March, April, May, June Report:
Adapting Particle Transport Monte Carlo Code to Solve the 1D Time Independent Schrödinger Equation

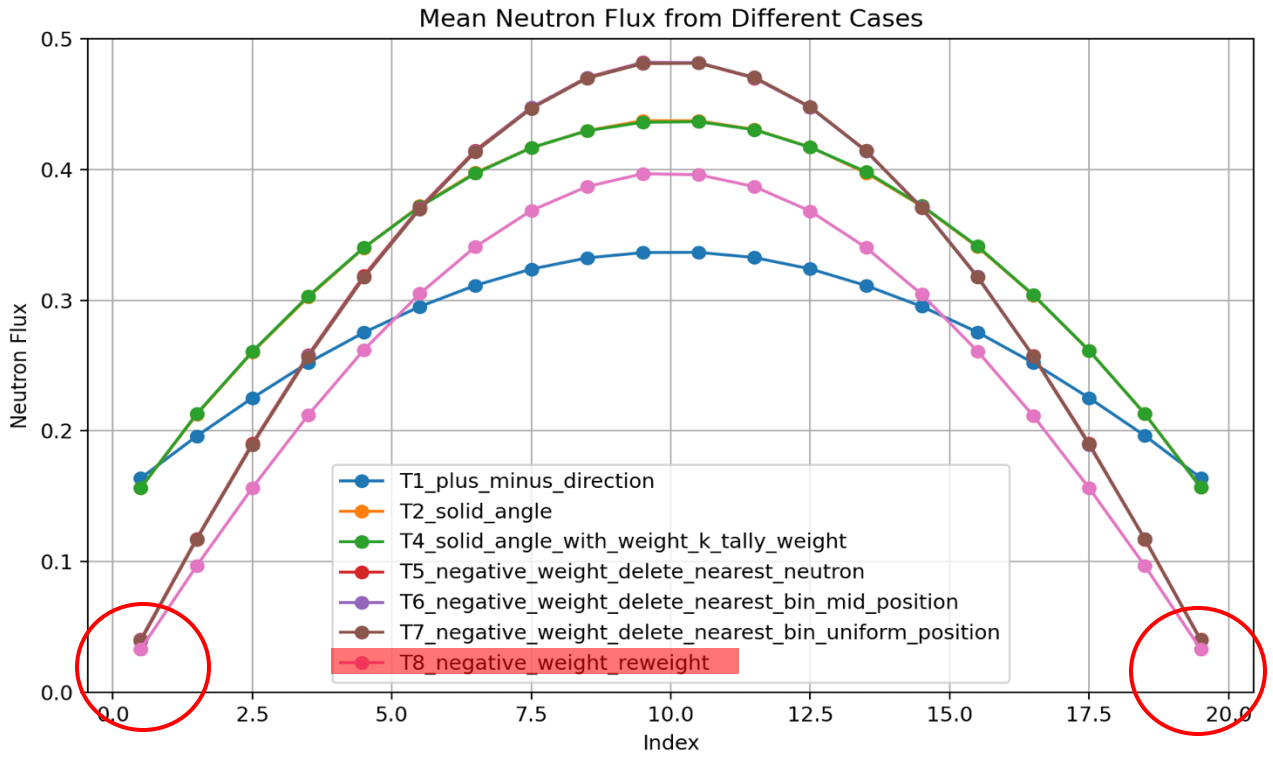
물리학과 2021313130 이유나

2025. 06. 24

Recap

- Vacuum boundary condition

- Input values
  - ➢Boundary condition: 0
  - ➢Number of neutrons: 100000
  - ➢Number of inactive cycles: 50
  - ➢Number of active cycles: 300
- Variables
  - ➢width = 20 cm
  - ➢sigma_s = 0.1 cm^{-1}
  - ➢sigma_c = 0.07 cm^{-1}
  - ➢sigma_f = 0.06 cm^{-1}
  - ➢num_bins = 20



- The code from Case T8 produces neutron flux values closest to zero at the boundaries. It is the most effective boundary correction method among all those tested.

Recap

- Vacuum boundary condition

```
Boundary condition (0 = vacuum, 1 = reflective): 0
Number of neutrons: 100000
Total number of cycles: 300
Number of inactive cycles: 50
Number of active cycles: 250
k_a=0.906384, k_sa=0.821535, std=0.000112
k1=0.879588, k2=0.533992, k2/k1=0.607093
```

- $k1 = 0.879588$
- $k_{average} = 0.906384 \pm 0.000252$ (68.27% CI)
- $k1$ is not within the range of $k_{average}$

Recap

- Vacuum boundary condition

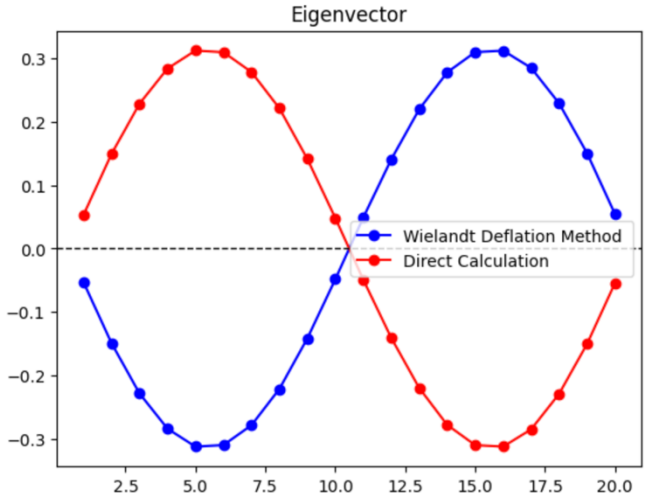
➢ Fission matrix

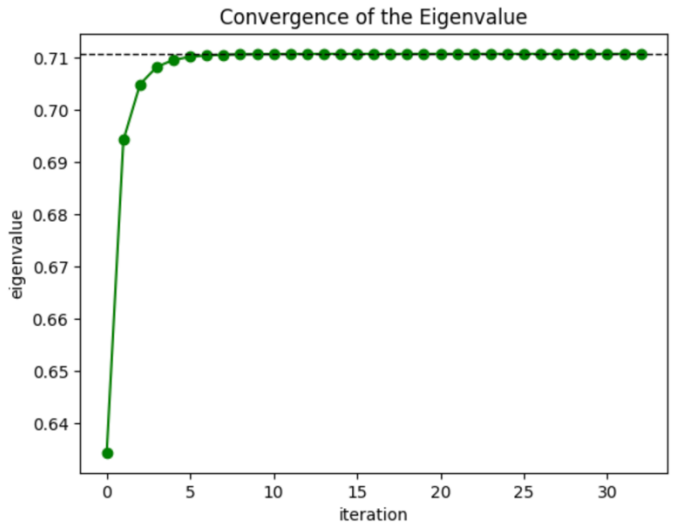
```
0.17990 0.07071 0.02856 0.01598 0.00957 0.00623 0.00422 0.00280 0.00194 0.00128 0.00100 0.00067 0.00047 0.00031 0.00021 0.00013 0.00004 −0.00013 −0.00016 −0.00037
0.09112 0.25691 0.11516 0.05430 0.03172 0.01981 0.01340 0.00890 0.00633 0.00443 0.00309 0.00224 0.00161 0.00115 0.00082 0.00055 0.00037 0.00014 −0.00005 −0.00039
0.03947 0.12302 0.27700 0.13046 0.06458 0.03869 0.02493 0.01653 0.01156 0.00801 0.00569 0.00409 0.00294 0.00211 0.00146 0.00103 0.00067 0.00040 0.00014 −0.00018
0.02026 0.05734 0.13604 0.28546 0.13679 0.06916 0.04203 0.02742 0.01832 0.01268 0.00894 0.00648 0.00447 0.00328 0.00239 0.00170 0.00114 0.00072 0.00026 0.00024
0.01202 0.03385 0.06639 0.14027 0.28940 0.14083 0.07195 0.04385 0.02872 0.01934 0.01355 0.00944 0.00685 0.00486 0.00336 0.00239 0.00166 0.00107 0.00061 0.00009
0.00776 0.02090 0.04038 0.07096 0.14311 0.28984 0.14227 0.07302 0.04479 0.02926 0.02007 0.01383 0.00959 0.00692 0.00491 0.00349 0.00232 0.00154 0.00084 0.00021
0.00497 0.01397 0.02611 0.04267 0.07302 0.14407 0.29165 0.14290 0.07377 0.04531 0.02953 0.02009 0.01393 0.00973 0.00686 0.00487 0.00331 0.00217 0.00132 0.00019
0.00422 0.00972 0.01717 0.02836 0.04472 0.07420 0.14454 0.29202 0.14408 0.07415 0.04564 0.02983 0.02014 0.01398 0.00961 0.00665 0.00476 0.00305 0.00153 0.00078
0.00275 0.00664 0.01162 0.01867 0.02904 0.04520 0.07452 0.14498 0.29213 0.14467 0.07409 0.04567 0.02971 0.02028 0.01379 0.00957 0.00660 0.00416 0.00254 0.00065
0.00212 0.00458 0.00832 0.01287 0.01947 0.02935 0.04553 0.07441 0.14496 0.29185 0.14465 0.07465 0.04588 0.02978 0.02001 0.01362 0.00913 0.00579 0.00336 0.00125
0.00092 0.00340 0.00579 0.00914 0.01363 0.02008 0.02941 0.04550 0.07436 0.14455 0.29245 0.14482 0.07435 0.04592 0.02961 0.01938 0.01302 0.00845 0.00496 0.00157
0.00100 0.00244 0.00423 0.00638 0.00962 0.01390 0.02029 0.02988 0.04575 0.07464 0.14421 0.29191 0.14489 0.07453 0.04492 0.02873 0.01892 0.01185 0.00663 0.00278
0.00036 0.00192 0.00295 0.00458 0.00680 0.00984 0.01406 0.02033 0.02976 0.04565 0.07389 0.14380 0.29174 0.14463 0.07390 0.04438 0.02763 0.01733 0.00992 0.00353
0.00034 0.00121 0.00214 0.00329 0.00483 0.00684 0.00991 0.01393 0.02006 0.02944 0.04537 0.07379 0.14299 0.29118 0.14416 0.07268 0.04311 0.02558 0.01412 0.00530
0.00036 0.00091 0.00156 0.00242 0.00344 0.00495 0.00694 0.00970 0.01393 0.02003 0.02937 0.04477 0.07316 0.14226 0.29078 0.14302 0.07105 0.03961 0.02091 0.00798
0.00006 0.00063 0.00113 0.00164 0.00246 0.00341 0.00483 0.00669 0.00947 0.01341 0.01950 0.02871 0.04392 0.07197 0.14027 0.28870 0.14074 0.06685 0.03322 0.01242
−0.00007 0.00038 0.00079 0.00113 0.00166 0.00240 0.00330 0.00458 0.00646 0.00895 0.01276 0.01848 0.02737 0.04229 0.06959 0.13719 0.28512 0.13547 0.05801 0.02310
−0.00017 0.00016 0.00044 0.00072 0.00106 0.00153 0.00211 0.00296 0.00393 0.00565 0.00807 0.01152 0.01680 0.02512 0.03849 0.06449 0.13002 0.27757 0.12282 0.03872
−0.00021 −0.00005 0.00015 0.00031 0.00055 0.00079 0.00114 0.00162 0.00227 0.00319 0.00444 0.00619 0.00918 0.01312 0.02013 0.03162 0.05414 0.11546 0.25917 0.08838
−0.00013 −0.00017 −0.00015 −0.00001 0.00012 0.00012 0.00036 0.00042 0.00072 0.00101 0.00128 0.00210 0.00282 0.00430 0.00616 0.00964 0.01612 0.02879 0.07029 0.18357
```

