Problem 1. 2-7

Suppose we consider a beam of neutrons incident upon a thin target with an intensity of $10^{12}\frac{neutrons}{cm^2 s}$. Suppose further that the total cross section for the nuclei in this target is $4b$. Using this information, determine how long one would have to wait, on the average, for a given nucleus in the target to suffer a neutron interaction.

Solution

We know that our reaction rate is equal to $intensity * cross\ section$. Our mean time before a reaction is the inverse of the rate:

$$\frac{1}{intensity * cross\ section} = 2.5 * 10^{11} s$$

Problem 2. 2-11

Using the data from BNL-325, compute the mean free paths of neutrons with the following energies in the specified materials:

(a) $14MeV$ neutrons in air, water, and uranium (characteristic of thermonuclear fusion neutrons),

(b) $1MeV$ neutrons in air, water, and uranium (fast breeder reactor neutrons), and

(c) $0.05eV$ neutrons in air, water, and uranium (thermal reactor neutrons).

Solution

Calculations are included as an attachment to this assignment after problem

	$14MeV$	$1MeV$	$0.05eV$
U	$3.54cm$	$2.91cm$	$1.42cm$
Water	$10.04cm$	$1.79cm$	$0.53cm$
Air	$1.27 * 10^2cm$	$5640cm$	$2041cm$

Problem 3. 2-12

Determine the kinetic energy at which the wavelength of a neutron is comparable to:

(a) the diameter of a nucleus,

(b) an atomic diameter,

(c) the interatomic spacing in graphite, and

(d) the diameter of a nuclear reactor core.

(Only rough estimates are required.)

Solution

The deBroglie wavelength of a particle is expressed by:

$$\lambda = \frac{h}{\sqrt{2Tm}}$$

from which we derive:

$$T = \frac{h^2}{2m\lambda^2}$$

Object	λ	T
the diameter of a nucleus	15 fm	3.63 MeV
an atomic diameter	350 pm	$6.68 * 10^{-9} MeV$
the interatomic spacing in graphite	0.142 nm	$4.06 * 10^{-9} MeV$
the diameter of a nuclear reactor core	1 m	$8.18 * 10^{-28} MeV$

Problem 4. 2-15

Using the Maxwell-Boltzman distribution $M(V,T)$, calculate the most probable energy of the nuclei characterized by such a distribution. Also calculate the average thermal energy of these nuclei.

Solution

The Maxwell-Boltzman distribution is expressed as:

$$M(V,T) = \left(\frac{m}{2\pi kT}\right)^{3/2} e^{\frac{-mV^2}{2kT}}$$

Substituting $E = \frac{1}{2}mV^2$, we get:

$$M(E,T) = \sqrt{\frac{4}{\pi}} E^{1/2} \left(\frac{1}{kT}\right)^{3/2} e^{\frac{-E}{kT}}$$

To determine the most likely value (median) we take the derivative and set it equal to zero:

$$\frac{dM}{dE} = 0 = \sqrt{\frac{4}{\pi}} \left(\frac{1}{kT}\right)^{3/2} \left(\frac{1}{2}E^{-1/2}e^{\frac{-E}{kT}} + E^{1/2}\left(\frac{-1}{kT}\right)e^{\frac{-E}{kT}}\right)$$

Which gives $E = \frac{1}{2}kT$.

To calculate the mean of the distribution, we solve the definite integral:

$$\bar{E} = \frac{1}{M}\int_0^\infty M(E,T)EdE$$

This gives a value of $\bar{E} = \frac{3}{2}kT$

Problem 5. 2-20

Determine the fission-rate density necessary to produce a thermal power density of $400kW/liter$ (typical of a fast breeder reactor core). Assume that the principle fissile isotope is $^{239}_{94}$Pu.

