

UNIVERSITY OF BIRMINGHAM



Department of Metallurgy & Materials

Irradiation Damage Simulations of Platinum Group Metal Modified Austenitic Stainless Steels for Reactor Core Components

by Ben Palmer

A thesis submitted to the University of Birmingham for the degree of Doctor of Philosophy

Supervised by Dr Brian J Connolly and Dr Mark S D Read

**Irradiation Damage Simulations of Platinum Group Metal Modified Austenitic Stainless Steels
for Reactor Core Components**

by

Ben Palmer

Abstract

This work is split into two investigations. The first focuses on deriving the interatomic potentials required to model irradiated austenitic stainless steels doped with small amounts of Palladium or Ruthenium. The second centers on a Fortran program developed to predict the radioactivity of a sample irradiated by a proton beam.

Austenitic stainless steels are an important material, and it is used to construct core components within nuclear reactors. Sensitisation of these steels, by processes such as welding, depletes Chromium from grain boundaries removing the corrosion resistant Cr_2O_3 passive layer. Previous (experimental) work has shown that these steels, when doped with palladium or ruthenium, retain corrosion resistance at the grain boundary. This work investigates whether or not these PGMs deplete or saturate at the grain boundary while under irradiation using Molecular Dynamics (MD) simulations. Interatomic potentials have been derived for Fe-Pd and Fe-Ru as they were not available, and these are essential for the MD simulations of the irradiated grain boundaries. The effect of irradiation on Fe-Ru and Fe-Pd showed depletion/saturation/no effect.

In addition to simulations, it would also be desireable to investigate whether PGMs are depleted or saturated at the grain boundary by emulating neutron damage with a proton beam. The irradiated targets are expected to be radioactive due to the transmutation of target nucleons. A Fortran program, Activity, was written and developed to calculate how radioactive a target sample would be after proton beam irradiation, and from this it can be determined when it would be safe to handle.

Contents

1	Introduction	6
1.1	Nuclear Power in the UK	7
1.1.1	Generation 1 Reactors	7
1.1.2	Generation 2 Reactors	7
1.1.3	Generation 3 Reactors	7
1.2	An Approaching Energy Gap for the UK	7
1.2.1	Proposed Generation III+ Nuclear Power Plants	8
1.2.2	Generation IV Proposed Designs	9
1.3	Nuclear Fission and Neutron Spectra	11
1.4	Materials Developed for Nuclear Power	12
1.5	Austenitic Stainless Steel	13
1.5.1	Stainless Steels	13
1.5.2	Austenitic Stainless Steels in the Nuclear Industry	13
1.5.3	Issues Associated with Austenitic Stainless Steels	13
1.5.4	Sensitization of Austenitic Stainless Steels	13
1.5.5	Sensitization of Austenitic Stainless Steels	14
1.5.6	Intergranular Stress Corrosion Cracking	14
1.6	Irradiation Damage inside the Reactor Core	15
1.7	Doping with Palladium, Ruthenium and other Platinum Group Metals	16
1.7.1	Palladium and Ruthenium at the Grain Boundary	16
1.8	Doping Alloys with Palladium and Ruthenium	16
1.8.1	Cathodic Modification	16
1.8.2	Effect on Intergranular Stress Corrosion Cracking	16
1.8.3	Other Notable Corrosion Resistance Enhancing Alloys	16
1.9	Ion Irradiation	17
1.9.1	Emulating Neutron Radiation	17
1.9.2	Proton Activation	17
1.10	Scope and Objectives	18