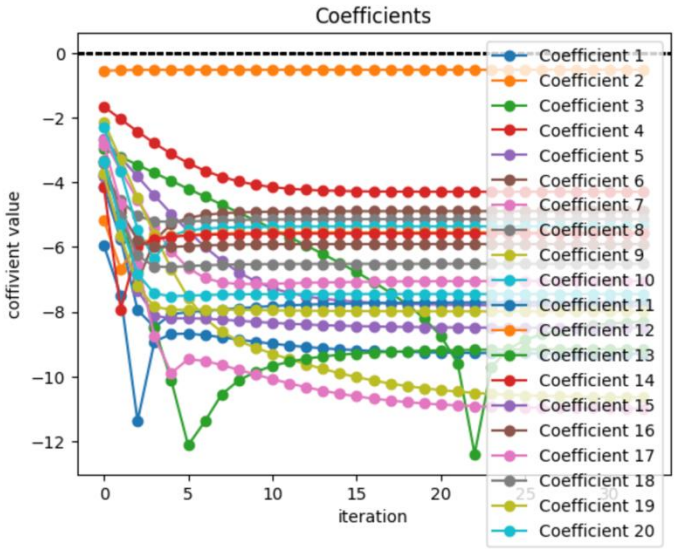
- The matrix is not exactly symmetric, but exhibits a strong symmetric pattern. This enables deflation methods to be effectively applied for computing higher eigenmodes.

Recap

- The Wielandt deflation method was used to compute the higher eigenvalues and their corresponding eigenvectors.
- The following figures show the second-largest eigenvalue and its corresponding eigenvector.

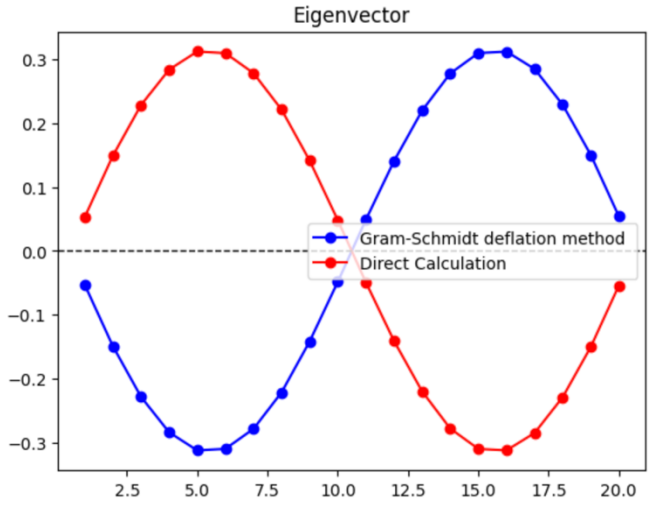


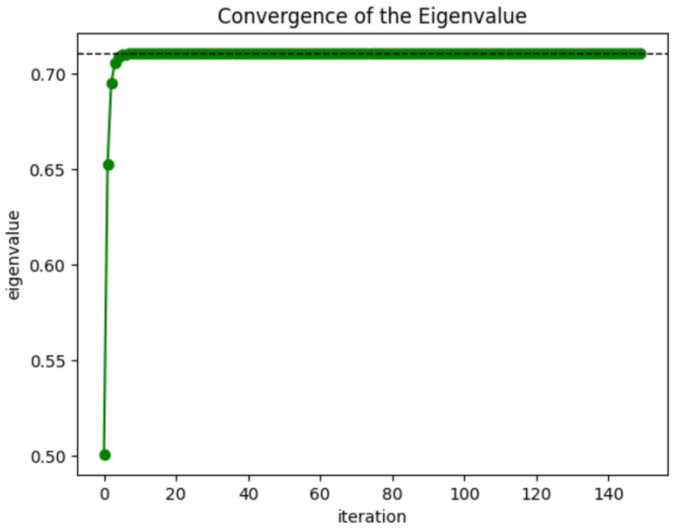


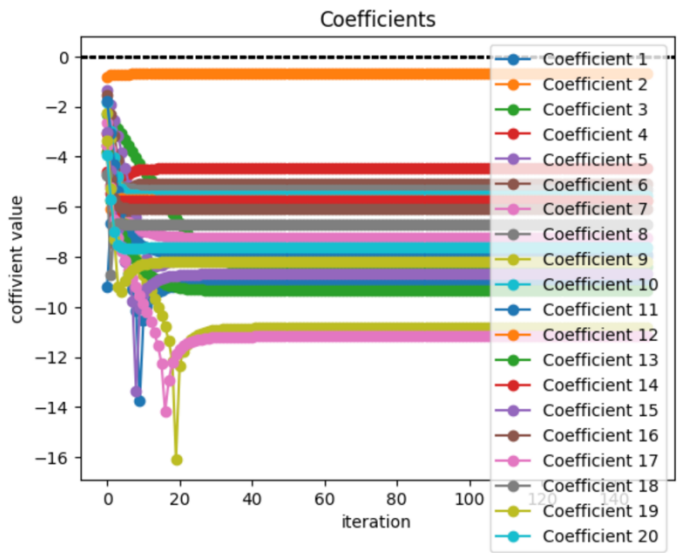


Recap

- The Gram-Schmidt deflation method was used to compute the higher eigenvalus and their corresponding eigenvectors.
- The following figures show the second-largest eigenvalue and its corresponding eigenvector.



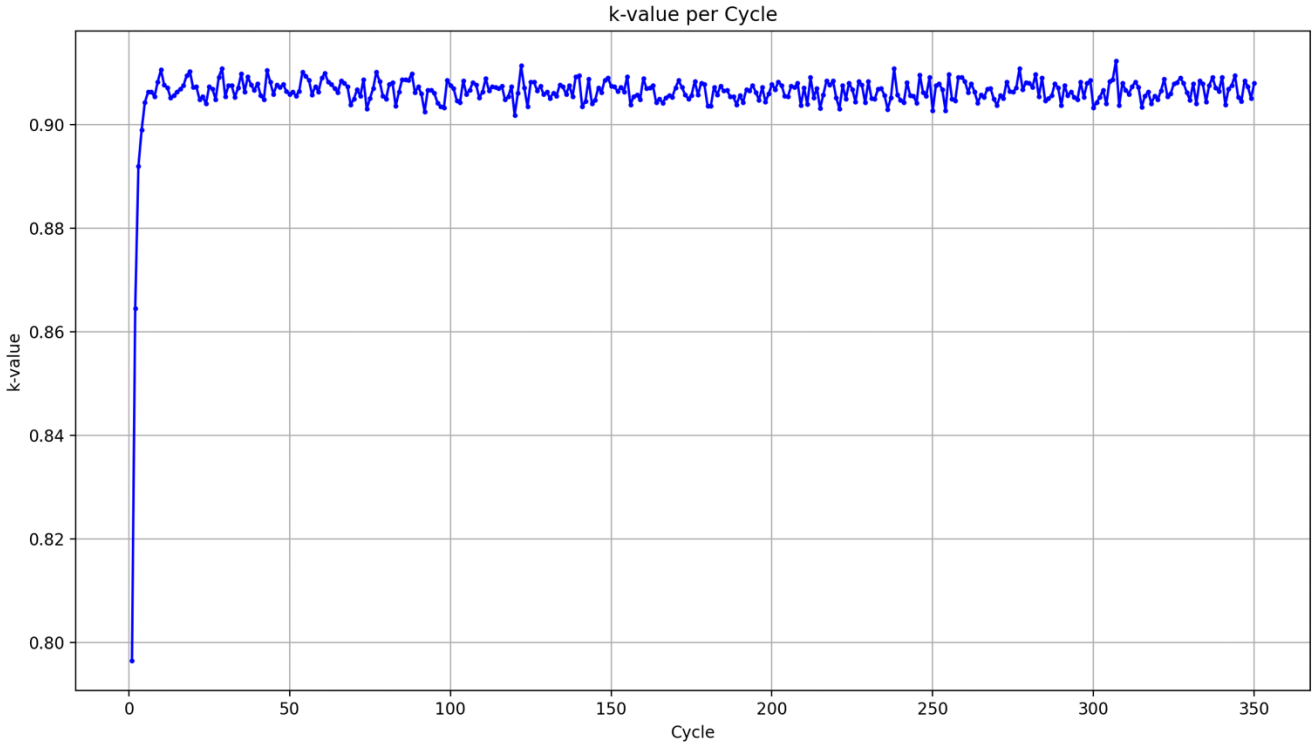




Why Use Averaging for k-value Calculation?

