;
}

/* Bring subsystems into contact with each other unless "-D"
   was specified */
if(strcmp(argv[1], "-D"))
{
    k = CONTACT_K;

    /* Simulate the evolution of the system */
    while(t <= SIMULATION_TIME)
    {
        /* Integrate the equations of motion using a Velocity
           Verlet algorithm */
        velocity_verlet(x1, v1, a1, x2, v2, a2, k);

        /* Calculate the energy of each subsystem every
           MEASUREMENT time-steps */
        if(j %MEASUREMENT == 0)
        {
            /* Fourier transforms */
            fourier(x1, g1);
            fourier(x2, g2);

            /* Calculate energies per mode */
            for(i = 0; i < N; i++)
            {
                Vn1[i] = (KAPPA*g1[i]*g1[i]*PI*PI*i*i)/N
                        + (3*LAMBDA*pow(2*PI*i*g1[i], 4))/(32*pow(N, 3));
                Vn2[i] = (KAPPA*g2[i]*g2[i]*PI*PI*i*i)/N
                        + (3*LAMBDA*pow(2*PI*i*g2[i], 4))/(32*pow(N, 3));
            }

            /* Print results */
            fprintf(micro, "#_tstep_=_%d_#\n", j/MEASUREMENT);

            for(i = 0; i < N; i++)
                fprintf(micro, "%f\t%f\t%f\t%f\t%f\t%f\t%f\n", t, i,
                        x1[i], g1[i], Vn1[i], x2[i], g2[i], Vn2[i]);
            fprintf(micro, "\n");
            fprintf(meso, "%f\t%f\t%f\n", t, energy(x1, v1),
                    energy(x2, v2));
            fprintf(macro, "%f\t%f\t%f\n", t, temperature(v1),
                    temperature(v2));

            printf("Simulation_in_progress..._[%0.1f%%]\r",
                  100*t/SIMULATION_TIME);
        }

        j++;
        t += dt;
    }
}

```

```

}

printf("[Simulation_completed]\n");
}
else /* Option "-D" activated. Calculate diffusion coefficients */
{
/* Open diffusion coefficients data file */
printf("\nReached_EQUILIBRIUM_TIME.\n");
printf("Opening_%.diff.dat.\n", output);
diff = fopen(strcat(strcpy(filename, output), ".diff.dat"), "w");

if(diff == NULL)
{
printf("Error_opening_file_%.diff.dat\n", output);
return -1;
}

/* Initial energies */
e1 = E1;
e2 = ENERGY - E1;

/* Calculate energy step */
delta_e = (e1 - e2)/E_STEPS;

/* Clear D arrays */
for(i = 0; i < MEMORY; i++)
{
D1[i] = 0;
D2[i] = 0;
}

while(e1 >= e2)
{
t = 0;

while(t <= DIFF_TIME)
{
/* Integrate the equations of motion using a Velocity
Verlet algorithm */
velocity_verlet(x1, v1, a1, x2, v2, a2, k);

/* Move Lej data back in time */
for(j = MEMORY - 1; j > 0; j--)
{
Le1[j] = Le1[j - 1];
Le2[j] = Le2[j - 1];
}

/* Calculate new variation of energies */
Le1[0] = Le2[0] = 0;

for(i = 0; i < N; i++)
{
Le1[0] += -CONTACT_K*(x1[i] - x2[i])*v1[i];
Le2[0] += -CONTACT_K*(x2[i] - x1[i])*v2[i];
}
}
}

```