Solution

Each fission of $^{239}_{94}$Pu produces $211.5MeV$. Therefore:

$$\begin{aligned}
TPD &= \frac{E}{fission} * FRD \\
FRD &= \frac{TPD}{\frac{E}{fission}} \\
&= \frac{400\frac{J}{s*L}}{211.5\frac{MeV}{fission}/6.2415*10^{12}\frac{MeV}{J}} \\
&= 1.18*10^{16}\frac{fissions}{L*s}
\end{aligned}$$

In [219]:

```
from pyne import data
import numpy as np
from scipy.interpolate import interp1d
import scipy.constants as const
from tabulate import tabulate
```

Problem 2-11

Some utility functions and values

In [220]:

```
def sigma_interpolate(filename, path=None):
    # fix path and filename
    if path is not None:
        filename = path + filename
    # load data into np array
    data = np.loadtxt(filename, delimiter=',', skiprows=1)
    # create the interpolator
    interpolator = interp1d(data[:,0],data[:,1])
    # return the interpolator
    return interpolator
```

In [221]:

```
headers = ["Energy", "Mean free path (cm)"]
```

In [222]:

```
const.Avogadro
```

Out[222]:

```
6.022140857e+23
```

Uranium

In [223]:

```
data.atomic_mass('U235')
```

Out[223]:

```
235.043930131
```

In [224]:

```
data.atomic_mass('U238')
```

Out[224]:

```
238.050788423
```

Uranium density is expressed in units of $\frac{g}{cm^3}$

```
In [225]: U_density = 19.05
```

```
In [226]: data.atomic_mass('U')

Out[226]: 238.0289104847141
```

The concentration of each isotope is expressed by:

$$N_{isotope} = \rho_{element} * \frac{A_v}{M_{element}} * M_{isotope}$$

where A_v is equal to Avogadro's number (`const.Avogadro` in code)

```
In [227]: N_U_238 = U_density * const.Avogadro / data.atomic_mass('U') * data.na
          tural_abund('U238')
          N_U_238

Out[227]: 4.784676466003686e+22
```

```
In [228]: N_U_235 = U_density * const.Avogadro / data.atomic_mass('U') * data.na
          tural_abund('U235')
          N_U_235

Out[228]: 3.472081292127316e+20
```

```
In [229]: path = '/Users/jrpowers-luhn/nucnotes/ne470/homework/02/'
```

```
In [230]: sigma_U_238 = sigma_interpolate('U-238.txt', path=path)
          sigma_U_235 = sigma_interpolate('U-235.txt', path=path)
```

```
In [231]: res = []
          for E in [14*10**6, 10**6, 0.05]:
              Sigma = N_U_235 * sigma_U_235(E) / 10**24 + N_U_238 * sigma_U_238(
          E) / 10**24
              res.append((E, Sigma))
```

```
In [232]: res = []
          for E in [14*10**6, 10**6, 0.05]:
              Sigma = N_U_235 * sigma_U_235(E) / 10**24 + N_U_238 * sigma_U_238(
          E) / 10**24
              lamb = 1 / Sigma
              res.append((E, lamb))
          print tabulate(res, headers=headers)
```

```
            Energy    Mean free path (cm)
          --------  ---------------------
           1.4e+07                3.54178
           1e+06                  2.91285
           0.05                   1.42219
```

Water

The density of water is $1.0 \frac{g}{cm^3}$. The particle density of water is simply:

$$N_{H_2O} = \frac{A_v}{M_{H_2O}}$$

where M_{H_2O} is:

$$M_{H_2O} = 2 * M_{Hydrogen} + M_{Oxygen}$$

```
In [233]: data.atomic_mass('H1')

Out[233]: 1.00782503223
```

```
In [234]: data.atomic_mass('O16')

Out[234]: 15.99491461957
```

We simplify our calculations by assuming that these are isotopically pure. Since the natural abundance of $^{16}\mathrm{O}$ and $^{1}\mathrm{H}$ far exceed other isotopes, this is a reasonable assumption

```
In [235]: data.natural_abund('O16')

Out[235]: 0.9975700000000001
```

```
In [236]: data.natural_abund('H1')

Out[236]: 0.999885
```

In [237]:
```
m_water = 2*data.atomic_mass('H1') + data.atomic_mass('O16')
```

In [238]:
```
N_water = const.Avogadro / m_water
```

In [239]:
```
N_hydrogen = 2 * N_water
N_oxygen = N_water
```

Now we load our cross sections from the ENDL data file

In [240]:
```
sigma_H = sigma_interpolate('H-1.txt', path)
sigma_O = sigma_interpolate('O-16.txt', path)
```