Q. Why is the k-value calculated as the average over active cycles rather than using the final k-value?

- Input values
  - Boundary condition: 0
  - Number of neutrons: 100000
  - Number of inactive cycles: 50
  - Number of active cycles: 300
- Variables
  - width = 20 cm
  - sigma_s = 0.1 cm^{-1}
  - sigma_c = 0.07 cm^{-1}
  - sigma_f = 0.06 cm^{-1}
  - num_bins = 20



Why Use Averaging for k-value Calculation?

A.

- Inactive cycles are used to allow the source distribution to reach steady state.
- Active cycles are used to collect k-values once the source has reached steady-state.
- Although k-values from active cycles may appear to converge, they still fluctuate around the true mean due to statistical uncertainty.
- Each active cycle generates a single k-value; averaging these (tallying) yields a more reliable estimate according to the Central Limit Theorem.
- Simply taking the last k-value ignores these fluctuations and provides no measure of uncertainty.

Current Progress

1) ~~Study Monte Carlo particle transport method~~ ✓ **Done**
2) ~~Implement code for fundamental mode~~ ✓ **Done**
3) ~~Implement code for higher modes~~ ✓ **Done**
4) Interpret as a Schrödinger equation solution ✓ **In progress**
5) Apply to more complex situations
   - Various potentials
6) Study numerical and QMC methods
7) Compare performance between methods

Methods Used in Reference Studies

- Main reference paper

J. Shentu, S.-H. Yun, and N.-Z. Cho, **"A Monte Carlo method for solving heat conduction problems with complicated geometry"** *Nuclear Engineering and Technology*, vol. 39, no. 3. Elsevier BV, pp. 207–214, 30-Jun-2007.

- Additional reference paper

S. K. Fraley, T. J. Hoffman, and P. N. Stevens, ***"A Monte Carlo Method of Solving Heat Conduction Problems"*,** ORNL/NUREG/CSD-3, Oak Ridge National Laboratory, prepared for the U.S. Nuclear Regulatory Commission, 1977.

Methods Used in Reference Studies

- Steady state differential equation of heat conduction for a stationary, isotropic solid:

$$\nabla \cdot K(\vec{r})\nabla T(\vec{r}) + q'''(\vec{r}) = 0$$

$K(\vec{r})$: thermal conductivity, $q'''(\vec{r})$: internal heat source

- Steady state neutron diffusion equation under isotropic scattering, no absorption, and no fission condition:

$$\nabla \cdot \frac{1}{3\Sigma_s}\nabla\phi(\vec{r}) + S(\vec{r}) = 0$$

$\phi(\vec{r})$: neutron flux, Σ_s: scattering cross section, $S(\vec{r})$: internal neutron source

Methods Used in Reference Studies

- The previous two equations are formally identical.
- Hence, solving the heat conduction problem with $K(\vec{r})$ and $q'''(\vec{r})$ is equvialent to solving the neutron diffusion problem with $\Sigma_s = \frac{1}{3K(\vec{r})}$ and $S(\vec{r}) = q'''(\vec{r})$.

Methods Used in Reference Studies

- The neutron diffusion equation is an approximation to the neutron transport equation, derived using Fick's Law.
- Fick's Law is invalid within three mean free paths of either a neutron source or the surface of a material.

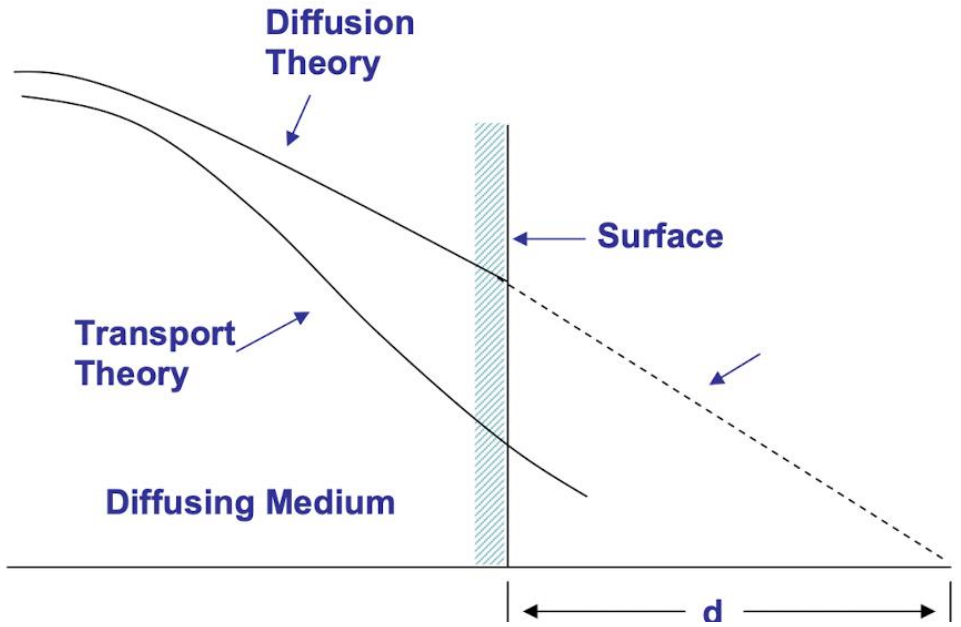


Figure from MIT Neutron Science and Reactor Physics 2006 by John A. Bernard

Methods Used in Reference Studies

- Since neutron diffusion theory approximates neutron transport accurately except near boundaries, conversely it is applicable to simulate diffusion using transport theory in the interior region.
- The reason for this is that MCNP simulates only neutron transport and not diffusion, so to use MCNP, neutron transport must be simulated.
- Thus, a suitable correction is needed for the boundary region.

Methods Used in Reference Studies

- Two approaches:

1. Enlarging the valid range of the transport approximation
   - If the invalid range (several mean free paths near the boundary) is reduced, more reliable data can be obtained near the boundary.
2. Applying a boundary correction
   - If a correction is applied near the boundary, accurate results can be obtained in that region.

Methods Used in Reference Studies

- Two approaches

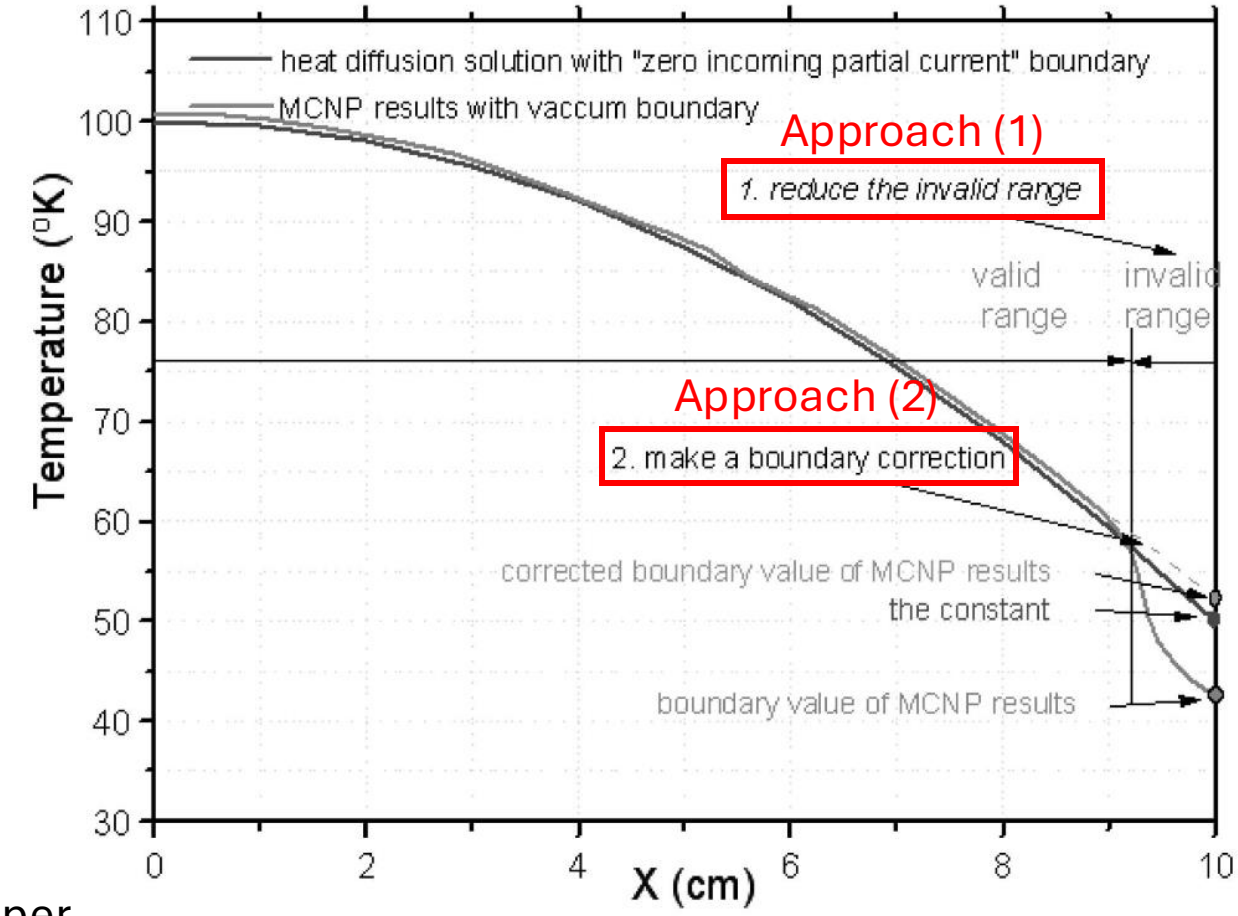


Figure from main reference paper

Methods Used in Reference Studies

- First approach: Enlarging the valid range (= reducing the mean free path) of the transport approximation
- Rewrite the heat conduction equation using a scaling factor β:

$$\nabla \cdot \frac{1}{\beta} K(\vec{r}) \nabla T(\vec{r}) + \frac{1}{\beta} q'''(\vec{r}) = 0$$

- Relationship between the mean free path and β:

$$\lambda = \frac{1}{\Sigma_s} = \frac{3K(\vec{r})}{\beta}$$

- Therefore, a larger β results in a shorter mean free path, or equivalently, a larger valid range.

