NE 255: Homework # 02

Due 04 October 2016

Professor Rachel Slaybaugh

Andrew S Voyles

Problem 1

Determine the macroscopic scattering cross section of UO_2 as a function of a generic enrichment factor $\gamma = N_{U-235}/N_{U-238}$ where N is the atom density. Find its value assuming a density of 10 g/cm^3, $\sigma_s^U \simeq 8.9$ b and $\sigma_s^O \simeq 3.75$ b, and 5% weight enrichment.

Solution

The macroscopic cross section for a compound is defined in terms of its constituent atoms:

$$\Sigma_s^{UO_2} = N_U \sigma_s^U + N_O \sigma_s^O \tag{1}$$

As described, we will define a generic enrichment factor, γ, in terms of both atom density N_i and isotopic atom fractions α_i:

$$\begin{aligned} \gamma &= \frac{N_{U-235}}{N_{U-238}} \\ &= \frac{\alpha_{U-235} N_{UO_2}}{(1-\alpha_{U-235})\, N_{UO_2}} \\ &= \frac{\alpha_{U-235}}{1-\alpha_{U-235}} \end{aligned} \tag{2}$$

Inverting this, we can calculate γ in terms of α_{U-235}, which will useful for later on:

$$\alpha_{U-235} = \frac{\gamma}{\gamma+1} \tag{3}$$

The molar mass of UO_2 (in g/mol) can be thus calculated in terms of the atomic enrichment in U-235:

$$\begin{aligned} M_{UO_2} &= M_U + 2M_O \\ &= \alpha_{U-235} M_{U-235} + (1-\alpha_{U-235})\, M_{U-238} + 2M_O \\ &= \frac{\gamma}{\gamma+1} M_{U-235} + \frac{1}{\gamma+1} M_{U-238} + 2M_O \\ &= \frac{\gamma}{\gamma+1}(235) + \frac{1}{\gamma+1}(238) + 32 \end{aligned} \tag{4}$$

The number density of UO_2 (in atoms/cm^3) is thus simply:

$$\begin{aligned} N_{UO_2} &= \frac{\rho_{UO_2} N_A}{M_{UO_2}} \\ &= \frac{\rho_{UO_2} N_A}{\frac{\gamma}{\gamma+1}(235) + \frac{1}{\gamma+1}(238) + 32} \\ &= \rho_{UO_2} N_A \frac{\gamma+1}{3\left(89\gamma+90\right)} \end{aligned} \tag{5}$$

By stoichiometry, $N_U = N_{UO_2}$ and $N_O = 2N_{UO_2}$, making our final expression:

.

$$\begin{aligned} \Sigma_s^{UO_2} &= N_U \sigma_s^U + N_O \sigma_s^O \\ &= \boxed{\rho_{UO_2} N_A \frac{\gamma+1}{3\left(89\gamma+90\right)} \left(\sigma_s^U + 2\sigma_s^O\right)} \end{aligned} \tag{6}$$

For a density of 10 g/cm^3, $\sigma_s^U \backsimeq 8.9$ b and $\sigma_s^O \backsimeq 3.75$ b, and 5% weight enrichment, we can calculate this macroscopic cross section:

$$\begin{aligned} \alpha_{U-235} &= \frac{\frac{\omega_{U-235}}{M_{U-235}}}{\frac{\omega_{U-235}}{M_{U-235}} + \frac{1-\omega_{U-235}}{M_{U-238}}} \\ &= \frac{\frac{5\%}{235}}{\frac{5\%}{235} + \frac{1-5\%}{238}} \\ &= 5.061\% \end{aligned} \tag{7}$$

$$\begin{aligned} \gamma &= \frac{\alpha_{U-235}}{1-\alpha_{U-235}} \\ &= \frac{5.061\%}{1-5.061\%} \\ &= 0.0533 \end{aligned} \tag{8}$$

Thus,

.

$$\begin{aligned} \Sigma_s^{UO_2} &= \rho_{UO_2} N_A \frac{\gamma+1}{3\left(89\gamma+90\right)} \left(\sigma_s^U + 2\sigma_s^O\right) \\ &= \left(10\ {}^{g}/_{cm^3}\right)\left(6.022 \cdot 10^{23}\ {}^{atoms}/_{mol}\right) \frac{0.0533+1}{3\left(89\left(0.0533\right)+90\right) {}^{g}/_{mol}} \left(8.9 + 2 \times 3.75\right) 10^{-24}\ cm^2 \\ &= \boxed{0.3659\ cm^{-1}} \end{aligned} \tag{9}$$

Problem 2

A major challenge in scientific computing is navigating the disparate architectures that comprise new supercomputers.

(a) Look up the Top 10 supercomputing list and briefly describe the architecture of the top three machines. List the number of machines of each type on the top 10 (e.g. X are GPU-accelerated, Y contain MICs, etc.)

