

Chapter 1

Nuclear Engineering Analysis

Nuclear engineering analysis is a broad subject title. Nuclear engineering and medical physics are disciplines that are ultimately very applied; while the “pure sciences” are often afforded the luxury of only considering hypothetical problems, nuclear engineers must respond to delivering real engineering systems to benefit humanity, e.g. providing nuclear power, advanced imaging devices, curing cancer, and the like. Therefore, nuclear engineers must design systems and devices that are based on nuclear interactions using subatomic particles that are not detectable by the ordinary human senses.

A fundamental and practical understanding of mathematics and physics must synergistically be employed to be effective as a nuclear engineer or medical physicist. The successful nuclear engineer must be adaptable, and able to create appropriate mathematical models, so that those models may be used to answer questions and solve real engineering problems. How does one achieve this? The answer is education, training, hard work, and practice.

In most cases, the inherent complexity of the systems involved in nuclear related disciplines require computation. Models are often implemented as computer programs; almost every “real” problem in nuclear engineering requires the use of a computer in some manner, and programming skills among any variety of languages and tools is essential. It is envisioned that users of this text will be assigned problems that involve computation, particularly as progress through the material culminates in solving partial differential equations.

Computer Programming

Computer programming plays several important roles in nuclear engineering analysis: modeling problems, exploring new ideas, automating well-known techniques, and creating new tools for other engineers. A programming language is a tool for creating computer programs. As engineers, we should be aware of each tool's strengths so that we can choose the right tool for each job. In this book, a few programming languages are noted: *Mathematica*, *TK-Solver*, and FORTRAN or C.

Mathematica is an advanced commercial programming language with built-in support for common tasks in science, engineering, and mathematics. *Mathematica* is particularly well-suited to trying out new ideas or doing your homework for you. This tool is expensive, but worth the price (especially with a student version discount).

TK-Solver is a useful tool for modeling engineering problems and solving systems of equations. One advantage of TK-Solver is that it is often possible to model new systems without doing any new programming. Variables can be interchangeably mixed at will as input or output, and therefore TK-Solver models can be readily used as “design optimization” tools.

FORTRAN or **C** languages are often used for programs that need to be fast, such as for neutron transport simulators. You may need to know FORTRAN if you are trying to improve processing data from a “legacy” nuclear application, because FORTRAN has always been popular among nuclear engineers; many codes that have been “nuclear certified” in FORTRAN may never be re-written in other languages, since certification in a new language is cost-prohibitive. Therefore, it is envisioned that FORTRAN will be alive and well in the foreseeable future simply due to its widespread (and certified) use in the nuclear industry. Message Passing Interface (MPI) libraries have enabled FORTRAN and C codes to be readily parallelized on multiprocessor MIMD (multiple instruction, multiple data) computers.

Later, if you find yourself working on a programming project which will last more than a month, it would be worthwhile to spend a week mastering a programming language that is well-suited to a particular task. Choosing the right tool can speed up the work, and often also results in a higher quality result. Flexibility in programming is an important key for success, and some alternative programming tools are OCaml, Python, and a few others.

OCaml, and its Microsoft equivalent **F#**, are similar to Fortran, but with many improvements. Arrays are easier to use, and there is support for additional data structures, which are helpful when optimizing a program for large data sets. Also, OCaml programs tend to be shorter than similar programs written in Fortran.

Python is well-suited for creating websites [Django] or graphical user interfaces and using programs that were written in other languages. Python is very easy to learn, and Python programs can often be understood by engineers who have never used Python. **Numpy** is numerical python, a powerful library package imported in python that expands the use of python as a numerical solver.

Erlang makes it much easier to create programs that are distributed across many computers. Erlang is also capable of using programs that were written in other languages, so a high-performance computer program could use OCaml for fast calculations while Erlang takes care of coordinating tasks between computers.

R was designed specifically for statistical analysis and is also a good choice for plotting. Engineers who expect to dedicate many years of their careers to computer programming may also learn **Scheme**, **Smalltalk**, and **Haskell** to broaden their understanding of what is possible in computing.

Any of the above languages have their advantages and disadvantages. In any event, nuclear engineers must be adaptable programmers to be effective, even if to successfully navigate their educational necessities. In

addition, what separates nuclear engineers from all other engineering disciplines is that nuclear engineers have the added requirement of uniquely understanding radiation, radiation interaction, and radioactive materials.

Radiation

The word *Radiation* often refers to “ionizing” radiation, ionizing a gas through which the radiation passes. Other radiation, e.g. radio waves, is “non-ionizing”. *Ionization* is the process in which a neutral atom or molecule is given a net electrical charge. The amount of energy to remove the least tightly bound e^- from an atom is the first ionization potential. In the early 1900’s, many changes were occurring in Physics that would have a profound influence on 20th century theory. In particular, discussions surrounding quantum theory and the electromagnetic spectrum were compelling. Discoveries included:

EM Waves, Electrons, X-rays, Natural Radioactivity,
 α (He nuclei), β (+,- electrons), and γ (photons) radiation

Radiation Emission for Nuclear Engineers or Medical Physicists refers to a release of energy. Particles and waves that have energy actually have a “particle-wave” duality.

De Broglie Waves

Louis-Victor-Pierre-Raymond, 7e duc De Broglie, a French physicist best known for his research on quantum theory and for his discovery of the wave nature of electrons. DeBroglie was awarded the 1929 Nobel Prize for Physics; he was born Aug. 15, 1892, Dieppe, France, and died March 19, 1987, in Paris. DeBroglie Established particle-wave duality of matter—in certain instances, matter acts as a particle; in other instances it acts as a wave.