Methods Used in Reference Studies

- However, reducing the mean free path increases the computational cost.
- Therefore, optimal scaling involves a trade-off between approximation accuracy and computational efficiency.

Methods Used in Reference Studies

- Second approach: Applying a linear extrapolation to the boundary.
  - The shorter the distance the boundary correction must cover, the better.
  - A small mean free path implies a short invalid range of the transport approximation.
  - Over short distances, the true temperature distribution tends to exhibit linearity, regardless of its overall shape.
  - Thus, a large β gives a smaller invalid range and a more accurate boundary correction.

Methods Used in Reference Studies

- To determine the appropriate scaling factor and boundary correction distance, the following two problems were tested:
- Problem 1. A one-dimensional homogeneous slab with $L = 10cm$, $K = \frac{0.5\mathrm{W}}{\mathrm{cm}\cdot{}^{\circ}K}$, $q''' = \frac{10W}{cm^3}$ throughout the slab. One side is reflective, and the other size is maintained at zero temperature.

Methods Used in Reference Studies

- Problem 2. A one-dimensional heterogeneous slab with $L = 10cm$, $q''' = \frac{10W}{cm^3}$ throughout the slab. One side is reflective, and the other size is maintained at zero temperature.
- The heat conductivities are given as follows:

$$K = \begin{cases} \frac{0.2\mathrm{W}}{\mathrm{cm} \cdot {}^\circ K}, & 0cm \le x \le 2cm, \\ \frac{0.3\mathrm{W}}{\mathrm{cm} \cdot {}^\circ K}, & 2cm \le x \le 4cm, \\ \frac{0.4\mathrm{W}}{\mathrm{cm} \cdot {}^\circ K}, & 4cm \le x \le 6cm, \\ \frac{0.5\mathrm{W}}{\mathrm{cm} \cdot {}^\circ K}, & 6cm \le x \le 10cm \end{cases}$$

Methods Used in Reference Studies

- The Root Mean Squared Error (RMSE) between the analytical solution and the transport solution (scaled, boundary corrected and translated) was computed for a range of scaling factors (5, 7, 10, 13, 16, 20, 25, 30) and boundary correction distance in mean free path units (0.0, 0.25, 0.5, 0.75, 1.0, 1.25, 1.5, 1.75, 2.0).
- The specific values and plots are not shown in this presentation. However, based on the RMSE results reported in the paper, it is concluded that the boundary correction should begin at a distance of 1.0 mean free path, and the scaling factor should be set between 10 and 20.

Methods Used in Reference Studies

- As a summary, the procedure to solve the diffusion problem using transport theory is as follows:

Step 1. Choose a scaling factor and scale the diffusion problem:

$$\Sigma_i = \frac{\beta}{3\Sigma_i}, S = \frac{q'''}{\beta}$$

Step 2. Solve the scaled problem using MCNP.

Step 3. Apply a linear correction near the boundary.

Step 4. Translate the corrected results to satisfy the boundary condition.

Methods Used in Reference Studies

- Illustration of the scaling and correction procedure.

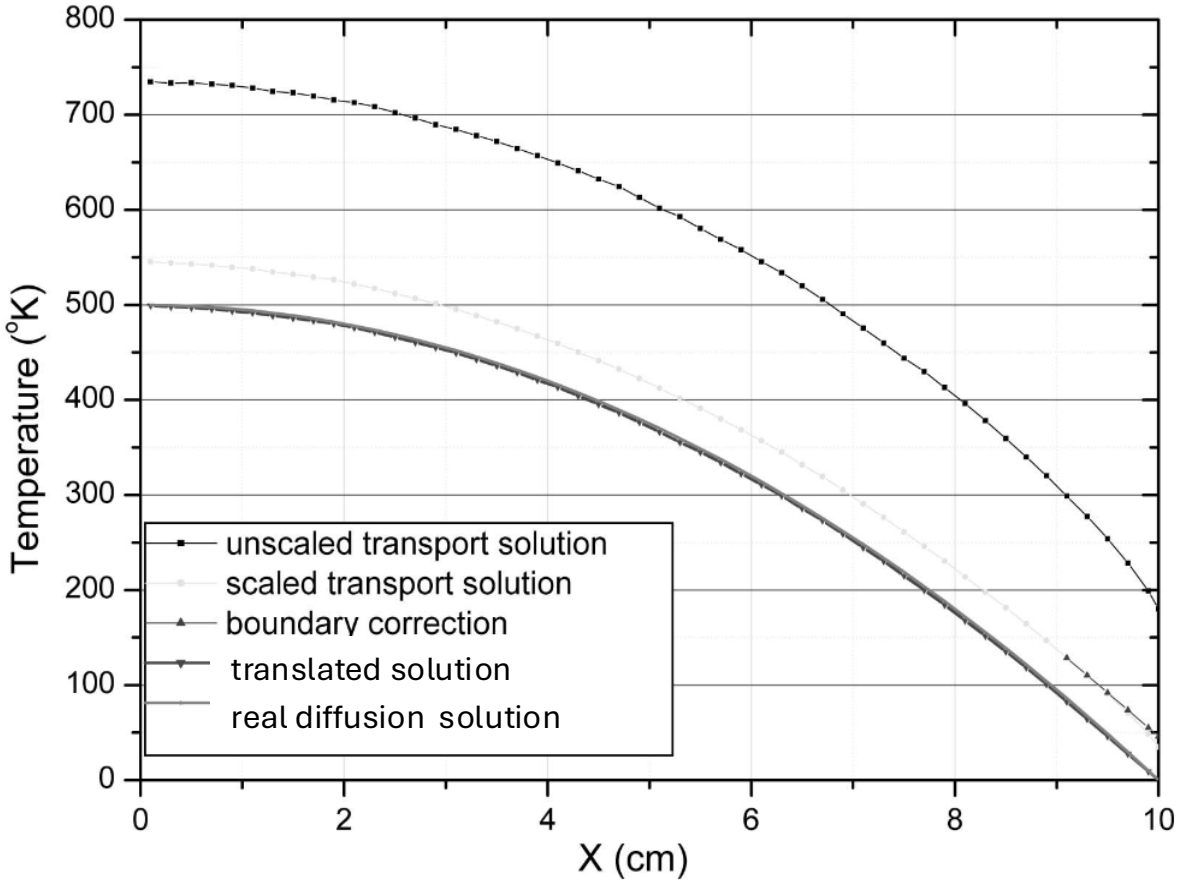


Figure from main reference paper

Methods Used in Reference Studies

- Takeaways from the reference studies:
  - The neutron diffusion and heat conduction equations were directly compared.
  - The optimal scaling factor depends on the problem and must be manually determined.

Methods Used in This Study

- Two approaches were used to find the scaling factor that appropriately scales the Schrödinger equation to the neutron diffusion equation:

1. **Dimensionless Approach**
   - Transform both equations into a dimensionless form and compare their coefficients.
2. **Direct Scaling Apporach**
   - Compare the coefficients directly and determine a scaling factor that aligns the equations.

Methods Used in This Study

1. Dimensionless Approach

- Dimensionless Schrödinger Equation

$$\tilde{x} \equiv \frac{x}{L},\ [L] = [x] \rightarrow \frac{d}{dx} = \frac{d\tilde{x}}{dx}\frac{d}{d\tilde{x}} = \frac{1}{L}\frac{d}{d\tilde{x}},\ \frac{d^2}{dx^2} = \frac{d}{dx}\left(\frac{1}{L}\frac{d}{d\tilde{x}}\right) = \frac{1}{L^2}\frac{d^2}{d\tilde{x}^2}$$

$$\tilde{H} \equiv \frac{mL^2}{\hbar^2}H,\ \left[\frac{mL^2}{\hbar^2}\right] = \frac{M\cdot L^2}{L^2\cdot M^2\cdot L^2/S^2} = \frac{S^2}{ML^2},\ [H] = \frac{M\cdot L^2}{S^2}$$

$$\tilde{H} = \frac{mL^2}{\hbar^2}H = \frac{mL^2}{\hbar^2}\left(-\frac{\hbar^2}{2m}\frac{1}{L^2}\frac{d^2}{d\tilde{x}^2} + V(L\tilde{x})\right)$$

$$= -\frac{d^2}{2d\tilde{x}^2} + \frac{mL^2}{\hbar^2}V(L\tilde{x})$$

$$\tilde{E} \equiv \frac{mL^2}{\hbar^2}E$$

$$\tilde{\psi}(\tilde{x}) \equiv \sqrt{L}\,\psi(L\tilde{x}),\ [\sqrt{L}] = [\psi]$$

$$H\psi(x) = E\psi(x) \rightarrow \frac{mL^2}{\hbar^2}H\cdot\sqrt{L}\psi = \frac{mL^2}{\hbar^2}E\cdot\sqrt{L}\psi$$

$$\therefore\ \boxed{-\frac{1}{2}\frac{d^2}{d\tilde{x}^2}\tilde{\psi}(\tilde{x}) + \frac{mL^2}{\hbar^2}V(L\tilde{x})\tilde{\psi}(\tilde{x}) = \tilde{E}\tilde{\psi}(\tilde{x})}$$

