1 Extended Tao-Eldrup model

We are given following equations from [2]:

$$P_{nm} = \frac{\int\limits_{Z_{nm}\frac{R}{R+\Delta}}^{Z_{nm}} J_m(r)^2 r dr}{\int\limits_0^{Z_{nm}} J_m(r)^2 r dr} \tag{1}$$

$$\lambda_{nm} = \lambda_p P_{nm} + \lambda_i (1 - P_{nm}) \tag{2}$$

$$\tau = \frac{\sum\limits_n \sum\limits_m g_m \exp\left(-\frac{E_{nm}}{kT}\right)}{\sum\limits_n \sum\limits_m g_m \lambda_{nm} \exp\left(-\frac{E_{nm}}{kT}\right)} \tag{3}$$

$$E_{nm} = \frac{\hbar^2}{4m_e} \frac{{Z_{nm}}^2}{(R+\Delta)^2}$$

where Z_{nm} is the n-th node of the Bessel function of the first kind $J_m(r)$ and g_m is the statistical weight of the m-th state ($g_0 = 1$, $g_{m>0} = 2$).

2 Bessel functions

2.1 $J_0(x)$ from go's library

2.1.1 For $|x| < 2$

They used some arbitrary constants that I didn't know where they came from. I used simple expansion from formula

$$J_n(x) = \sum_{k=0}^{\infty} \frac{(-1)^k}{k!\Gamma(n+k+1)} \left(\frac{x}{2}\right)^{n+2k} \tag{4}$$

and I calculated 8 first terms.

2.1.2 Other x's

They used Hanakel's asymptotic expansion

$$J_n(x) = \sqrt{\frac{2}{\pi x}} \left(P_n(x)\cos(x_0) - Q_n(x)\sin(x_0)\right) \tag{5}$$

where $x_0 = x - (\pi/2 + 1/4)\pi$. And i did the same.

2.2 My idea of how to calculate things

2.2.1 Approxmiating

Values of Bessel functions of the first kind are kindly provided by the Go's mathematics package. Because Bessel functions' derivatives can be defined by

$$\frac{d}{dx} J_n(x) = J_{n-1}(x) - \frac{n}{x} J_n(x)$$

or by

$$\frac{d}{dx}J_n(x) = \frac{1}{2}(J_{n-1}(x) - J_{n+1}(x))$$

we can use approximation techniques for finding zeroes that use derivatives. I'm thinking mainly about Newton's or Halley's methodes.

Recursive sequence in Newton's method is defined in our case by

$$x_{n+1} = x_n - \frac{J_v(x_n)}{J_v'(x_n)} = x_n - \frac{J_v(x_n)}{J_{v-1}(x) - \frac{v}{x_n}J_v(x)} = x_n - \frac{2J_v(x)}{J_{v-1}(x) - J_{v+1}(x)}$$

The last equation might be actually worse than I innitialy thought because we would need to calculate Bessel function 3 times compared to 2 times in the second to last equation.

Halley's method can be defined by

$$x_{n+1} = x_n - \frac{2J_v(x_n)J_v'(x_n)}{2(J_v'(x_n))^2 - J_v(x_n)J_v''(x_n)}$$

which looks not as easy by I will check how fast it converges compared to Newton's method.

2.2.2 Initial guess

This wasn't easy because you can't use finite equation for it because it decreses precision with bigger values.

For zeros of $J_0(x)$ I used approximation given by McMahon's asymptotic expansion [1, equation 10.21.19]

$$j_{v,m} \sim a - \frac{\mu - 1}{8a} - \frac{4(\mu-1)(7\mu-31)}{3(8a)^3} - \frac{32(\mu-1)(83\mu^2 - 982\mu + 3779)}{15(8a)^5} - \dots \quad (6)$$

where $j_{v,m}$ is the m-th zero of the Bessel function $J_v(x)$, $\mu = 4v^2$ and $a = (m + \frac{1}{2}v - \frac{1}{4})\pi$.

For zeros $j_{v,m}$ where $v > 0$ I just put it in bounds $j_{v-1,m-1} < j_{v,m} < j_{v-1,m+1}$ [1, equation 10.21.2] and approximated using Newton's formula until I got what I wanted. I check values for big v and m and got correct anserw so it is assumed that my code is working.

2.3 How it's done by GNU Scientific Library

2.3.1 Zeros of Bessel function

2.4 Numerical integration

Next mission is to calculate 1 by using numerical integration algorithms. Use of simple midpoint rule is inefficient so there's need to use Gauss' quadrature. I arbitrarily chose 10 points given in [1, table 3.5.2]