Solution

According to the Top500 project[1], the top three machines (ranked in terms of peak achieved floating-point-operations per second, FLOPS) are the Sunway TaihuLight, the TH-2, and Titan:

The Sunway TaihuLight is at the #1 spot on the Top500 list, and is a 93 PFLOP supercomputer in Wuxi, China, which first came online in June 2016. It uses 40,960 SW26010 processors clocked at 1.45 GHz, each of which contains 256 processing cores, for a total of 10,649,600 cores across the cluster. These are custom many-core processors using the RISC architecture, which eschews typical cache structure for inter-core communication for a dedicated network between core clusters, which greatly improves the parallel efficiency of the cores. The machine is supported by 1.3 PBytes of RAM, and consumes approximately 15 MW of power. The machine runs on Sunway RaiseOS, a custom high-performance computing Linux distribution.

The Tianhe-2 is at the #2 spot on the Top500 list, and is a 34 PFLOP supercomputer in Guangzhou, China, which first came online in June 2013. It uses 16,000 computing nodes, each of which contains two Intel Xeon E5-2692 12-core CPUs running at 2.2 GHz supported by 64 GB of RAM, and three Xeon Phi 31S1P co-processors, which contain 57 cores running at 1.1 GHz, supported by 8 GB of RAM. This is a combined 3,120,000 processing cores and 1.34 PBytes of RAM across the system, which consumes 17.6 MW of power. The machine runs on Kylin, a custom high-performance computing Linux distribution.

Titan is at the #3 spot on the Top500 list, and is a 17 PFLOP supercomputer Oak Ridge National Lab, in Oak Ridge, Tennessee, which first came online in October 2012. It uses 18,688 computing nodes, each of which contains one AMD Opteron 6274 16-core CPU running at 2.2 GHz supported by 32 GB of RAM, and one Nvidia Tesla K20X GPU, which contains 2688 CUDA cores running at 732 MHz, supported by 6 GB of RAM. This is a combined 50,532,352 processing cores and 694 TBytes of RAM across the system, which consumes 8.2 MW of power.

In this top 10 list, 6 computers (Sunway TaihuLight, Sequoia, Mira, Trinity, Hazel Hen, Shaheen II) use massively multi-core custom primary CPUs as an architecture, 2 computers (Titan, Piz Daint) are GPU-accelerated, 1 computer (Tianhe-2) uses MICs / co-processors, and 1 computer (K-Computer) uses an alternative customs CPU architectures.

(b) Describe the main characteristics of GPUs, MICs (multi integrated cores), and CPUs-memory, clock speed, structure, etc.

Solution

The CPU is the heart of a computer, the primary integrated circuit responsible for computation. The CPU receives instructions from some compiled binary code, decodes these instructions into a series of electrical commands, and then electrically connects its internal transistors, to perform the requested

[1] https://www.top500.org/

computation or operation. CPUs have small memory buffers called *caches*, which are responsible for holding commands, outputs, and memory addresses, as well as facilitating data I/O between the CPU and the motherboard, which have sizes on the order of a few MB in modern CPUs.

The commands read and executed by a CPU are measured in *clock cycles*, a unit of time used to regulate the operation of the CPU. Clock cycles can be used to measure the time required to complete the electrical operation involved in reading in binary instructions, decoding them, and executing the requested operation. Each operation consists of multiple instruction signals, so the more clock cycles that a CPU can execute per unit time, the more operations it is capable of performing per unit time. Clock speed, the number of clock cycles per second, is measured in Hz, and is a common measure of the internal processing speed of a CPU.

A co-processor or multi-integrated core (MIC) is a sort of secondary processor, which is always paired with at least one primary CPU in a computing node. This design greatly accelerates computation by offloading intense and parallelizable sections of computation from the primary CPUs (which are optimized for several large serial tasks, due to their high clock speed) to the multiple cores in the co-processor, each of which are optimized for small, independent tasks, due to their lower clock speed. This allows a MIC to split a long serial computation into a very short parallel operation, assuming that the code is capable of being translated into an algorithm supporting parallel operation, such as finite element analysis.

A GPU (graphical processing unit) is another type of processor, very similar in design to a MIC. Like a MIC, GPUs are processors comprised of hundreds to thousands of slower-clock speed cores, each of which has its own memory buffer. Traditionally, GPUs are used generate computer images and video, a process which is extremely demanding on the CPU. By moving these processes to a separate processor, the GPU, the CPU is freed up to focus on more intensive serial operations. Since the process of rendering these graphics is a naturally parallel task, the many small GPU cores turn a very CPU-intensive process into a much more trivial parallel task. To handle all of the data used during these parallel tasks, most modern GPUs have their own dedicated memory, on the order of several to tens of GB.

(c) Based on what we've talked about so far, postulate challenges of solving the neutron transport equation in a way that would work on all of these architectures.