Methods Used in This Study

1. Dimensionless Approach

- Dimensionless Neutron Diffusion Equation

$$-D\frac{d^2}{dx^2}\phi(x)+\Sigma_a\phi(x)=\frac{1}{k}\nu\Sigma_f\phi(x)$$

Let $M=-D\frac{d^2}{dx^2}+\Sigma_a$, $A=\frac{1}{k}\nu\Sigma_f$

$$\left(-D\frac{d^2}{dx^2}+\Sigma_a\right)\phi(x)=\frac{1}{k}\nu\Sigma_f\phi(x)=A\phi(x)\longrightarrow M\phi(x)=A\phi(x)$$

$$[M]=\frac{1}{L},\ [\phi]=\frac{1}{L^2S},\ [\Sigma]=\frac{1}{L},\ [A]=\frac{1}{L}$$

$$\tilde{M}\equiv LM,\ \tilde{x}\equiv\frac{x}{L},\ \tilde{\phi}(\tilde{x})=L^2T\phi(L\tilde{x}),\ \tilde{A}=LA$$

$$\tilde{M}=L\cdot\left(-D\frac{1}{L^2}\frac{d^2}{d\tilde{x}^2}+\Sigma_a\right)=-\frac{D}{L}\frac{d^2}{d\tilde{x}^2}+L\Sigma_a$$

$$LM\cdot L^2T\phi(L\tilde{x})=LA\cdot L^2T\phi(L\tilde{x})\longrightarrow\tilde{M}\tilde{\phi}(\tilde{x})=\tilde{A}\tilde{\phi}(\tilde{x})$$

$$\therefore\ \boxed{-\frac{D}{L}\frac{d^2}{d\tilde{x}^2}\tilde{\phi}(\tilde{x})+L\Sigma_a\tilde{\phi}(\tilde{x})=\tilde{A}\tilde{\phi}(\tilde{x})}$$

Methods Used in This Study

1. Dimensionless Approach

- Comparing Coefficients (when $V \neq 0$)

$$\left(-\frac{D}{L}\frac{d^2}{d\tilde{x}^2}\tilde{\phi}(\tilde{x}) + L\Sigma_a\tilde{\phi}(\tilde{x}) = \tilde{A}\tilde{\phi}(\tilde{x})\right) \times \lambda$$

$$\rightarrow -\lambda\frac{D}{L}\frac{d^2}{d\tilde{x}^2}\tilde{\phi}(\tilde{x}) + \lambda L\Sigma_a\tilde{\phi}(\tilde{x}) = \lambda\tilde{A}\tilde{\phi}(\tilde{x})$$

$$-\frac{1}{2}\frac{d^2}{d\tilde{x}^2}\tilde{\psi}(\tilde{x}) + \frac{mL^2}{\hbar^2}V(L\tilde{x})\tilde{\psi}(\tilde{x}) = \tilde{E}\tilde{\psi}(\tilde{x})$$

$$\frac{\lambda D}{L} = \frac{1}{2} \rightarrow \lambda\cdot\frac{1}{3\Sigma_t}\cdot\frac{1}{L} = \frac{1}{2} \quad \therefore \Sigma_t = \frac{2\lambda}{3L} = \Sigma_a + \Sigma_s$$

$$\lambda L\Sigma_a = \frac{mL^2}{\hbar^2}V(L\tilde{x}) \quad \therefore \Sigma_a = \frac{mL}{\lambda\hbar^2}V(L\tilde{x}) = \Sigma_f + \Sigma_c$$

$$\rightarrow \Sigma_s = \Sigma_t - \Sigma_a$$

$$\lambda\tilde{A} = \tilde{E} \rightarrow \lambda L\frac{1}{k}\nu\Sigma_f = \frac{mL^2}{\hbar^2}E \quad \therefore E = \lambda\cdot\frac{\hbar^2\nu\Sigma_f}{mLk}$$

$$\tilde{\phi}(\tilde{x}) = \tilde{\psi}(\tilde{x}) \rightarrow L^2T\phi(x) = \sqrt{L}\psi(x) \quad \therefore \phi(x) = \frac{1}{L^{3/2}T}\psi(x)$$

$$\text{Let } T = \frac{mL}{\hbar} \rightarrow \phi(x) = \frac{\hbar}{L^{3/2}\cdot mL}\psi(x) = \frac{\hbar}{L^{5/2}m}\psi(x)$$

Methods Used in This Study

1. Dimensionless Approach

- Comparing Coefficients (when $V = 0$)

$$-\lambda \frac{D}{L}\frac{d^2}{d\tilde{x}^2}\tilde{\phi}(\tilde{x}) = (\lambda\tilde{A} - \lambda L\Sigma_a)\tilde{\phi}(\tilde{x})$$

$$-\frac{1}{2}\frac{d^2}{d\tilde{x}^2}\tilde{\psi}(\tilde{x}) = \tilde{E}\tilde{\psi}(\tilde{x})$$

$$\frac{\lambda D}{L} = \frac{1}{2} \rightarrow \lambda \cdot \frac{1}{3\Sigma_t}\cdot\frac{1}{L} = \frac{1}{2} \quad \therefore \Sigma_t = \frac{2\lambda}{3L} = \Sigma_a + \Sigma_s$$

$$\lambda\tilde{A} - \lambda L\Sigma_a = \tilde{E} \rightarrow \lambda L\frac{1}{k}\nu\Sigma_f - \lambda L\Sigma_a = \frac{mL^2}{\hbar^2}E \quad \therefore E = \lambda\cdot\frac{\hbar^2}{mL}\left(\frac{1}{k}\nu\Sigma_f - \Sigma_a\right)$$

$$\tilde{\phi}(\tilde{x}) = \tilde{\psi}(\tilde{x}) \rightarrow L^2T\phi(x) = \sqrt{L}\psi(x) \quad \therefore \phi(x) = \frac{1}{L^{3/2}T}\psi(x)$$

$$\text{Let } T = \frac{mL}{\hbar} \rightarrow \phi(x) = \frac{\hbar}{L^{3/2}\cdot mL}\psi(x) = \frac{\hbar}{L^{5/2}m}\psi(x)$$

Methods Used in This Study

1. Dimensionless Approach

- Why different methods are needed depending on the potential V:
  - In neutron diffusion simulations, the absorption cross section Σ_a must be nonzero because neutron fission is required to compute the k-eigenvalue and neutron flux.
  - Since fission cross section Σ_f is always nonzero, and $\Sigma_a = \Sigma_f + \Sigma_c$, the absorption cross section Σ_a is also nonzero.
  - However, in the Schrödinger equation, we may consider cases where the potential $V = 0$. In such cases, comparing terms involving Σ_a and V directly becomes invalid.
  - To resolve this, we move the Σ_a term to the other side of the equation to enable a meaningful comparison.

Methods Used in This Study

1. Dimensionless Approach

- Determining the scaling factor λ:
  - Even though both equations are made dimensionless, we still need a scaling factor to get appropriate values for simulation.
  - The value of λ depends on the specific physical problem (potential, particle mass).
  - Currently, λ must be manually determined for each case.

Algorithm

1. Read input values

input.txt

```
1   Boundary condition (zero = vacuum, one = reflective): 0
2   Number of neutrons: 100000
3   Total number of inactive cycles: 50
4   Total number of active cycles: 250
5   Particle name: electron
6   Particle mass (in scientific notation and SI unit): 9.10938e-31
7   Potential (in scientific notation and SI unit): 1.602e-18
```

2. Convert all units to CGS (cm, g, s)

Algorithm

3. Compute $\Sigma_s, \Sigma_f, \Sigma_c$ using the scaling factor λ:

$\Sigma_t = \frac{2}{3 * width} * \lambda$

If $(V == 0)$

do $\{r = random\_number\}$ while $(r == 0)$

$\Sigma_a = r * \Sigma_t$

Else

$\Sigma_a = \frac{m * width * V}{\hbar^2 * \lambda}$

$\Sigma_s = \Sigma_t - \Sigma_a$

do $\{r = random\_number\}$ while $(r == 0)$

$\Sigma_c = r * \Sigma_a$ // randomly chosen within range of Σ_a

$\Sigma_f = \Sigma_a - \Sigma_c$ // determined by subtraction from Σ_a

Algorithm

3. (continued) Adjust $\Sigma_s, \Sigma_f, \Sigma_c$ until Σ_f exceeds a threshold (0.02):

while $(\Sigma_f < 0.02)$ // limit to avoid excessive computation time

do $\{r = random\_number\}$ while $(r == 0)$

if $(V == 0)$

$\Sigma_a = r * \Sigma_t$

else

exit(EXIT_FAILURE) // terminate program

do $\{r = random\_number\}$ while $(r == 0)$

$\Sigma_s = \Sigma_t - \Sigma_a$

$\Sigma_c = r * \Sigma_a$

$\Sigma_f = \Sigma_a - \Sigma_c$

Algorithm

4. The rest of the algorithm follows the same logic as in the T8 implementation.

5. For the ground state energy, the results shown in the following slides used the final k value, plugging it into the equations below. However, the algorithm has now been updated to compute and tally the energy for each cycle using its corresponding k value and then calculate the average and standard deviation.