```

if(param == NULL)
    return -1;

fprintf(param, "#——_File:_%s_——#\n#\n", output);
fprintf(param, "#_This_file_contains_the_parameters_used_for_
_the_simulation\n");
fprintf(param, "#_of_two_Fermi-Pasta-Ulam_systems_brought_
_into_contact_by\n");
fprintf(param, "#_means_of_linear_springs.The_simulation_
_data_files_are:\n");
fprintf(param, "#\n#1)_%.micro.dat_(positions,_speeds,_
_etc.)\n", filename);
fprintf(param, "#_2)_%.meso.dat_(total_energies_of_each_
_subsystem)\n", filename);
fprintf(param, "#_3)_%.macro.dat_(mean_kinetic_energy_per_
_particle_of_each\n#_subsystem\n", filename);
fprintf(param, "#\n#\n#——_SIMULATION_PARAMETERS_——#\n#\n");
fprintf(param, "#_N=_%d_(Number_of_particles_per_chain)\n", N);
fprintf(param, "#_EQUILIBRIUM_TIME=_%d\n", EQUILIBRIUM_TIME);
fprintf(param, "#_SIMULATION_TIME=_%d\n", SIMULATION_TIME);
fprintf(param, "#_DIFF_TIME=_%d\n", DIFF_TIME);
fprintf(param, "#_MEASUREMENT=_%d_(time_steps_to_the_next_
_measurement)\n", MEASUREMENT);
fprintf(param, "#_TAU=_%f_(simulation_time_step)\n", TAU);
fprintf(param, "#_MASS=_%f_(each_of_the_masses_in_the_
_simulation)\n", MASS);
fprintf(param, "#_KAPPA=_%f_(linear_spring_constant)\n", KAPPA);
fprintf(param, "#_LAMBDA=_%f_(non-linear_cubic_force_constant)\n",
LAMBDA);
fprintf(param, "#_CONTACT_K=_%f_(contact_spring_constant)\n",
CONTACT_K);
fprintf(param, "#_ENERGY=_%f_(total_energy_per_mass)\n", ENERGY);
fprintf(param, "#_E1=_%f_(initial_energy_per_mass_in_the_first_
_subsystem)\n", E1);
fprintf(param, "#_E2=_%f_(initial_energy_per_mass_in_the_second_
_subsystem)\n", ENERGY - E1);
fprintf(param, "#_MEMORY=_%d_(estimated_memory_of_the_dynamics)\n",
MEMORY);
fprintf(param, "#_E_STEPS=_%d_(Number_of_energy_steps_when_
_calculating_D(e))\n", E_STEPS);
fprintf(param, "#\n#\n#——_GNUPLOT_SETTINGS_——#\n#\n");
fprintf(param, "set_term_png\nset_xlabel_t\n");
fprintf(param, "\n#_Macroscopic_temperature\n");
fprintf(param, "set_title_Temperatures\n");
fprintf(param, "set_ylabel_temperature\n");
fprintf(param, "set_output_%s.macro.png\n", filename);
fprintf(param, "plot_./%s.macro.dat\%u_1:2_w_1_title_T1",
filename);
fprintf(param, "\%u_1:3_w_1_title_T2",
filename);
fprintf(param, "\%s.macro.dat\%u_1:((%2+$3)/2)_w_p_7_ps_2"
_title_mean_temperature\n", filename);
fprintf(param, "\n#_Coarse-grained_mesoscopic_information\n");
fprintf(param, "set_title_Energy_of_subsystems_1_and_2_versus_
_time\n");
fprintf(param, "set_ylabel_energy\n");
fprintf(param, "set_output_%s.meso.png\n", filename);

```