In [241]:
```
res = []
for E in [14*10**6, 10**6, 0.05]:
    Sigma = N_hydrogen * sigma_H(E) / 10**24 + N_oxygen * sigma_O(E) /
10**24
    lamb = 1 / Sigma
    res.append((E, lamb))
print tabulate(res, headers=headers)
```

```
  Energy    Mean free path (cm)
--------  ---------------------
 1.4e+07               10.0399
 1e+06                  1.7888
 0.05                   0.532354
```

Air

Air is composed of:

Molecule	Abundance fraction, f
N_2	78%
O_2	21%
Ar	1.0%

N_{\mathrm{Ar}} = \rho_{air} * \frac{A_v}{M_{air} * f

```
In [242]: rho_air = 0.001204
```

As before, we assume that air is composed of isotopically pure N-14, O-16, and Ar-40

```
In [243]: data.natural_abund('N14')

Out[243]: 0.99636
```

```
In [244]: data.natural_abund('O16')

Out[244]: 0.9975700000000001
```

```
In [245]: data.natural_abund('Ar40')

Out[245]: 0.996035
```

```
In [246]: M_air = 0.78 * 2 * data.atomic_mass('N14') + 0.21 * 2 * data.atomic_ma
          ss('O16') + 0.01 * data.atomic_mass('Ar40')
```

```
In [247]: N_Ar = rho_air * const.Avogadro / M_air * 0.01
          N_N = rho_air * const.Avogadro / M_air * 0.78 * 2
          N_O = rho_air * const.Avogadro / M_air * 0.21 * 2
```

```
In [248]: sigma_O = sigma_interpolate('O-16.txt', path)
          sigma_N = sigma_interpolate('N-14.txt', path)
          sigma_Ar = sigma_interpolate('Ar-40.txt', path)
```

```
In [249]: res = []
          for E in [14*10**6, 10**6, 0.05]:
              Sigma = (N_Ar * sigma_Ar(E) + N_O * sigma_O(E) + N_N * sigma_N(E))
          / 10**24
              lamb = 1 / Sigma
              res.append((E, lamb))
          print tabulate(res, headers=headers)

            Energy    Mean free path (cm)
          --------  ---------------------
           1.4e+07                12715.8
           1e+06                   5639.99
           0.05                    2041.32
```

Problem 2-12

In [269]:
```
const.h
```

Out[269]:
```
6.62607004e-34
```

In [270]:
```
const.physical_constants['neutron mass'][0]
```

Out[270]:
```
1.674927471e-27
```

In [280]:
```
def de_broglie_energy(wavelength):
    # Returned value is in MeV
    h = const.h
    n_mass = const.physical_constants['neutron mass'][0]
    E_joules = (h / wavelength)**2 / (2 * n_mass)
    E_MeV = E_joules * const.physical_constants['joule-electron volt r
elationship'][0] / 10**6
    return E_MeV
```

Our lengths of interest are stored in the following `dict` (in units of meters)

In [283]:
```
wavelengths = {
    'H-nucleus': 2.4*10**-15,
    'U-nucleus': 15*10**-15,
    'H-atom': 50*10**-12,
    'U-atom': 350*10**-12,
    'Graphite inter-atomic spacing': 0.142*10**-9,
    'Nuclear core diameter': 1
}
```

In [284]:
```
out_table = []
headers = ("Thing", "KE (MeV)")
for thing, wavelength in wavelengths.items():
    out_table.append((thing, de_broglie_energy(wavelength)))
print tabulate(out_table, headers=headers)
```

```
Thing                                 KE (MeV)
-----------------------------  -------------
Nuclear core diameter            8.18042e-28
H-atom                           3.27217e-07
Graphite inter-atomic spacing    4.05694e-08
H-nucleus                      142.021
U-nucleus                        3.63574
U-atom                           6.67789e-09
```

In [ ]:
```

```