If (V==0) $E = \lambda \cdot \frac{\pi^2}{mL}\left(\frac{1}{k}\nu\Sigma_f - \Sigma_a\right)$

Else $E = \lambda \cdot \frac{\pi^2 \nu\Sigma_f}{mLk}$

Methods Used in This Study

2. Direct Scaling Approach

- Although similar to the dimensionless approach, this method is less intuitive. Comparing equations with different dimension can be awkward.
- The value of λ depends on the specific physical setup (potential, particle mass) and must be determined manually for each case.

$$-\frac{\hbar^2}{2m}\frac{d^2}{dx^2}\psi(x) + V(x)\psi(x) = E\psi(x)$$

$$\left(-D\frac{d^2}{dx^2}\phi(x) + \Sigma_a\phi(x) = \frac{1}{k}\nu\Sigma_f\phi(x)\right) \times \lambda$$

$$\frac{\hbar^2}{2m} = D\lambda = \frac{\lambda}{3\Sigma_t} \quad \therefore \Sigma_t = \frac{2m\lambda}{3\hbar^2}$$

$$\text{if } V \neq 0 \rightarrow \Sigma_a = \frac{1}{\lambda}V(x), \; E = \frac{\nu\Sigma_f\lambda}{k}$$

$$\text{else} \rightarrow E = \lambda\left(\frac{\nu\Sigma_f}{k} - \Sigma_a\right)$$

$$\psi(x) = \phi(x)$$

Methods Used in This Study

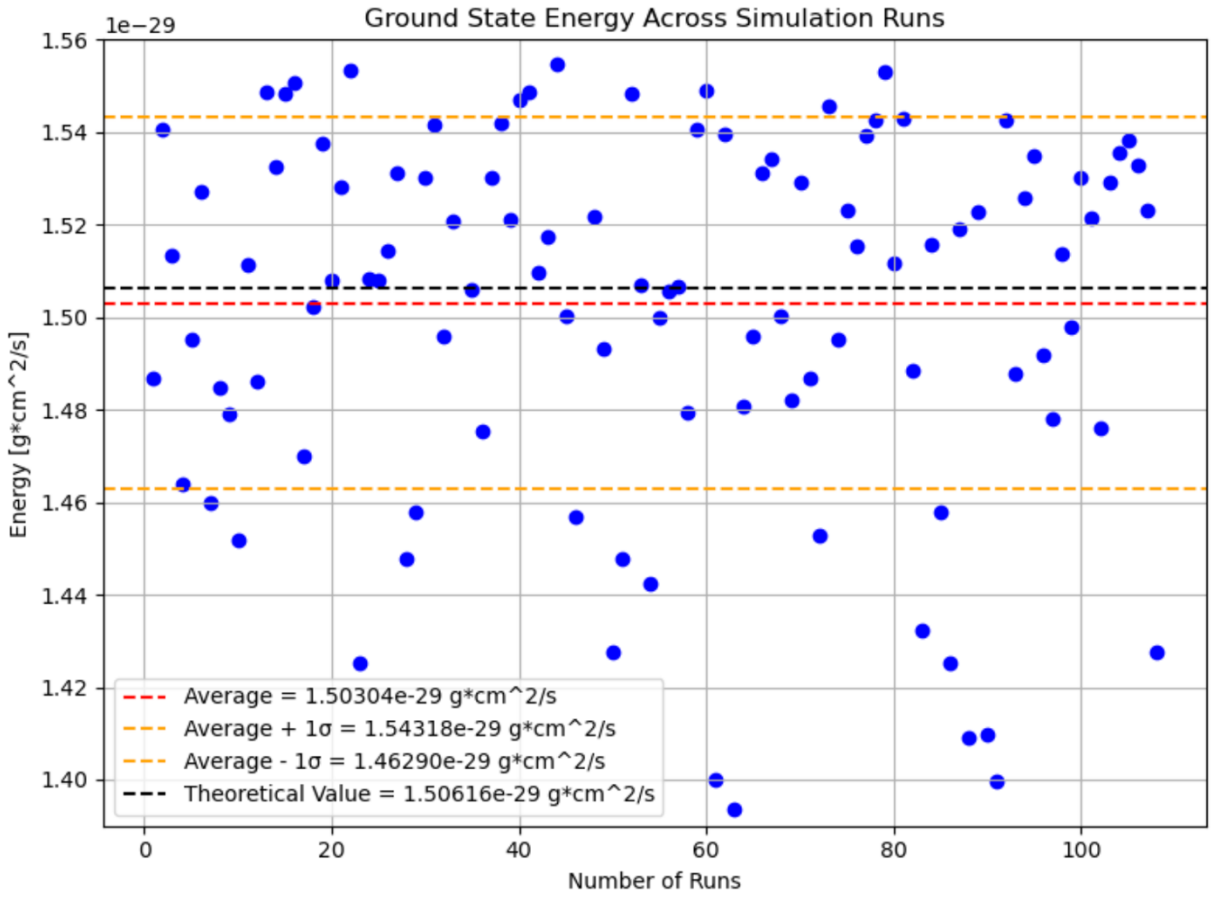
2. Direct Scaling Approach

- This approach was tested using the same algorithm as the dimensionless method.
- As it produced similar results, it will not be discussed further in this presentation.

Results – 1. Dimensionless Approach

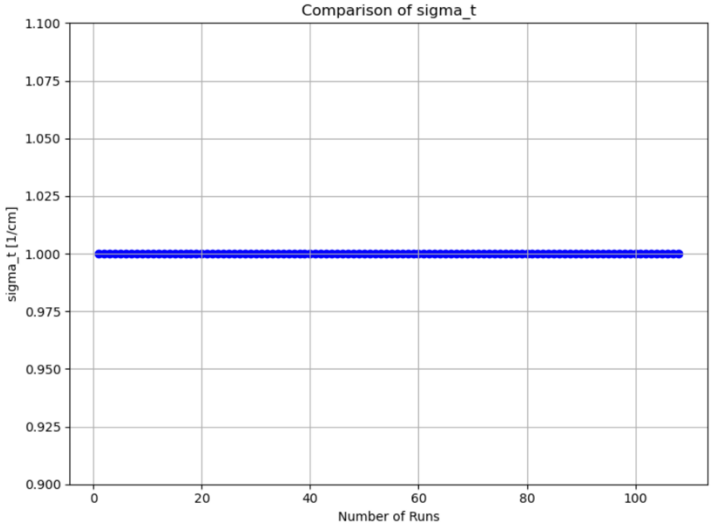
- Input values
  - Boundary condition: 0 (Vacuum)
  - Number of neutrons: 100000
  - Inactive cycles: 50
  - Active cycles: 250
  - Particle: electron
  - Particle mass (kg): 9.10938e-31
  - Potential: 0
  - **Scaling factor: 30**
- Variables
  - width (cm) = 20.0
- **Total runs: 107**

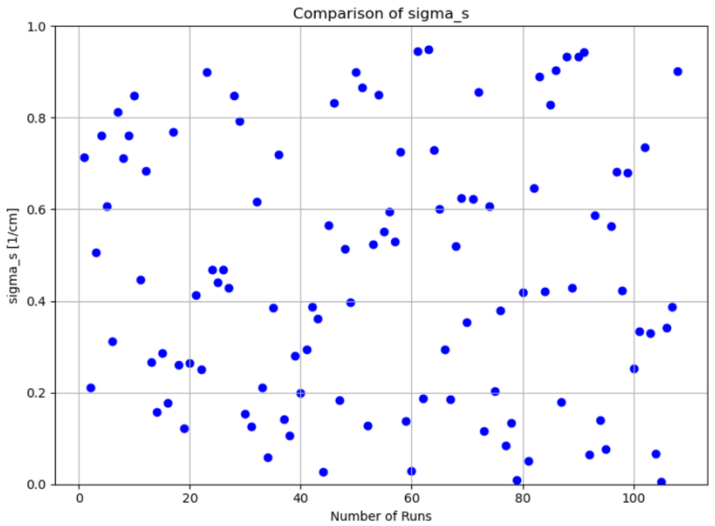
Results – 1. Dimensionless Approach

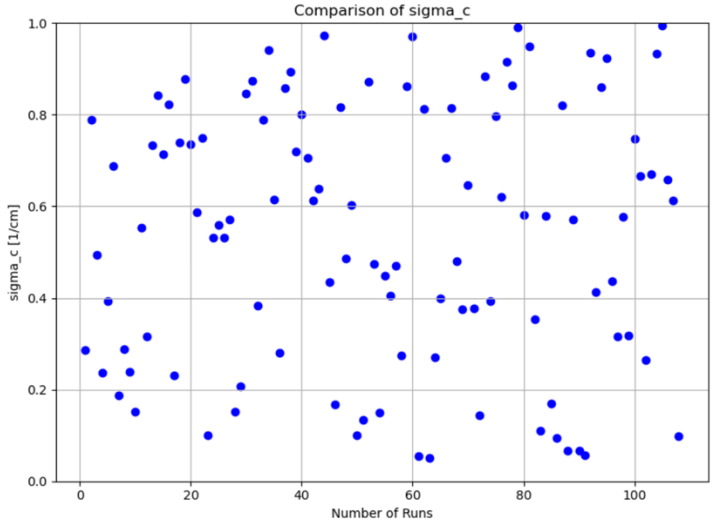


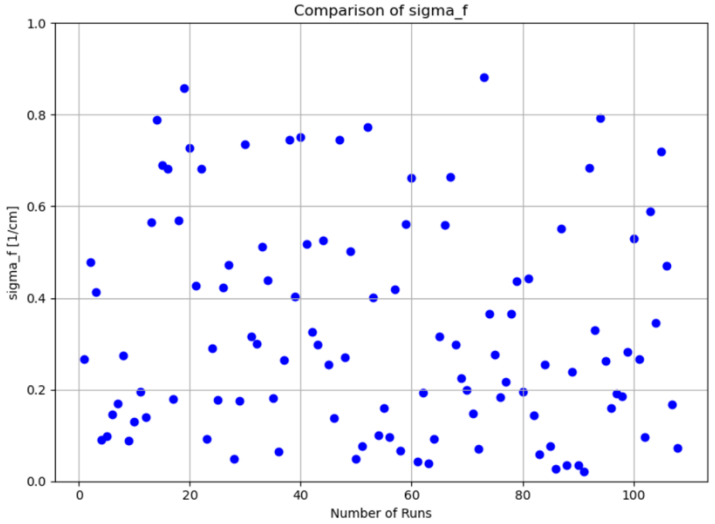
- Theoretical Energy:
  $1.50616e-29\ \mathrm{g}\cdot cm^2/s^2$
- Average Energy $(68.27\%$ Confidence Interval$)$
  $(1.50304 \pm 0.04014)\mathrm{e}-29\ \mathrm{g}\cdot cm^2/s^2$
- The theoretical value lies within the 1σ range of the average simulated energy.