```

fprintf(param, "plot \\.\/%s.meso.dat\"_u_1:2_w_l_t title \"e1\",",
        filename);
fprintf(param, "\.\/%s.meso.dat\"_u_1:3_w_l_t title \"e2\",",
        filename);
fprintf(param, "\.\/%s.meso.dat\"_u_1:($2+_3)_w_l_t title "
        "\"E_e1+_e2\"\\n", filename);
fprintf(param, "\n#_Microscopic_information\\n");
fprintf(param, "set_title \"Energy_distribution_among_modes"
        "\"_versus_time_in_subsystem_1\"\\n");
fprintf(param, "set_ylabel \"n\"\\n");
fprintf(param, "set_xlabel \"energy\"\\n");
fprintf(param, "set_pm3d_map\\n");
fprintf(param, "set_output \\.\/%s.micro.modes1.png\"\\n", filename);
fprintf(param, "splot [[0:%d][0:%f] \\.\/%s.micro.dat\"_u"
        "\"_1:2:5_notitle\\n\\n", N/2 - 1, ENERGY, filename);
fprintf(param, "set_title \"Energy_distribution_among_modes_"
        "\"_versus_time_in_subsystem_2\"\\n");
fprintf(param, "set_output \\.\/%s.micro.modes2.png\"\\n", filename);
fprintf(param, "splot [[0:%d][0:%f] \\.\/%s.micro.dat\"_u"
        "\"_1:2:8_notitle\\n", N/2 - 1, ENERGY, filename);

fclose(param);
return 0;
}

/* Assign initial positions and speeds to the Fermi–Pasta–Ulam
   chain with periodic boundary conditions.

   The initial conditions are chosen in the following way:


$$x[i] = A1 \sin(2\pi n i/N) + A2 \sin(2\pi (n+1) i/N) + \dots$$


$$+ \text{Adelta}_n \sin(2\pi (n + \text{delta}_n) i/N),$$


$$v[i] = 0,$$


   where  $n$  is an integer. */
int initial_conditions(double x[], double v[], double a[],
                     double energy, int mode_n, int delta_n)
{
    double A; /* Amplitude of the first normal mode */
    int i, j; /* Counters */

    /* Choose the amplitude in such a way that you get the right
       energy */
    for(i = 0; i < N; i++)
    {
        for(j = 0; j < delta_n; j++)
        {
            A = (N/(mode_n*PI))*sqrt((-1
                + sqrt(1 + 6*energy/(delta_n*N)))/3.0);
            x[i] = A*(sin(2*PI*(mode_n + j)*i/N));
        }

        v[i] = 0;
    }

    /* Calculate the initial accelerations */
    a[0] = -(KAPPA/MASS)*(2*x[0] - x[1] - x[N - 1])\
            -(LAMBDA/MASS)*(pow(x[0] - x[1], 3)

```

```

+ pow(x[0] - x[N - 1], 3));

for (i = 1; i < N - 1; i++)
  a[i] = -(KAPPA/MASS)*(2*x[i] - x[i + 1] - x[i - 1])\
          -(LAMBDA/MASS)*(pow(x[i] - x[i + 1], 3)
          + pow(x[i] - x[i - 1], 3));

a[N - 1] = -(KAPPA/MASS)*(2*x[N - 1] - x[0] - x[N - 2])\
            -(LAMBDA/MASS)*(pow(x[N - 1] - x[0], 3)
            + pow(x[N - 1] - x[N - 2], 3));

return 0;
}

/* Velocity Verlet integration */
int velocity_verlet(double x1[], double v1[], double a1[],
                   double x2[], double v2[], double a2[], double k)
{
  double a_old1[N], a_old2[N]; /* Previous accelerations */
  int i; /* Counter */

  /* First, store the old accelerations */
  for (i = 0; i < N; i++)
  {
    a_old1[i] = a1[i];
    a_old2[i] = a2[i];
  }

  /* Calculate the new accelerations */
  /* Subsystem 1 */
  a1[0] = -(KAPPA/MASS)*(2*x1[0] - x1[1] - x1[N - 1])\
          -(LAMBDA/MASS)*(pow(x1[0] - x1[1], 3)
          + pow(x1[0] - x1[N - 1], 3))\
          -(k/MASS)*(x1[0] - x2[0]);

  for (i = 1; i < N - 1; i++)
    a1[i] = -(KAPPA/MASS)*(2*x1[i] - x1[i + 1] - x1[i - 1])\
            -(LAMBDA/MASS)*(pow(x1[i] - x1[i + 1], 3)
            + pow(x1[i] - x1[i - 1], 3))\
            -(k/MASS)*(x1[i] - x2[i]);

  a1[N - 1] = -(KAPPA/MASS)*(2*x1[N - 1] - x1[0] - x1[N - 2])\
              -(LAMBDA/MASS)*(pow(x1[N - 1] - x1[0], 3)
              + pow(x1[N - 1] - x1[N - 2], 3))\
              -(k/MASS)*(x1[N - 1] - x2[N - 1]);