Solution

We have discussed the two primary computational solution methods: deterministic methods, which discretize a large problem into multiple small elements, each of which is solved analytically. These problems are often hard to perform in parallel, but are well-suited to solution using many high-clock speed CPUs, and are easily executed on multi-core supercomputers. Due to the nature of matrix diagonalization, many of these finite-element methods are now being ported to parallel computation, however.

Monte Carlo methods, on the other hand, use random sampling of events in a probability distribution to look at aggregate behavior of systems. In neutron transport, we can simulate the lifetime and interactions of a single neutron as it is transported through a volume. By performing this for many ($\gg 10^8$) neutrons, we can solve the NTE for neutron fluxes in various volumes by building up a statistically significant number of particle histories. Since each particle is transported independently, these Monte Carlo problems are very well-suited to parallel computation, and are frequently performed on MIC and GPU-accelerated machines.

Problem 3

We often measure convergence by comparing one iteration to the previous iteration (rather than the solution, since we presumably don't know what it is). Imagine that you have software that gives the following solution vectors:

$$x_{n-1} = \begin{pmatrix} 0.45 \\ 0.95 \\ 0.2 \\ -0.05 \\ 0.6 \end{pmatrix} \qquad x_n = \begin{pmatrix} 0.5 \\ 0.9 \\ 0.3 \\ -0.1 \\ 0.5 \end{pmatrix}$$

Calculate:

(a) The absolute and relative error using the 1 norm.

Solution

For some vector $\mathbf{x}$, the ℓ_1 norm is calculated as:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{N} |x_i| \tag{10}$$

For some norm $\|\cdot\|$, we will define the absolute error between solution vectors x_n and x_{n-1} as:

$$\epsilon_A = \|x_n - x_{n-1}\| \tag{11}$$

Similarly, the relative error is:

$$\epsilon_R = \frac{\|x_n - x_{n-1}\|}{\|x_n\|} = \frac{\epsilon_A}{\|x_n\|} \tag{12}$$

Using these definitions and the ℓ_1 norm:

$$\epsilon_A = \sum (0.05, 0.05, 0.1, 0.05, 0.1)^T = 0.35 \tag{13}$$

$$\epsilon_R = \frac{0.35}{\sum (0.5, 0.9, 0.3, 0.1, 0.5)^T} = 0.1522 \tag{14}$$

(b) The absolute and relative error using the 2 norm

Solution

For some vector $\mathbf{x}$, the ℓ_2 norm is calculated as:

Problem 3 continued on next page. . .

$$\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{N} x_i^2} \tag{15}$$

Using these definitions:

$$\epsilon_A = \sqrt{\sum (0.0025, 0.0025, 0.01, 0.0025, 0.01)^T} = 0.1658 \tag{16}$$

$$\epsilon_R = \frac{0.1658}{\sqrt{\sum (0.25, 0.81, 0.09, 0.01, 0.25)^T}} = 0.1397 \tag{17}$$

(c) The absolute and relative error using the infinity norm.

What is most restrictive (that is, what would cause the code to converge first)?

Image that now $x_{n-1} = (0.49, 0.92, 0.4, -0.09, 0.51)^T$. Recalculate the convergence values.

What do you observe? What does that mean about how you might select convergence criteria?

Solution

For some vector $\mathbf{x}$, the ℓ_∞ norm is calculated as:

$$\|\mathbf{x}\|_\infty = \max_{1 \le i \le N} |x_i| \tag{18}$$

Using these definitions:

$$\epsilon_A = \max (0.05, 0.05, 0.1, 0.05, 0.1)^T = 0.1 \tag{19}$$

$$\epsilon_R = \frac{0.1}{\max (0.5, 0.9, 0.3, 0.1, 0.5)^T} = 0.111 \tag{20}$$

Table 1: Tabulated results for the various error criteria.

Norm	ϵ_A	ϵ_R
ℓ_1	0.35	0.1522
ℓ_2	0.1658	0.1397
ℓ_∞	0.1	0.1111

These results can be seen tabulated in Table 1. For most iterative methods, when the true solution is unknown, iteration continues until the error between two subsequent iterations falls below a set threshold. This leaves us with two ways to interpret convergence speed in the context of this problem. Since the ℓ_∞ norm has the smallest of both the absolute and relative errors, it would predict that the two solution vectors are the closest to convergence, i.e., it would be the first to converge starting at the current timestep, assuming that the iterative method employed has monotonically decreasing errors between iterations. A second interpretation would be to look at the ℓ_1 norm, which has the largest of both the absolute and relative errors. Many iterative solvers would see this as being "far" away from convergence,

Problem 3 continued on next page. . .

and take either a larger time step, or iterate more aggressively towards the solution, potentially reaching convergence first, at the risk of overshooting due to the "bang-bang" nature of aggressive iteration.