Results – 1. Dimensionless Approach



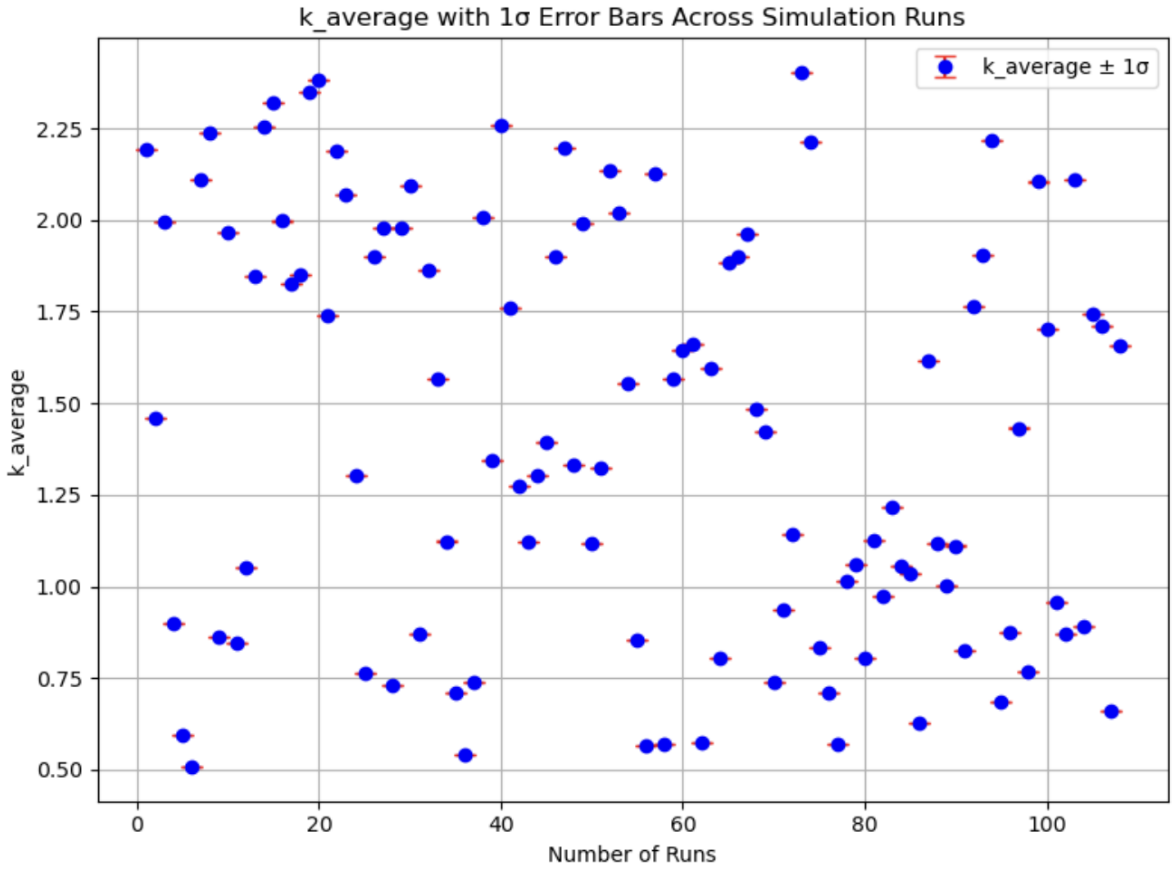






- The scaling factor was chosen to make sigma_t = 1.
- Based on the earlier equation comparison for $V = 0$, Σ_s, Σ_c, and Σ_f are randomly set. Thus, each simulation run gives different cross section values.

Results – 1. Dimensionless Approach



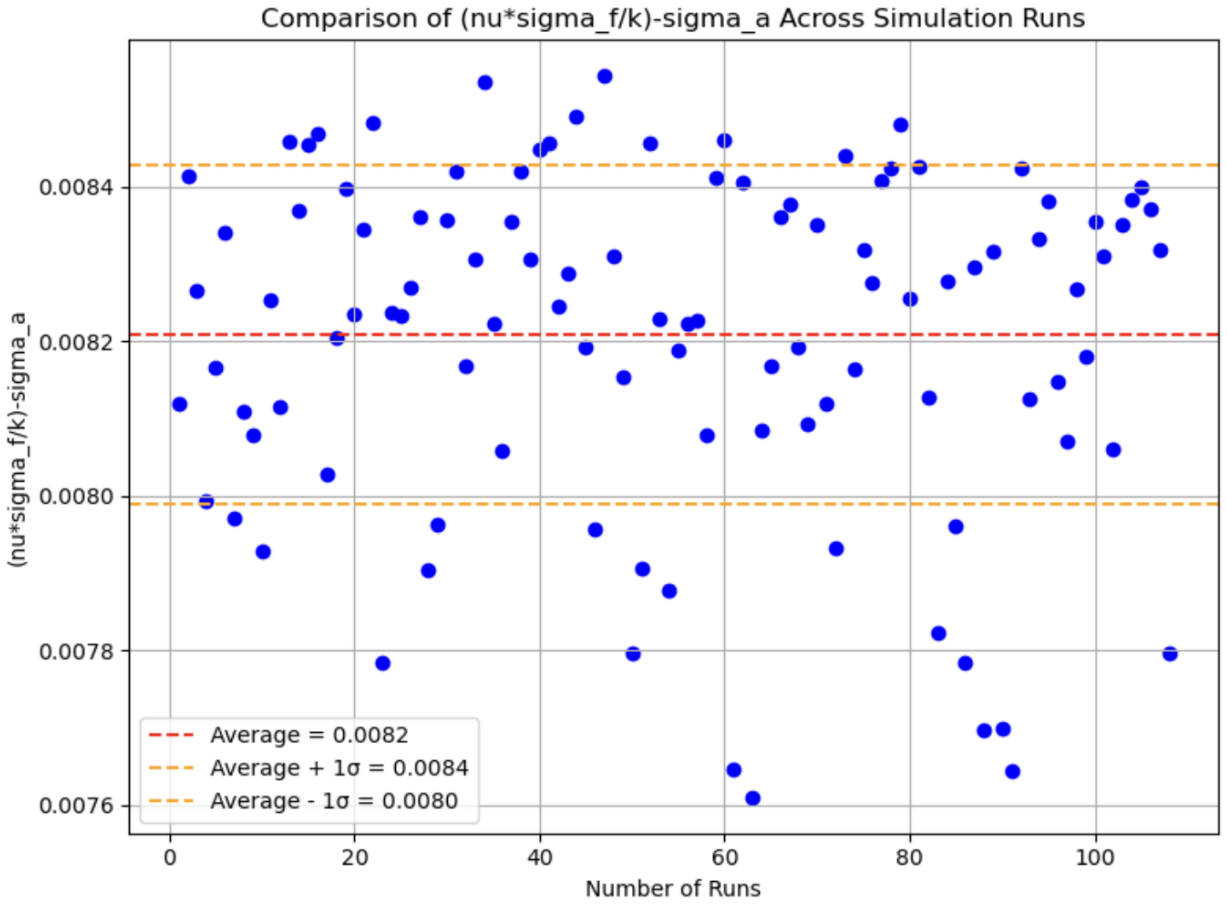
- Each simulation run produces a different k value due to variations in the cross section values.

Results – 1. Dimensionless Approach

- From the cross section plots and k value plots, we observe that each simulation run yields different values.
- However, the computed ground state energy remains within a relatively narrow range: from $1.36e-29$ to $1.56e-29$ with an average of $1.50304e-29$ and a standard deviation of $0.04014e-29$.
- This consistency arises because, when caluclating the energy for $V=0$, the quantity $(\frac{\nu\Sigma_f}{k}-\Sigma_a)$ remains similar across runs, as will be shown in the next slide.

$$E=\lambda\cdot\frac{\hbar^2}{mL}\left(\frac{1}{k}\nu\Sigma_f-\Sigma_a\right)$$

Results – 1. Dimensionless Approach



- Average Value (68.27% Confidence Interval)
  0.0082 ± 0.002

Results – 1. Dimensionless Approach

- To compute the higher modes, a fission matrix computed from a single simulation run was used. The inputs values were as the follows:
- Input values
  - Boundary condition: 0 (Vacuum)
  - Number of neutrons: 100000
  - Inactive cycles: 50
  - Active cycles: 250
  - Particle: electron
  - Particle mass (kg): 9.10938e-31
  - Potential: 0
  - **Scaling factor: 30**
- Variables
  - width (cm) = 20.0

Results – 1. Dimensionless Approach

- Cross section values:
  - $\Sigma_s = 0.592151$
  - $\Sigma_c = 0.198041$
  - $\Sigma_f = 0.209808$
- Computed ground state energy:
  - Theoretical Energy:
    - $1.50616e-29\ \mathrm{g}\cdot cm^2/s^2$
  - Average Energy $(68.27\%\ \text{Confidence Interval})$
    - $(1.50977 \pm 0.00505665)\mathrm{e}-29\ \mathrm{g}\cdot cm^2/s^2$
  - The theoretical value lies within the 1σ range of the average simulated energy.

