



Departamento de Física
UNIVERSIDADE DE AVEIRO

Modelação em Física Estatística

2018.06.21

Exame

1. Considera um gás de fotões que se encontram numa caixa cúbica de lado L a uma temperatura T .

a) (3 valores) Deduz uma expressão para o número de estados com energia entre ϵ e $\epsilon + d\epsilon$, $g(\epsilon)d\epsilon$.

Considerando a unidade de energia $u_\epsilon = \frac{\hbar c}{L}$ mostra que se pode escrever $g(\epsilon)d\epsilon = \frac{\epsilon'^2}{2\pi^2} d\epsilon'$, onde

$\epsilon' = \epsilon/u_\epsilon$. Ignora as diferentes polarizações possíveis dos fotões que introduziriam um fator 2 nos resultados.

Resolução:

O número de estados com um dado módulo do vetor de onda é: $g(k)dk = \frac{4\pi k^2}{\left(\frac{2\pi}{L}\right)^3} dk = \frac{V}{2\pi^2} k^2 dk$. Como a

energia dos estados é dada por $\epsilon_k = \hbar c k$ o número de estados com energia entre ϵ e $\epsilon + d\epsilon$ é

$$g(\epsilon)d\epsilon = g(k)dk = \frac{V}{2\pi^2} k^2 \frac{d\epsilon}{\frac{d\epsilon}{dk}} \text{ e portanto } g(\epsilon)d\epsilon = \frac{V}{2\pi^2 (\hbar c)^3} \epsilon^2 d\epsilon. \text{ Em termos da energia expressa}$$

em unidades de $u_\epsilon = \frac{\hbar c}{L}$ podemos escrever,

$$g(\epsilon')d\epsilon' = g(\epsilon)u_\epsilon d\epsilon/u_\epsilon = \frac{V}{2\pi^2 (\hbar c)^3} \epsilon'^2 d\epsilon' u_\epsilon^3 = \frac{1}{2\pi^2} \epsilon'^2 d\epsilon'.$$

b) (2 valores) Obtém uma expressão para o número médio de fotões na caixa a uma temperatura T , $\langle N \rangle$.

Resolução: O número de fótons é dado por: $N = \int_0^\infty \frac{g(\epsilon') d\epsilon'}{e^{\epsilon'/T'} - 1}$. Substituindo $g(\epsilon') d\epsilon' = \frac{\epsilon'^2}{2\pi^2} d\epsilon'$

obtem-se $N = \frac{1}{2\pi^2} \int_0^\infty \frac{\epsilon'^2 d\epsilon'}{e^{\epsilon'/T'} - 1}$. Fazendo a substituição, $x = \epsilon'/T'$ temos $N = \frac{T'^3}{2\pi^2} \int_0^\infty \frac{x^2 dx}{e^x - 1}$.

Como $\int_0^\infty \frac{x^2 dx}{e^x - 1} = 2.4041138$ obtém-se $N = \frac{T'^3}{\pi^2} 1.2020569$ ou ainda $N = \frac{k_B^3 T^3}{u_E^3 \pi^2} 1.2020569$.

c) **(2 valores)** Obtém uma expressão para a energia total média do gás de fótons a uma temperatura T, $\langle E \rangle$.

Resolução: A energia total é dada por: $E' = E/u_E = \int_0^\infty \frac{\epsilon' g(\epsilon') d\epsilon'}{e^{\epsilon'/T'} - 1}$. Fazendo a substituição das variáveis

obtemos $E' = \frac{T'^4}{2\pi^2} \int_0^\infty \frac{x^3 dx}{e^x - 1} = \frac{\pi^2}{30} T'^4$ ou $E/u_E = \frac{k_B^4 \pi^2}{30 u_E^4} T^4$.

d) **(3 valores)** Um plano preto (não-reflector) a temperatura fixa T_a é paralelo a outro plano que se encontra a uma temperatura fixa T_b , com $T_a > T_b$. O fluxo efectivo de energia entre os dois planos no vácuo é $J_0 = \sigma_{SB}(T_a^4 - T_b^4)$. Um terceiro plano preto é inserido entre os dois e atinge uma temperatura estacionária T_1 . Calcula a temperatura T_1 em termos de T_a e T_b . Calcula o fluxo de energia, J_1 entre os planos na situação final. Considera agora que foram colocados, entre os planos a temperatura T_a e T_b , não um plano mas N planos. Qual o fluxo de energia J_N entre os planos? Qual a dependência de T_i , com $i = 1, \dots, N$, nas temperaturas T_a e T_b ?