Repeating our analysis for the second x_{n-1} vector (with calculations not listed here for clarity, but verified in the `hw02_03.m` script included in the parent repo), we see the results tabulated in Table 2. Through inspection of these results, we that while the absolute errors have decreased in magnitude somewhat significantly from before, the relative errors have decreased by a smaller magnitude. Since the new x_{n-1} vector was closer to x_n than the old x_{n-1} vector, this illustrates why relative errors are typically used as convergence tolerances, as they normalize the scale of an error to its nominal value.

It is also very important to point out that the ℓ_∞ norms do not change at all for our new x_{n-1} vector. This is because , unlike the ℓ_1 and ℓ_2 norms which look at the contributions of all errors between a pair of solution vectors, the ℓ_∞ only looks at the magnitude of the largest error contribution. While this can provide some stability against overshoot/undershoot by only focusing on minimizing the most significant error, it ignores changes in the larger solution space, which can be used to give more feedback about one's proximity to convergence.

Table 2: Tabulated results for the various error criteria, using the new x_{n-1}.

Norm	ϵ_A	ϵ_R
ℓ_1	0.15	0.0652
ℓ_2	0.1034	0.0871
ℓ_∞	0.1	0.1111

Problem 4

You have a piece of software that takes mesh spacing as an input variable. Imagine that you have the following relative error values for each mesh spacing (h) / number of mesh cells (N cells):

h	N cells	rel err
1.0	8	8.44660179e-03
0.5	16	2.30286448e-03
0.1	80	9.84273963e-05
0.05	160	2.48043656e-05
0.01	800	9.98488163e-07

One of the ways we characterize method performance is the order of convergence. We'd like to know how the error changes as we change the resolution of our discretization, in this case, mesh spacing.

Using a log-log plot, plot relative error as a function of

(a) mesh spacing and

Solution

Both plots shown here can be generated by running the `hw02_04.m` script included in the parent repo.

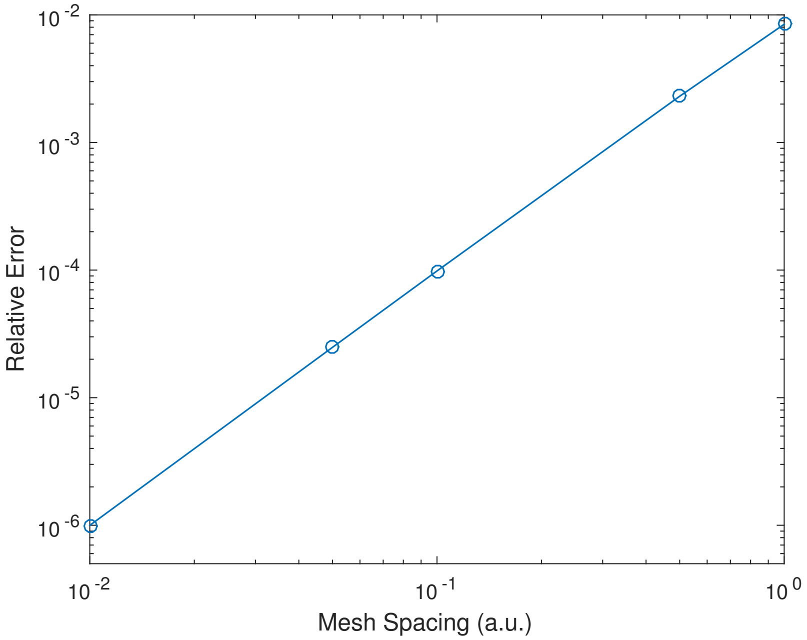

Figure 1: Relative error as a function of mesh spacing.

(b) cell count.

What is the functional relationship between error and mesh spacing / number of cells?

Problem 4 continued on next page. . .

Solution

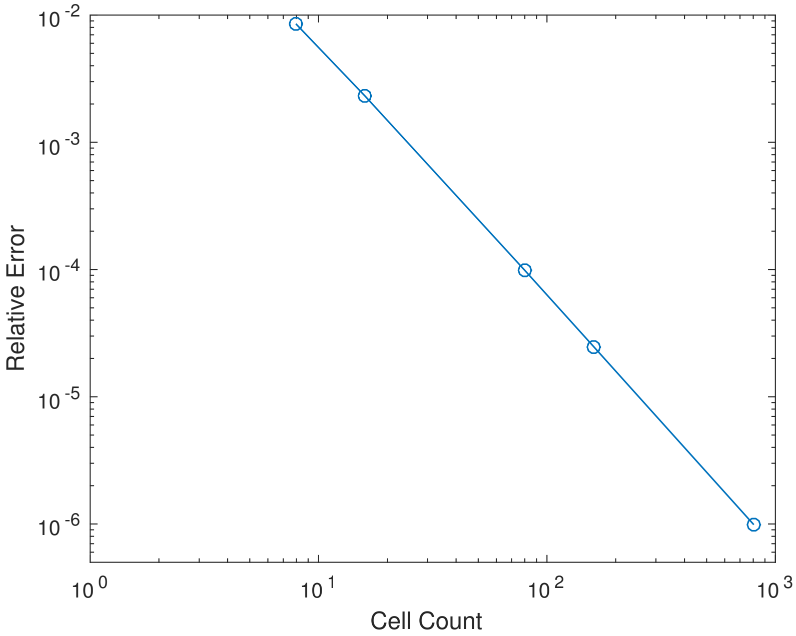

Figure 2: Relative error as a function of cell count.