Results – 1. Dimensionless Approach

- The computed fission matrix is shown below:

```
0.60644 0.14134 0.02545 0.00633 0.00185 0.00054 0.00015 0.00007 0.00001 0.00001 0.00000 −0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.19959 0.74621 0.17860 0.03461 0.00894 0.00250 0.00076 0.00026 0.00008 0.00002 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.03407 0.20656 0.75169 0.18397 0.03555 0.00920 0.00255 0.00079 0.00025 0.00008 0.00003 0.00001 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00795 0.03756 0.20246 0.75339 0.18625 0.03523 0.00922 0.00264 0.00079 0.00023 0.00009 0.00003 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00246 0.00955 0.03787 0.19950 0.75377 0.18813 0.03597 0.00908 0.00264 0.00076 0.00025 0.00008 0.00003 0.00001 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000
0.00067 0.00279 0.00980 0.03700 0.19722 0.75373 0.18977 0.03567 0.00926 0.00269 0.00083 0.00023 0.00007 0.00002 0.00001 0.00000 0.00000 0.00000 0.00000 0.00000
0.00023 0.00087 0.00288 0.00956 0.03699 0.19579 0.75300 0.19014 0.03620 0.00921 0.00259 0.00082 0.00024 0.00007 0.00002 0.00001 0.00000 0.00000 0.00000 0.00000
0.00013 0.00027 0.00082 0.00263 0.00935 0.03681 0.19466 0.75344 0.19041 0.03589 0.00925 0.00266 0.00080 0.00026 0.00008 0.00002 0.00001 0.00000 0.00000 0.00000
0.00003 0.00006 0.00025 0.00082 0.00263 0.00937 0.03658 0.19424 0.75390 0.19142 0.03625 0.00940 0.00272 0.00079 0.00026 0.00008 0.00003 0.00001 0.00000 0.00000
0.00001 0.00001 0.00009 0.00024 0.00082 0.00272 0.00935 0.03632 0.19320 0.75377 0.19306 0.03640 0.00932 0.00274 0.00077 0.00024 0.00008 0.00002 0.00000 0.00000
0.00001 0.00001 0.00003 0.00009 0.00027 0.00079 0.00261 0.00949 0.03618 0.19201 0.75414 0.19249 0.03636 0.00929 0.00270 0.00080 0.00025 0.00007 0.00001 −0.00001
0.00000 0.00000 0.00001 0.00003 0.00009 0.00023 0.00079 0.00267 0.00931 0.03621 0.19177 0.75301 0.19368 0.03626 0.00935 0.00263 0.00078 0.00027 0.00009 0.00002
0.00000 0.00000 0.00000 0.00001 0.00003 0.00008 0.00026 0.00080 0.00264 0.00932 0.03612 0.19065 0.75351 0.19356 0.03629 0.00948 0.00270 0.00070 0.00027 0.00003
0.00000 0.00000 0.00000 0.00000 0.00000 0.00002 0.00009 0.00025 0.00082 0.00262 0.00934 0.03591 0.19029 0.75419 0.19574 0.03699 0.00956 0.00279 0.00084 0.00026
0.00000 0.00000 0.00000 0.00000 0.00000 0.00001 0.00003 0.00007 0.00026 0.00082 0.00269 0.00927 0.03565 0.18972 0.75348 0.19713 0.03720 0.00950 0.00287 0.00076
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00001 0.00004 0.00007 0.00025 0.00083 0.00269 0.00923 0.03539 0.18816 0.75289 0.19996 0.03750 0.00966 0.00234
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00001 0.00003 0.00008 0.00025 0.00081 0.00265 0.00924 0.03567 0.18760 0.75210 0.20321 0.03729 0.00797
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00001 0.00003 0.00008 0.00025 0.00078 0.00258 0.00927 0.03534 0.18376 0.75253 0.20747 0.03374
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00001 0.00002 0.00007 0.00024 0.00079 0.00253 0.00889 0.03393 0.17769 0.74642 0.19933
0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00000 0.00001 0.00002 0.00004 0.00018 0.00054 0.00180 0.00630 0.02554 0.14175 0.60784
```

Results – 1. Dimensionless Approach

- The Wielandt deflation method was applied using the fission matrix to compute the higher k values. These k values were then substitued into the equation below, using the same scaling factor as that applied for the ground state energy.

$$E = \lambda \cdot \frac{\pi^2}{mL}\left(\frac{1}{k}\nu\Sigma_f - \Sigma_a\right)$$

Results – 1. Dimensionless Approach

- Second eigenenergy:
  - Theoretical: $6.02469e-29\ \mathrm{g}\cdot cm^2/s^2$
  - Simulated: $4.80418e-28\ \mathrm{g}\cdot cm^2/s^2$

- Third eigenenergy:
  - Theoretical: $1.35554e-28\ \mathrm{g}\cdot cm^2/s^2$
  - Simulated: $5.29246e-28\ \mathrm{g}\cdot cm^2/s^2$

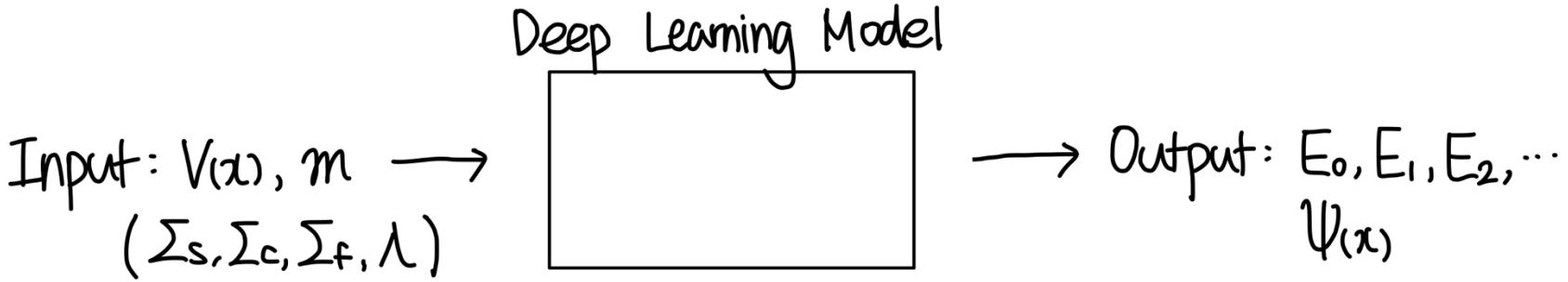
- The results for the higher modes are quite different from the expected values, so more investigation is needed to understand why.

New Idea – Adding Machine Learning

- **NDMC** (Neutron Diffusion Monte Carlo) is an effective and intuitive way to solve the Schrödinger equation, especially for complex potentials.
- It uses cross sections to simulate neutron behavior inside a reactor.
- While results are accurate, running many cycles can take time as cross section vary.
- **Machine learning can enhance this approach** by predicting eigenenergies and wavefunctions directly from the potential and particle mass – without waiting for the full simulation.

New Idea – Adding Machine Learning

- Machine Learning Model



Training the Deep Learning Model

- Inputs: NDMC cross sections $(\Sigma_s,\ \Sigma_f,\ \Sigma_c)$
- Outputs (Two Options):
  - Option 1: Ground state energy and wave function from DMC (Diffusion Monte Carlo) + Higher states from deflation methods
  - Option 2: Ground state energy and wave function from NDMC + Higher states from deflation methods

Why Consider DMC with NDMC?

- **Diffusion Monte Carlo (DMC)** is a quantum Monte Carlo method that calculates the ground state energy of any quantum system within a given error.
- While DMC is **highly accurate** capturing essential quantum features such as phase, coherence, superposition, and interference it cannot easily compute higher excited states. It also directly simulates the wavefunction, which makes it **less intuitive and more complicated to handle** compared to NDMC.

Why Consider DMC with NDMC?

- **Neutron Diffusion Monte Carlo (NDMC)** is a classical Monte Carlo method used in this study to solve the Schrödinger equation. Although it **does not incorporate quantum features,** it can **map potentials and particle mass** into **neutron transport parameters** (cross sections). These parameters effectively **parameterize the quantum system**, making them ideal **inputs for machine learning models**, whereas DMC results are harder to use in this way.
- Additionally, while DMC is excellent for computing ground states, it is **not naturally suited for extracting higher states**. NDMC, on the other hand, allows us to **construct a fission matrix**, from which we can apply **deflation techniques** to systematically compute excited states, an approach that is much more convenient than modifying DMC.
- Therefore, in this work, I plan to use NDMC to generate the fission matrix and machine learning input features and use DMC to compute accurate ground state energies to train our deep learning models.

Comparison of Current Work with Reference Studies

- My research focuses on solving the eigenvalue problem, whereas the reference studies are limited to a basic transport problem, considering only scattering (with no fission or capture).
- The reference study uses a simple linear extrapolation at the boundary. In contrast, I assign negative weights to neutrons that exit the reactor, reflect them back with those weights and apply weight manipulation techniques such as Russian Roulette and forced scattering.
- Rather than just comparing the equations, I transform them into a dimensionless form, providing a more systematic and logical basis for determining the scaling factor.

Comparison of Current Work with Reference Studies

- I compute higher-order eigenstates using deflation methods, extending the analysis beyond the fundamental mode.
- I plan to integrate machine learning with NDMC and DMC methods to accelerate the solution of the Schrödinger equation.
- They use MCNP, while I developed my own codes based on MCNP algorithms.

Things to Consider

- Interpreting neutron flux as a wave function
- Energy computed via NDMC is always greater than zero.
- What is the optimal scaling factor? Can I automate the process of determining it?
- Since cross sections must positive, the potential must always be above zero.
- For the machine learning model, cross section values likely need to be uniquely determined for each input, but this is currently not the case.

To-Do List

1) ~~Study Monte Carlo particle transport method~~ ✓ **Done**
2) ~~Implement code for fundamental mode~~ ✓ **Done**
3) ~~Implement code for higher modes~~ ✓ **Done**
4) Interpret as a Schrödinger equation solution ✓ **In progress**
5) Apply to more complex situations and gather many data samples
   - Various potentials
6) Build, train and evaluate machine learning model
7) Study numerical and QMC methods
8) Compare performance between methods

Resources (Not updated)

- Alex F Bielajew . (2020). *Fundamentals of the Monte Carlo method for neutral and charged particle transport*.
- Brown, F. B. (n.d.). *Monte Carlo Techniques for Nuclear Systems*. Lecture.
- *Boundary conditions - diffusion equation*. Nuclear Power. (2021, October 28). https://www.nuclear-power.com/nuclear-power/reactor-physics/neutron-diffusion-theory/boundary-conditions-diffusion-equation/
- Leppänen, J. (2007). *Development of a new Monte Carlo Reactor Physics Code* (thesis). *Development of a new Monte Carlo reactor physics code*. VTT, Espoo.
- Shentu, J., Yun, S.-H., & Cho, N.-Z. (2007). A Monte Carlo method for solving heat conduction problems with complicated geometry. *Nuclear Engineering and Technology*, *39*(3), 214. https://doi.org/10.5516/net.2007.39.3.207