  /* Subsystem 2 */
  a2[0] = -(KAPPA/MASS)*(2*x2[0] - x2[1] - x2[N - 1])\
          -(LAMBDA/MASS)*(pow(x2[0] - x2[1], 3)
          + pow(x2[0] - x2[N - 1], 3))\
          -(k/MASS)*(x2[0] - x1[0]);

  for (i = 1; i < N - 1; i++)
    a2[i] = -(KAPPA/MASS)*(2*x2[i] - x2[i + 1] - x2[i - 1])\
            -(LAMBDA/MASS)*(pow(x2[i] - x2[i + 1], 3)
            + pow(x2[i] - x2[i - 1], 3))\
            -(k/MASS)*(x2[i] - x1[i]);

```

```

a2[N - 1] = -(KAPPA/MASS)*(2*x2[N - 1] - x2[0] - x2[N - 2])\
            -(LAMBDA/MASS)*(pow(x2[N - 1] - x2[0], 3)\
            + pow(x2[N - 1] - x2[N - 2], 3))\
            -(k/MASS)*(x2[N - 1] - x1[N - 1]);

/* Calculate the new speeds */
for(i = 0; i < N; i++)
{
    v1[i] += (a1[i] + a_old1[i])*TAU/2;
    v2[i] += (a2[i] + a_old2[i])*TAU/2;
}

/* Calculate the new positions */
for(i = 0; i < N; i++)
{
    x1[i] += v1[i]*TAU + a1[i]*TAU*TAU/2;
    x2[i] += v2[i]*TAU + a2[i]*TAU*TAU/2;
}

return 0;
}

/* Calculate the energy of a Fermi-Pasta-Ulam chain with periodic
boundary conditions */
double energy(double x[], double v[])
{
    double K, V, Va; /* Kinetic and potential energies */
    int i; /* Counter */

    /* Kinetic energy */
    K = 0;

    for(i = 0; i < N; i++)
        K += MASS*(v[i]*v[i])/2;

    /* Linear potential energy */
    V = (KAPPA/2)*(x[0] - x[N - 1])*(x[0] - x[N - 1]);

    for(i = 1; i < N; i++)
        V += (KAPPA/2)*(x[i] - x[i - 1])*(x[i] - x[i - 1]);

    /* Non-linear potential energy */
    Va = (LAMBDA/4)*pow(x[0] - x[N - 1], 4);

    for(i = 1; i < N; i++)
        Va += (LAMBDA/4)*pow(x[i] - x[i - 1], 4);

    return K + V + Va;
}

/* Calculate the temperature of a Fermi-Pasta-Ulam chain */
double temperature(double v[])
{
    int i;
    double T = 0;

```

```

for (i = 0; i < N; i++)
{
    T += (MASS*v[i]*v[i])/N;
}

return T;
}

/* Fourier transform of signal x, stored in g */
int fourier(double x[], double g[])
{
    double s[N], c[N];
    int i, j;

    /* N (size of the x and g arrays) must be defined! */
    for (i = 0; i < N; i++)
    {
        s[i] = 0;
        c[i] = 0;

        for (j = 0; j < N; j++)
        {
            s[i] += 2*x[j]*cos(2*i*PI*j/N)/N;
            c[i] += 2*x[j]*sin(2*i*PI*j/N)/N;
        }
    }

    for (i = 0; i < N; i++)
        g[i] = sqrt(s[i]*s[i] + c[i]*c[i]);

    return 0;
}

```