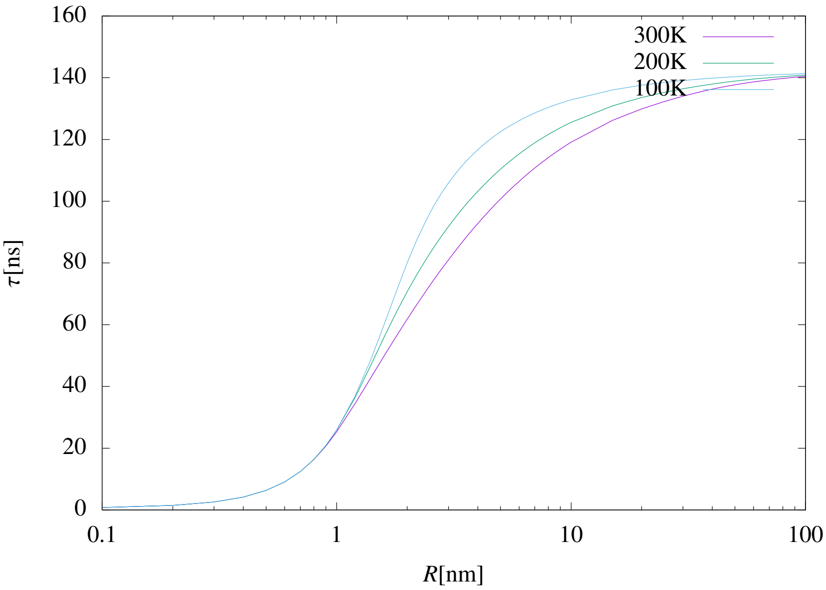

Figure 1: Temperature

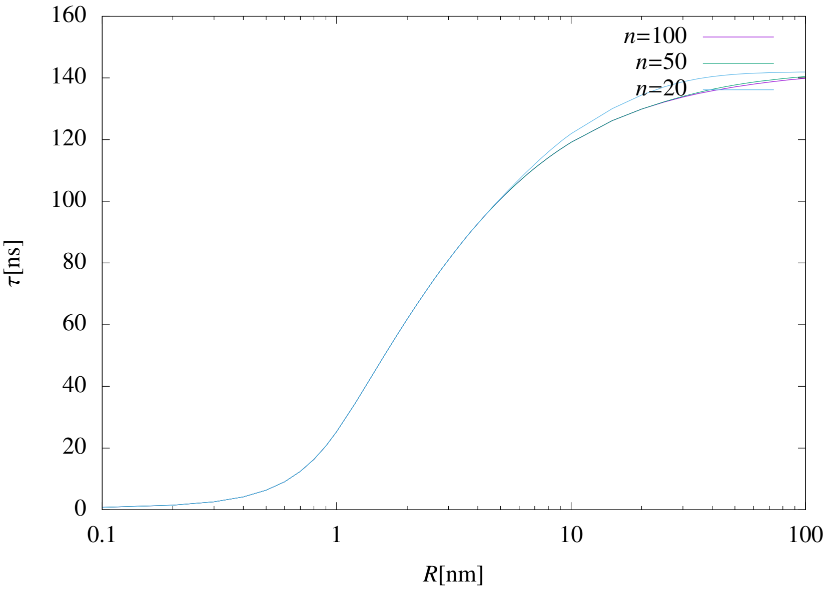

Figure 2: Number of $n = m$ to sum

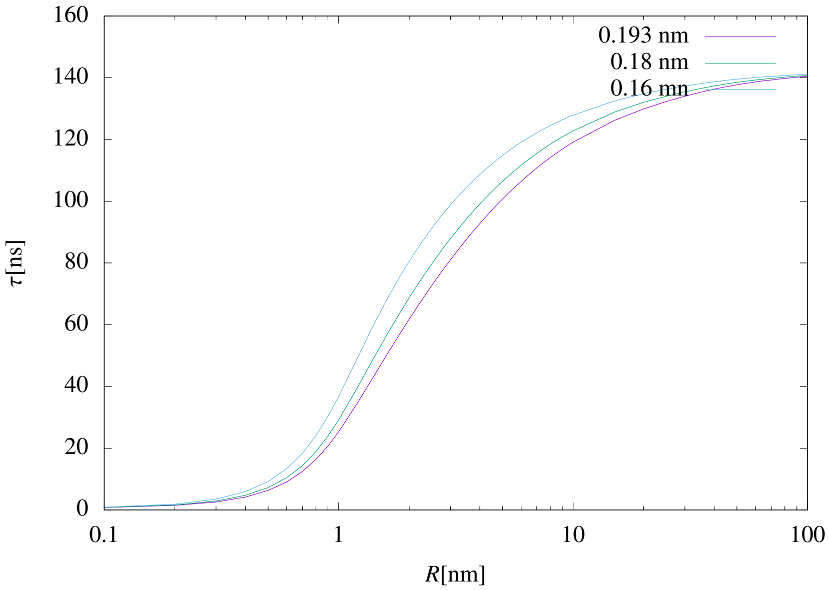

Figure 3: Δ parameter

3 A study of the extended Tao-Eldrup model's parameters

4 Calculating R from ETE

4.1 Initial guess

I went for approxmiating τ as

$$\tau \approx \frac{140}{1 + x/5^{-3/2}}, \tag{7}$$

which is a CDF of a log-logistic distribution.

It looks quite good but it doesn't matter so much as it will just get corrected. I will check later fo comparison against different approximation. I used wxMaxima to calculate inverse of this function which in my case will be

$$R \approx \left(\frac{39480499x}{4993932(140 - x)}\right)^{\frac{2}{3}} \tag{8}$$

References

[1] *NIST Digital Library of Mathematical Functions*. http://dlmf.nist.gov/, Release 1.1.3 of 2021-09-15. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller, B. V. Saunders, H. S. Cohl, and M. A. McClain, eds. URL: `http://dlmf.nist.gov/`.

[2] R. Zaleski. Principles of positron porosimetry. *Nukleonika*, 60(4):795–800, 2015. `doi:10.1515/nuka-2015-0143`.