We see in Figure 1 that the points nearly perfectly across a line, whose slope was calculated as $k = \dfrac{\Delta\,(rel\ err)}{\Delta h} = 1.95$, which can be rounded to 2.0. Since this is in log-log space, the relative error, ϵ_R, can be concluded to be functionalized as $\boxed{\epsilon_R \propto h^2.}$

Similarly, in Figure 2 the points nearly perfectly across a line, whose slope was calculated as $k = \dfrac{\Delta\epsilon_R}{\Delta N} = -1.95$, which can be rounded to -2.0. Since this is in log-log space, the relative error, ϵ_R, can be concluded to be functionalized as $\boxed{\epsilon_R \propto {}^{1}\!/_{N^2}.}$

Problem 5

What are six underlying assumptions in the neutron transport equation? Write at least one sentence explaining what each assumption means and why we need or want to make it.

Solution

(a) All particles are zero-dimensional points, which behave classically, and ignoring quantum / fine structure effects. This means the the effective de Broglie wavelength of transport particles (primarily neutrons) is far smaller than the atomic length scale of the materials they are being transported through. This allows us to neglect quantum effects, and generalize the system, although later perturbations may be introduced, namely through cross sections.

(b) Neutrons travel in straight paths in between interaction vertices. This saves the need to parameterize the transport and statistical nature of particle transport even further. By choosing a suitable coordinate system, and properly discretizing the system, particle transport can be achieved with acceptable resolution.

(c) Neglect neutron-neutron interactions. All transported particles are independent. This linearizes the NTE, and allows for parallel calculations of the each transported particle.

(d) The system volume is time-independent, i.e., $\frac{dV}{dt} = 0$. Since the NTE is fundamentally accounting the neutrons in some control volume, this simplifies the transport, as a time-dependent volume would be 1) far harder to model in the first place, and 2) require changing boundary conditions, which are difficult to determine *a priori* as a function of time.

(e) The boundary of the control volume is a non-reentrant surface, that is, any neutrons which exit the control volume do not reenter at another point. This saves the trouble of tracking neutrons outside our control volume, which we typically define as slightly larger than the area of interest for a problem.

(f) Material properties are isotropic, time-independent, and do not have significant deviations about an expected value. These prevent us from needing to do material properties at an atomic-scale resolution, but instead use bulk properties representative of the material. In addition, over the length of typical particle steps, we expect most matter to be semi-homogeneous and isotropic.

Problem 6

Consider the transport equation:

$$\begin{aligned} \underbrace{\frac{1}{v}\frac{\partial\psi}{\partial t}(\mathbf{r},E,\hat{\Omega},t)}_{\text{A}} + \underbrace{\hat{\Omega}\cdot\nabla\psi(\mathbf{r},E,\hat{\Omega},t)}_{\text{B}} + \underbrace{\Sigma_t(\mathbf{r},E)\psi(\mathbf{r},E,\hat{\Omega},t)}_{\text{C}} = \underbrace{S(\mathbf{r},E,\hat{\Omega},t)}_{\text{D}}. \\ \underbrace{\int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\to E,\hat{\Omega}'\to\hat{\Omega})\psi(\mathbf{r},E',\hat{\Omega}',t)d\hat{\Omega}'dE'}_{\text{E}} \\ + \underbrace{\frac{\chi_p(E)}{4\pi}\int_0^\infty\int_{4\pi}\nu(E')\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},E',\hat{\Omega}',t)d\hat{\Omega}'dE'}_{\text{F}} \end{aligned} \tag{21}$$

(a) Briefly describe what each term in the Transport Equation physically represents.

Solution

A) Time rate of change of the neutron angular flux, the change in the neutron angular flux for an energy group over the entire volume, as a function of time..

B) Streaming losses, the rate at which neutrons exit the control volume.

C) Total interaction losses, the rate at which neutrons are absorbed or outscattered from an energy and solid angle group.

D) External source gains, the rate at which neutrons enter the system, or from a generic point/line/distributed/etc source (other than fission) in the system.

E) Inscattering source gains, the rate at which neutrons are scattered into an energy and solid angle group.

F) Fission source gain, the rate at which neutrons are born through fission, into a particular energy group.

(b) Rewrite the time independent form of the equation to include azimuthal symmetry. Show the steps needed to get there.

