

Isoterme per i gas di Bose e di Fermi

L. P.

La funzione di partizione dei gas di Bose e Fermi nell'ensemble gran canonico è data dall'espressione (4.72)

$$\log Z_{GC} = (\pm) \sum_{\mathbf{p}, \sigma} \log \left[1 \pm \exp \left(\frac{\mu - \epsilon(\mathbf{p}, \sigma)}{k_B T} \right) \right]. \quad (1)$$

Il segno superiore vale per i fermioni, quello inferiore per i bosoni. L'energia di singolo stato $\epsilon(\mathbf{p}, \sigma)$ è data da

$$\epsilon(\mathbf{p}, \sigma) = \frac{p^2}{2m}. \quad (2)$$

Nel limite termodinamico otteniamo l'espressione (4.75):

$$\log Z_{GC} = (\pm) \frac{V}{h^3} (2s + 1) \int_0^\infty dp \, 4\pi p^2 \log \left[1 \pm z \exp \left(-\frac{p^2}{2mk_B T} \right) \right], \quad (3)$$

dove h è la costante di Planck ed abbiamo introdotto la fugacità

$$z = e^{\mu/k_B T}. \quad (4)$$

Il fattore $(2s + 1)$ conta il numero di stati che differiscono per il valore dello spin σ . Effettuando il cambio di variabile $x = (p^2 / (2mk_B T))$ otteniamo

$$\log Z_{GC} = \frac{V}{\lambda^3} (2s + 1) (\mp) \phi_{5/2}(\mp z), \quad (5)$$

dove λ è la lunghezza d'onda termica di de Broglie

$$\lambda = \left(\frac{h^2}{2\pi m k_B T} \right)^{1/2}, \quad (6)$$

e $\phi_\alpha(z)$ è la funzione polilog

$$\phi_\alpha(z) = \sum_{k=1}^{\infty} \frac{z^k}{k^\alpha}. \quad (7)$$

Il numero medio di particelle si ottiene calcolando la derivata di questa espressione rispetto a $\log z$. Otteniamo così

$$N = z \frac{\partial \log Z_{GC}}{\partial z} = \frac{V}{\lambda^3} (2s + 1) (\mp) \phi_{3/2}(\mp z). \quad (8)$$

Quindi la densità $\rho = N/V$ dei gas è data in funzione di z e di λ dalla relazione

$$\rho\lambda^3 = (2s + 1)(\mp)\phi_{3/2}(\mp z). \quad (9)$$

D'altra parte si ha, per le proprietà dell'ensemble gran canonico,

$$\log Z_{GC} = \frac{pV}{k_B T}, \quad (10)$$

per cui si ottiene

$$\frac{p\lambda^3}{k_B T} = (2s + 1)(\mp)\phi_{5/2}(\mp z). \quad (11)$$

Possiamo usare queste espressioni per ottenere una rappresentazione parametrica dell'equazione di stato dei gas, in funzione di z . Nel caso dei bosoni, z varia nell'intervallo $[0, 1)$, mentre nel caso dei fermioni, z può essere qualunque numero non negativo. Per $\rho \geq \rho_c$, dove $\rho_c = \zeta_R(3/2)\lambda^{-3}$, il gas di Bose è condensato, con una pressione costante ed uguale a $p_c = \zeta_R(5/2)k_B T/\lambda^3$. Qui $\zeta_R(s)$ è la funzione zeta di Riemann. Per entrambi i tipi di gas, quando $z \ll 1$, si ottiene l'equazione di stato classica

$$\frac{p}{k_B T} = \rho. \quad (12)$$

Questi risultati possono essere valutati mediante l'accluso codice Python. La funzione polilog è valutata mediante la rappresentazione integrale

$$\phi_\alpha(z) = \frac{1}{\Gamma(\alpha)} \int_0^\infty dx x^{\alpha-1} \frac{z e^{-x}}{1 - z e^{-x}}, \quad (13)$$

dove $\Gamma(s)$ è la funzione gamma di Eulero, e che può essere verificata sviluppando l'integrando in serie di z e sommando.

```
# bosefermi.py -- Isotherms for the ideal quantum gas. Lengths are measured
# in units of the de Broglie thermal wavelength lambda=(h^2/(2 pi kB T))^(1/2).
# Energies are measured in units kB T
```

```
from scipy import integrate
from scipy.special import gamma as Gamma
import numpy as np
import matplotlib.pyplot as plt

def integrand(x, z, alpha):
    """Integrand for the polilog """
    w = np.exp(-x)
    return x**(alpha-1)*((z*w)/(1-z*w))

def polilog(z, alpha):
    """Integral representation for the polilog function."""
    return integrate.quad(lambda x: integrand(x, z, alpha),
        0, np.Inf)[0]/Gamma(alpha)
```

```

def pbose(z):
    return polilog(z, 5/2)

def nbose(z):
    return polilog(z, 3/2)

def pfermi(z):
    return -polilog(-z, 5/2)

def nfermi(z):
    return -polilog(-z, 3/2)

# For Bose, the range of z is constrained to [0:1]
z = np.linspace(0, 1, 101)

Nc = 5
pB = np.zeros(len(z)+Nc)
nB = np.zeros(len(z)+Nc)
for i in range(len(z)):
    pB[i] = pbose(z[i])
    nB[i] = nbose(z[i])
pc = pB[len(z)-1]
nc = nB[len(z)-1]
for k in range(Nc): # Isotherm of the condensate
    pB[len(z)+k] = pc
    nB[len(z)+k] = nc+((3.0-nc)/Nc)*k

# For Fermi, z can take any positive value: we choose [0:5]
z = np.linspace(0, 5, 101)

pF = np.zeros(len(z))
nF = np.zeros(len(z))
for i in range(len(z)):
    pF[i]=pfermi(z[i])
    nF[i]=nfermi(z[i])

# In the plot, we also report the classical equation of state
lines = plt.plot(nB, pB, 'r-', nF, pF, 'b-', nB, nB, 'g--')
plt.setp(lines,linewidth=2)
plt.plot((1.75, nbose(1)), pbose(1)*np.ones(2), 'r--') # Critical pressure
plt.xticks([0, 1, 2, nbose(1), 3],
            ['0', '1', '2', r'$\rho_{\mathrm{c}}$', '3'])
plt.yticks([0, 1, 2, pbose(1), 3],
            ['0', '1', '2', r'$p_{\mathrm{c}}$', '3'])
plt.xlabel(r'$\rho\lambda^3$')
plt.ylabel(r'$p\lambda^3/k_{\mathrm{B}}T$')

```

```
plt.legend(["Bose", "Fermi", "Classical"], loc=0)
plt.title("Ideal gas isotherms")
plt.savefig('boseFermi.pdf')
```