Resolução:

Se colocarmos um plano entre os dois planos a temperatura T_a e T_b o fluxo de energia entre os planos é

$J_1 = \sigma_{SB}(T_a^4 - T_1^4) = \sigma_{SB}(T_1^4 - T_b^4)$. Resolvendo para $T_1^4 = \frac{1}{2}(T_a^4 + T_b^4)$ e substituindo na expressão para

J_1 obtém-se $J_1 = \frac{\sigma_{SB}}{2} (T_a^4 - T_b^4) = J_0/2$. Se colocarmos dois planos obtemos:

$T_a^4 - T_1^4 = T_1^4 - T_2^4 = T_2^4 - T_b^4$. Resolvendo para T_1^4 e T_2^4 obtém-se $T_1^4 = \frac{2}{3} T_a^4 + \frac{1}{3} T_b^4$ e

$T_2^4 = \frac{1}{3} T_a^4 + \frac{2}{3} T_b^4$. O fluxo de energia entre os planos é $J_2 = \sigma_{SB} (T_a^4 - T_1^4) = J_0/3$. Se tivermos N planos

temos $J_N = \frac{J_0}{N+1}$. A expressão geral para as temperaturas dos planos no caso de N planos vem dada por:

$$T_i^4 = \frac{N-i}{N+1} T_a^4 + \frac{i}{N+1} T_b^4, i=1, \dots, N$$

Nota: $\int_0^\infty \frac{x^{p-1} dx}{e^x - 1} = \zeta(p) \Gamma(p)$ onde $\zeta(p)$ é a função zeta de Riemann que toma valores

$\zeta(3) = 1.2020569 \dots$ e $\zeta(4) = \frac{\pi^4}{90}$ e $\Gamma(p)$ é a função gama que toma valores $\Gamma(3) = 2$

$\Gamma(4) = 6$.

2. Considera um gás de fótons que se encontram numa caixa cúbica de lado L a uma temperatura T. Efetua todos os cálculos usando a unidade de energia $u_\epsilon = \frac{\hbar c}{L}$ e a unidade de temperatura $u_T = u_\epsilon / k_B$.

a) **(3 valores)** Cria uma função **function [nestados,energias]=DensidadeEstadosFotoes(nmax)** cujo input é o máximo número quântico (em módulo), em cada direção do espaço, para as componentes do vetor de onda (com condições fronteira periódicas) e cujo output é o número total de estados, **nestados** e um vetor, **energias**, com as energias dos estados. A variável **nmax** foi chamada **nmin** em versões anteriores desta função para partículas materiais. Compara o resultado graficamente com a expressão fornecida na alínea 1. a) para o número de estados com energia entre ϵ' e $\epsilon' + d\epsilon'$.

Resolução: Ver função. Deve ter-se em atenção que $\epsilon_{\vec{k}} = \hbar c k$, com $\vec{k} = \frac{2\pi}{L} (n_x, n_y, n_z)$. Na unidade de

energia $u_\epsilon = \frac{\hbar c}{L}$ a energia de um estado calcula-se através de **energias(estado)=2*pi*sqrt(nx^2+ny^2+nz^2)**

e a expressão teórica da densidade de estados é **gteor=x.^2/(2*pi^2)**; onde x representa valores da energia de cada estado em unidades de u_ϵ .

b) **(3 valores)** O estado de um gás de fótons é completamente especificado pelo número de fótons em cada estado. Seja **np** um vetor de tamanho **nestados** que armazena o número de fótons que se encontra em cada estado. Para fazer

uma atualização de Monte Carlo, deste vetor, com o algoritmo de Metropolis a uma temperatura T, pode usar-se a função:

```
function [np]=atualiza_np_estado(np,estado,T,energias)
    dn=2*(randi(2)-1)-1;
    if (np(estado)==0)
        dn=1;
    end
    dE=energias(estado)*dn;
    if (np(estado)>0)
        pa=min([1,exp(-dE/T)]);
        if(np(estado)==1 && dn==-1)
            pa=min([1,2*exp(-dE/T)]);
        end
    else
        pa=min([1,0.5*exp(-dE/T)]);
    end
    if rand(1) < pa
        np(estado)=np(estado)+dn;
    end
end
```

Explica o algoritmo usado na função. Em particular explica o que representa **dn**, o que representa **pa** e porque tem o valor **pa=min([1,exp(-dE/T)])**; se **np(estado)>1** e **pa=min([1,0.5*exp(-dE/T)])**; se **np(estado)=0**.