Solution

The first step is to make the time-independent assumption:

$$\frac{\partial\psi}{\partial t} = 0 \tag{22}$$

This also removes all time dependence terms:

$$\begin{aligned}\hat{\Omega}\cdot\nabla\psi(\mathbf{r},E,\hat{\Omega})+\Sigma_t(\mathbf{r},E)\psi(\mathbf{r},E,\hat{\Omega}) &= S(\mathbf{r},E,\hat{\Omega})\\ &+\int_0^\infty\int_{4\pi}\Sigma_s(\mathbf{r},E'\to E,\hat{\Omega}'\to\hat{\Omega})\psi(\mathbf{r},E',\hat{\Omega}')d\hat{\Omega}'dE'\\ &+\frac{\chi_p(E)}{4\pi}\int_0^\infty\int_{4\pi}\nu(E')\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},E',\hat{\Omega}')d\hat{\Omega}'dE'\end{aligned}\tag{23}$$

For the azimuthal symmetry assumption, scattering is only a function of $\mu=\hat{\Omega}'\cdot\hat{\Omega}$, the cosine of the scattering angle. This gives us the simplifications:

$d\hat{\Omega}=\sin(\theta)d\theta d\varphi=d\mu d\varphi;\quad \mu=\cos(\theta);\ d\mu=\sin(\theta)d\theta$.

$$\int_{4\pi}d\hat{\Omega}=\int_0^{2\pi}d\varphi\int_{-\pi/2}^{\pi/2}\sin(\theta)d\theta=\int_0^{2\pi}d\varphi\int_{-1}^{1}d\mu=4\pi\tag{24}$$

$$\psi(\mathbf{r},\hat{\Omega},E)d\hat{\Omega}=\psi(\mathbf{r},\varphi,\mu,E)d\varphi d\mu=\psi(\mathbf{r},\mu,E)d\varphi d\mu\tag{25}$$

The scattering cross section is no longer dependent on φ:

$$\Sigma_s(\mathbf{r},\hat{\Omega}'\to\hat{\Omega})\to\Sigma_s(\mathbf{r},\hat{\Omega}'\cdot\hat{\Omega})=\Sigma_s(\mathbf{r},\mu)\tag{26}$$

Thus,

$$\int_{4\pi}d\hat{\Omega}\,\psi(\mathbf{r},\hat{\Omega},E)=\int_0^{2\pi}d\varphi\int_{-1}^{1}d\mu\,\psi(\mathbf{r},\hat{\Omega},E)=2\pi\int_{-1}^{1}d\mu\,\psi(\mathbf{r},\mu,E)\tag{27}$$

Finally, the fission term can now be written in terms of the scalar flux:

$$\begin{aligned}\frac{\chi_p(E)}{4\pi}\int_0^\infty\int_{4\pi}\nu(E')\Sigma_f(\mathbf{r},E')\psi(\mathbf{r},E',\hat{\Omega}')d\hat{\Omega}'dE' &= \frac{\chi_p(E)}{4\pi}\int_0^\infty\nu(E')\Sigma_f(\mathbf{r},E')dE'\int_{4\pi}\psi(\mathbf{r},E',\hat{\Omega}')d\hat{\Omega}'\\ &=\frac{\chi(E)}{2}\int_0^\infty dE'\,\nu(E')\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E')\end{aligned}\tag{28}$$

Combining these all together:

$$\begin{aligned}\mu\cdot\nabla\psi(\mathbf{r},E,\mu)+\Sigma_t(\mathbf{r},E)\psi(\mathbf{r},E,\mu) &= S(\mathbf{r},E,\mu)\\ &+2\pi\int_0^\infty dE'\int_{-1}^{1}\Sigma_s(\mathbf{r},E'\to E,\mu')\psi(\mathbf{r},E',\mu')d\mu'\\ &+\frac{\chi(E)}{2}\int_0^\infty dE'\,\nu(E')\Sigma_f(\mathbf{r},E')\phi(\mathbf{r},E')\end{aligned}\tag{29}$$

Problem 7

Solve the following differential equation by hand:

$$\frac{d^2y}{dx^2} + 3y(x) = \sin(x) \qquad x \in [0, 1]$$

where y(0) = 1, y(1) = 3.

What simplifications to the transport equation would have been required to get an equation form that looks like this?