Classical Particle

Position (x, y, z) $\vec{r} = \langle x, y, z \rangle$ position vector

Momentum $\vec{p} = m\vec{v}$

$E = h\nu = h\frac{c}{\lambda}$ where ν is the frequency and λ is the
Wavelength

$$p = \left(\frac{hc}{\lambda}\right)\frac{1}{c} \Rightarrow \frac{h}{\lambda} \rightarrow \left[\lambda = \frac{h}{p}\right] \quad \vec{p} = m\vec{v}$$

DeBroglie proposed that any material body will have a wavelength associated with its momentum (motion)...Actually this has been proven for e^- 's diffracted through a slit, like waves. Given the deBroglie particle-wave duality, we can find the deBroglie wavelength of a 10 g pellet moving at 10 m/s (fired from a paintball gun).

$$\begin{aligned} \lambda &= \frac{h}{mv} = \frac{6.63 \times 10^{-34} \text{ J} \cdot \text{s}}{0.01 \text{ kg} \cdot 10 \text{ m/s}} \frac{\text{kg} \cdot \text{m}^2}{\text{s}^2} \\ &= 6.63 \times 10^{-33} \text{ m} \end{aligned}$$

Since $1 \text{ \AA} = 10^{-10} \text{ m}$, this is $6.63 \times 10^{-23} \text{ \AA}$

Note that this is very small, in fact, it is too small to measure!

A more practical Example of DeBroglie's duality theory pertains to the radius of the nucleus. The radius of a nucleus is given by:

$$r_{nuc} = 1.2 \cdot 10^{-15} A^{1/3} \text{ meters, where } A \text{ is the mass number}$$

What minimum energy of photons might be used to probe an object $0.1 \times 10^{-9} \text{ m}$ (0.1 nm) in size?

$$E = \frac{hc}{\lambda} = \frac{6.63 \times 10^{-34} \text{ J} \cdot \text{s}}{0.1 \times 10^{-9}} \frac{3 \times 10^8 \text{ m/s}}{1.602 \times 10^{-16} \text{ J/keV}} = 12.4 \text{ keV}$$

The research of Planck, deBroglie, Einstein, and others led to the advent of Quantum Theory.

Quantum Theory

Quantum Theory – that energy can only be exchanged in “packets” or discrete amounts when information is exchanged from one state to another. With quantum theory comes the possibility that everything in physics has a finite probability. Some probabilities associated with events are very small; still, the likelihood is non-zero.

Nuclear Engineering deals with nuclear radiation processes that occur based on a *probability*. For example: neutrons bombarding nuclear fuel can cause fission, liberating “prompt fission neutrons”... those prompt neutrons are emitted within 10^{-17} s of a fission event... and there is a certain probability of neutron-nucleus interaction, called a *cross section*.

Cross Sections

Cross sections (often referred to as “nuclear data”) can be stated in either *microscopic* or *macroscopic* form. The units of the cross section define *how* they are used; typically, nuclear interaction cross sections are denoted by the Greek symbol “sigma”, be it either σ or Σ . Some texts adopt the convention that the lower case sigma be used as microscopic, and the upper case as macroscopic. However, it is the units defining the cross section that really matter. Microscopic and macroscopic cross sections are defined as follows:

- A *microscopic cross section* is the effective “target area” presented by a nucleus, and are typically given in barns ($1 \text{ barn} = 1.0 \times 10^{-24} \text{ cm}^2$).

- A *macroscopic cross section* accounts for the atomic density of nuclei in the substance, and is the effective “target area” presented by a nucleus to an incident neutron *per unit volume*. If we use the convention that the lower case sigma be used as microscopic, and upper case as macroscopic, the macroscopic case is equal to $\Sigma = N\sigma$, and N is the atom density. The macroscopic cross-section is typically given in units of cm^{-1} .

In either case, the amount of area can be directly correlated to likelihood for interaction, in effect, a probability. Cross sections are used in some applications in this book, but the intent of this text is not to teach nuclear physics, or nuclear interactions, or reactor theory. The intent of material presented here is to cover the essential applied mathematics and physics topics needed for a new student in nuclear engineering to gain experience and confidence in a limited yet focused set of analytic and numeric problem solving skills.

The remainder of the text contains a number of topics as follows:

- Chapter 2 discusses probability and applicable laws of probability, focusing on those typically encountered in nuclear related subjects.
- Chapter 3 introduces numerical computations; it is only an introduction, and the material covered serves as a lead in for an advanced course in computational numerical analysis.
- Chapter 4 briefly covers complex numbers for completeness.
- Chapter 5 reviews solution methods for ordinary differential equations, and emphasizes skills needed for flexibility in solving general problems.
- Chapter 6 presents power series solution methods, and how these solutions are consistent with alternative methods.

- Chapter 7 presents the solution methods needed for variable differential equations.
- Chapter 8 covers vector, matrices, and linear systems solution methods with sufficient detail for accomplishing solution approaches.
- Chapter 9 discusses Gram-Schmidt orthogonalization and Fourier Series, leading up to solutions of partial differential equations.
- Chapter 10 presents “Applied Solution Methods” Part 1, and covers topics essential for preparation involving multiple dimensional applications.
- Chapter 11 presents “Applied Solution Methods” Part 2, and explores applications in ordinary and partial differential equations. Separation of Variables, superposition, and eigenfunction expansion are covered in the discussion.
- Chapter 12 briefly discusses numerical solution methods of Partial Differential Equations; this material also serves as a lead-in for an advanced course in computational numerical analysis.

A detailed set of applications problems are included in the Appendix.

The depth and breadth of topics presented in each chapter varies; the goal is that at the close of a course that covers the material presented in this text, a student in nuclear engineering should be ready to tackle the rigors of neutron transport theory, reactor physics, radiation interactions, radiation shielding, and other challenging topics in a nuclear engineering curriculum.