Resolução:

A função usa o algoritmo de Metropolis para atualizar o número de fótons de um dado estado. A variável **dn=2*(randi(2)-1)-1**; toma o valor +/- 1 com igual probabilidade. Esta variável representa uma proposta de aumentar ou diminuir o número de fótons. A variação do número de fótons causa uma variação de energia, **dE=energias(estado)*dn**. Quando o estado não tem fótons o número de fótons só pode aumentar e **dn=1**. No algoritmo de Metropolis a probabilidade de aceitar o estado **f** do sistema, a partir do estado inicial **i**, obedece à

equação: $P_A(f \leftarrow i) = \min \left(1, \frac{P_p(i \leftarrow f) P_{st}(f)}{P_p(f \leftarrow i) P_{st}(i)} \right)$ onde $P_p(i \leftarrow f)$ é a probabilidade de propôr **i** a partir de

f, $P_p(f \leftarrow i)$ é a probabilidade de propôr **f** a partir de **i**, $P_{st}(f) = \exp(-\beta E_f)/Z$ é a probabilidade

estacionária do estado final e $P_{st}(i) = \exp(-\beta E_i)/Z$ é a probabilidade estacionária do estado inicial. Para

todos os estados com número de fótons superior a 1 a probabilidade de propôr **dn=1** é igual à probabilidade de propôr **dn=-1** e $P_p(i \leftarrow f)/P_p(f \leftarrow i) = 1$. Nesse caso,

$P_A(f \leftarrow i) = \min \left(1, \frac{P_{st}(f)}{P_{st}(i)} \right) = \min(1, \exp(-\beta dE))$. A probabilidade de aceitar o estado sem fótons a

a partir do estado com 1 fóton obedece, no entanto, a uma expressão diferente dado que $P_p(0 \leftarrow 1) = 1/2$ e

$P_p(1 \leftarrow 0) = 1$. Nesse caso, $P_A(0 \leftarrow 1) = \min \left(1, \frac{1}{2} \exp(-\beta dE) \right)$. A probabilidade de aceitar o estado

com 1 fóton a partir do estado com 0 fótons é $P_A(1 \leftarrow 0) = \min(1, 2 \exp(-\beta dE))$.

c) **(2 valores)** Faz uma função, **function [energias,np_med,emed2,nestados]=GasFotoes(T,tmax,nmax)** que simula durante um tempo **1.1*tmax**, um gás de fótons a temperatura **T** considerando estados até $\pm nmax$ e que devolve um vetor com as **energias** dos estados, o número médio de fótons em cada estado, **np_med**, o valor médio da energia ao quadrado, **emed2**, e o número de estados, **nestados**. Nesta função, em cada instante de tempo, todos os estados são atualizados sequencialmente, usando a função **atualiza_estado** mas, o estado de energia nula cujo índice é obtido de **i0=find(energias==0)**; tem sempre zero fótons (**é muito importante incluir esta restrição que consiste em não atualizar o número de fótons no estado de energia nula**). No início da simulação não há fótons no sistema. As médias efetuadas para obter as variáveis **np_med** e **emed2** são efetuadas para **t>0.1*tmax** de forma a desprezar 10% dos passos de Monte Carlo iniciais para equilibração do sistema.

Resolução: Ver ficheiro com a função GasFotoesv1.m Num sistema de partículas materiais podemos fixar o número de partículas mas num sistema de fótons que tem potencial químico nulo o número de fótons varia ao longo da simulação. Por isso, não é necessário, não é conveniente, nem seria suficiente, fazer uma atualização do número de partículas nos estados de uma partícula considerando apenas mudanças de partículas entre estados vizinhos, como se fez para uma gás de partículas materiais em que o número total de partículas se conserva (Nota que o tamanho máximo das listas que especificam os estados com partículas teria que ser estimado). Para amostrar os estados do sistema é suficiente percorrer todos os estados de uma partícula e atualiza-los usando a função **atualiza_estado** que simplesmente cria e destrói fótons em cada estado.

```
for t=1:1.1*tmax
for estado=1:nestados
    if (estado ~=i0)
        [np]=atualiza_np_estado(np,estado,T,energias);
    end
end
(...)
end
end
```

d) **(1 valor)** Obtém resultados de simulações para temperaturas entre 2 e 4 em passos de 0.05 com **nmin=5**; e **tmax=1000**; Calcula para cada temperatura o número médio de fótons presentes no sistema, a energia total, e a capacidade térmica a volume constante fazendo uma representação gráfica destas quantidades em função da temperatura. Compara os resultados numéricos com as correspondentes expressões analíticas a que se referem as deduções pedidas 1. b) e 1. c).

Resolução: Ver ficheiro com o programa GasFotoes.m

e) **(1 valor)** Faz uma função **[nE]=planck(npmed,energias, E)** que calcula a distribuição de Planck, isto é, o número médio de fótons, por unidade de energia, com uma dada energia, **nE**, para valores de energia especificados pelo vetor **E=0:dE:Emax**;

Resolução:

```
function [nE]=planck(npmed,energias, E)
dE=(E(2)-E(1));
np=length(E);
nE=zeros(np,1);
for ip=2:np
    i=find(energias>E(ip-1) & energias <= E(ip));
    nE(ip)=sum(npmed(i))/dE;
end
end
```