Solution

This differential equation will be solved through the use of Laplace transforms, which make trivial the solution of non-homogeneous differential equations, even with an initial condition for $\frac{dy}{dx}$ not provided. Taking the Laplace transform of both sides:

$$\begin{aligned} \mathscr{L}\left(\frac{d^2y}{dx^2} + 3y(x) = \sin(x)\right) \\ \left(s^2Y(s) - s \times y(x=0) - \frac{dy(x=0)}{dx}\right) + 3Y(s) &= \frac{1}{s^2+1} \\ \left(s^2Y(s) - s - y'(0)\right) + 3Y(s) &= \frac{1}{s^2+1} \\ \left(s^2+3\right)Y(s) &= s + y'(0) + \frac{1}{s^2+1} \end{aligned} \tag{30}$$

Thus, we obtain our Laplace-space transfer function. We can now back-calculate our y(x) solution, by partial fraction decomposition:

$$\begin{aligned} Y(s) &= \frac{\left(s + y'(0)\left(s^2+1\right)+1\right)}{(s^2+1)(s^2+3)} \\ &= \frac{1 + s^3 + s^2y'(0) + s + y'(0)}{(s^2+1)(s^2+3)} \\ &= \frac{{}^{1}\!/_{2}}{s^2+1} + \frac{{}^{1}\!/_{2}}{s^{+}3} + \frac{s}{s^2+3}\frac{y'(0)}{s^2+3} \end{aligned} \tag{31}$$

Inverting back into physical space,

$$\begin{aligned} y(x) &= \mathscr{L}^{-1}(Y(s)) \\ &= \frac{1}{2}\sin(x) - \frac{1}{2\sqrt{3}}\sin\left(x\sqrt{3}\right) + \cos\left(x\sqrt{3}\right) + \frac{y'(0)}{\sqrt{3}}\sin\left(x\sqrt{3}\right) \\ &= \frac{1}{2}\sin(x) + \cos\left(x\sqrt{3}\right) + \left(\frac{y'(0)}{\sqrt{3}} - \frac{1}{2\sqrt{3}}\right)\sin\left(x\sqrt{3}\right) \\ &= \frac{1}{2}\sin(x) + \cos\left(x\sqrt{3}\right) + C\sin\left(x\sqrt{3}\right) \end{aligned} \tag{32}$$

We can now use our other initial condition to find C:

Problem 7 continued on next page. . .

$$3 = \frac{1}{2}\sin(1) + \cos\left(\sqrt{3}\right) + C\sin\left(\sqrt{3}\right) \tag{33}$$

This has solution for:

$$C = \frac{-\left(2\cos\left(\sqrt{3}\right) + \sin(1) - 6\right)}{2\sin\left(\sqrt{3}\right)} \approx 2.7758 \tag{34}$$

Thus,

$$\boxed{y(x) = \frac{1}{2}\sin(x) + \cos\left(x\sqrt{3}\right) + 2.7758\sin\left(x\sqrt{3}\right)} \tag{35}$$

Many simplifications to the NTE are needed to obtain an equation of this form. Assuming steady-state removes all time dependence. The system is clearly 1-D, unless x is actually a spatial vector. As there is no angular variable, azimuthal symmetry has been applied, and the system integrated over all μ. Since there is no energy dependence, the system is a one-speed model, also removing inscattering and outscattering. As a result, the total, scattering, and fission cross section integrals have been integrated over all *mu*, and lumped into a term linear in neutron *scalar* flux, here represented as y(x). Finally, the diffusion approximation has been introduced, leading to the $\frac{d^2y}{dx^2}$ term.

Problem 8

At what energy is the lowest isolated resonance of ^{235}U, ^{238}U, ^{239}Pu, ^{240}Pu, ^{241}Pu, and ^{242}Pu? Why do we care about that?

Solution

For this problem, let us define the resonances as those occurring in the radiative capture (n, γ) neutron cross section for the actinides listed above. However, we could just as easily have looked at the neutron total cross section, which is the sum of the absorption (capture + fission) and scattering cross sections. This is is because, for some nucleus $^{\mathrm{A}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$, resonances at a neutron energy E_n correspond to finite-width states in the nucleus $^{\mathrm{A+1}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$ at $E_x = S_N + E_N$, where S_N is the neutron separation energy in $^{\mathrm{A+1}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$. The existence of a resonance implies that that excited state can be populated by an incident neutron on $^{\mathrm{A}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$. Potential elastic scattering is a coherent process, so it is independent of excited states, and thus has smooth curvature. However, resonance elastic scattering and inelastic scattering both require the existence of a state at $E_x = S_N + E_N$ in $^{\mathrm{A+1}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$, since the incident neutron is temporarily captured in a compound nucleus, leading to resonances in their cross sections.

Likewise, all absorption reactions require this same state at $E_x = S_N + E_N$ in $^{\mathrm{A+1}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$, since they form a compound nucleus $^{\mathrm{A+1}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$, either temporarily (fission, (n,x)) or permanently (capture). However, all of these channels are (for a given neutron at E_n) populating the same excited state in $^{\mathrm{A+1}}_{\mathrm{Z}}\mathrm{X}_{\mathrm{N}}$, causing them to all have resonances at the same energies, at least for the low-energy side of the resonance region.

Thus, by viewing the plots of radiative capture (n, γ) neutron cross sections from ENDF-VII/B[2], we can observe the lowest-energy isolated resonances for each of the above isotopes. These results are compiled in Table 3.

Table 3: Lowest isolated capture resonances for various actinide isotopes.

	E_n (eV)
^{235}U	0.27
^{238}U	6.67
^{239}Pu	0.29
^{240}Pu	1.06
^{241}Pu	0.26
^{242}Pu	2.68

These resonances are important in neutron transport for several reasons. At and below these energies, scattering reaction channels becomes essentially closed off, as scattering is highly unlikely (elastic) or impossible (inelastic, below the threshold for the first excited state in a compound nucleus) at low energies. Thus, absorption becomes the dominant reaction channel, by approximately 2-3 orders of magnitude in cross section. Since these are all actinides, this means that below these energies, fission (for odd A) or capture (for even A) become the dominant reaction channels. This is illustrated for two select isotopes, ^{238}U and ^{241}Pu, seen plotted from ENDF-VII/B in Figure 3 and Figure 4, respectively. This implies that below this energy, resonance escape is no longer possible for neutrons. In discretizing a large problem, cutoffs like this can be used to improve computation time, by setting energy bounds outside of which, various components of neutron transport can be ignored.

[2] `https://www-nds.iaea.org/exfor/endf.htm`

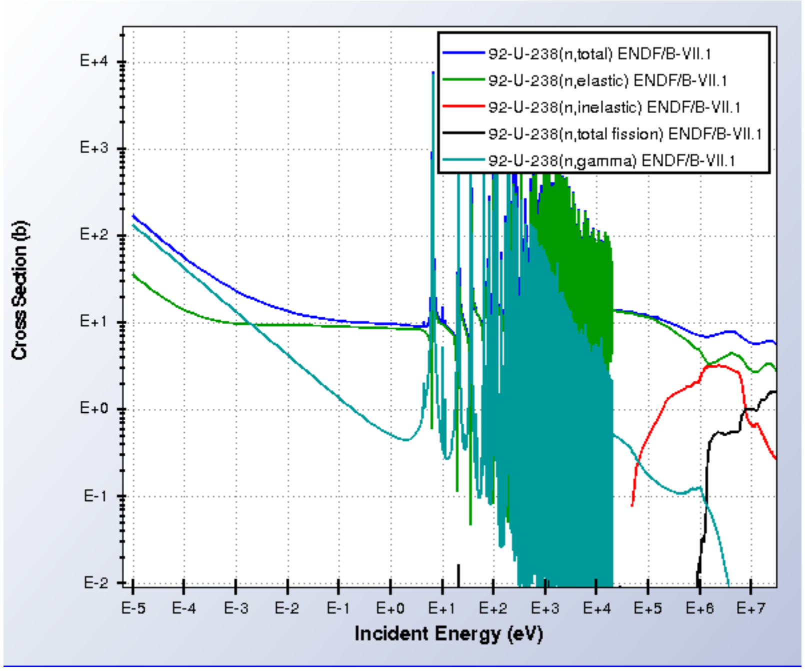

Figure 3: ^{238}U Neutron Cross sections

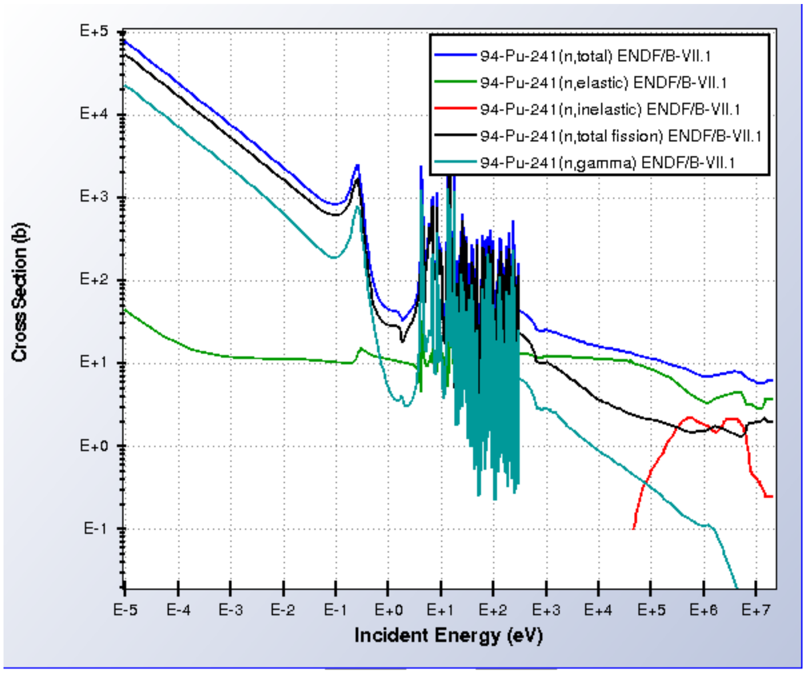

Figure 4: ^{241}Pu Neutron Cross sections