2 Background: Radiation Effects and Transport	19
2.1 Radiation Types Relevant to This Work	20
2.1.1 Introduction	20
2.1.2 Protons and Ions	20
2.1.3 Neutrons	20
2.1.4 Electrons	21
2.1.5 High Energy Photons	21
2.2 Material Damage	23
2.3 Radioactive Byproducts of Proton and Neutron Irradiation and Associated Dangers	24
2.3.1 Radioactivity, Absorbed Dose, Dose Equivalent and Exposure	24
2.3.2 Effect of Radiation on Organics	27
2.3.3 Damage Types	27
2.3.4 Collision Event	27
3 Background: Proton Activation and Radioactive Decay	28
3.1 Proton Accelerators	29
3.1.1 Linac	29
3.1.2 Cyclotron	29
3.1.3 Synchrotron	29
3.1.4 Radio Frequency Quadrupole	29
3.2 Neutron Sources	30
3.2.1 High-Flux Neutron Reactors	30
3.2.2 Spallation	30
3.3 Source Review	32
3.3.1 Table	32
3.4 Ion Irradiation to Investigate Neutron Damage	33
3.4.1 Ion Irradiation at the University of Birmingham	33
3.4.2 Transmutation of Nuclei by Neutrons and Protons	33
3.4.3 Nuclear Reaction Cross Sections	34
3.4.4 Radioactive Decay	35
3.4.5 Bateman Equation for Radioactive Decay	35
3.4.6 Simulating Ion Irradiation with SRIM	36

4 Background: Interatomic Potential Fitting	37
4.1 Experiment, Modelling and Theory	38
4.1.1 Introduction	38
4.1.2 Simulating Materials on a Variety of Scales in Time and Space	38
4.2 Force Matching	39
4.3 Properties of Cubic Crystals	40
4.3.1 Introduction	40
4.3.2 Bulk Modulus	40
4.3.3 Equation of State	40
4.3.4 Strain	41
4.3.5 Stress	41
4.3.6 Voigt Notation	41
4.3.7 Elastic Constants	41
4.3.8 Calculating Elastic Constants for a Cubic Crystal	42
4.4 Properties of Orthorhombic Crystals	43
4.4.1 Introduction	43
4.4.2 Calculation of Elastic Constants	44
4.5 Magnetism	47
4.5.1 Brief History of Magnetism	47
4.5.2 Origin of Magnetism	47
4.5.3 Ferromagnetism and Antiferromagnetism	47
4.6 Solid State Physics	48
4.6.1 Schrodinger Equation	48
4.6.2 Bravais Lattice	48
4.6.3 Reciprocal Lattice	48
4.6.4 Periodic Potential	48
4.7 Density Functional Theory	49
4.7.1 Brief Overview of DFT	49
4.7.2 Time Independent Schrdinger Equation	49
4.7.3 Born-Oppenheimer	51
4.7.4 Hohenberg-Kohn Theorem	51
4.7.5 Born-Oppenheimer	51
4.7.6 pseudopotentials	51
4.8 DFT: Magnetism	52
4.8.1 Collinear	52
4.8.2 Non-Collinear	52

4.9	Classical Molecular Dynamics	53
4.9.1	Verlet Timestep	53
4.10	Kinetic Monte Carlo	54
4.11	Interatomic Potentials	55
4.11.1	Introduction	56
4.11.2	Morse and Lennard Jones Pair Potentials	57
4.11.3	Finnis Sinclair Potentials	57
4.11.4	Embedded Atom Method	57
4.11.5	Two Band Embedded Atom Method	57
4.11.6	ZBL Function	61
4.12	Fitting Interatomic Potentials	63
4.12.1	Analysing a Potential	63
4.13	Continuous Optimization	64
4.13.1	Introduction	64
4.13.2	Continuous and Discrete Optimization	64
4.13.3	Global and Local Optimization	64
4.13.4	Global Optimization Algorithms	64
4.13.5	Newton Gauss	67
4.13.6	Levenberg Marquardt	67
5	Methodology: Proton Activation and Radioactive Decay	68
5.0.1	Introduction	68
5.1	Fortran	69
5.2	Activation by Ion Irradiation	70
5.2.1	Laplace Transform	70
5.2.2	Constructing the Differential Equations	70
5.2.3	Numerical Inversion of the Laplace Transform	71
5.2.4	Analytic Solution by Partial Fraction Expansion	72
5.2.5	Preference: Analytic over Numeric	73
5.3	Computational Methods	74
5.4	SRIM	75
5.5	Activation by Neutron Irradiation	76
5.5.1	Introduction	76

6 Methodology: Interatomic Potential Fitting	77
6.0.1 Introduction	77
6.1 Math Techniques	78
6.1.1 Lagrange Polynomial Interpolation	78
6.1.2 Splines	79
6.1.3 Simulated Annealing	79
6.1.4 Levenberg-Marquardt Optimisation	79
6.2 Interatomic Potential Fitting	80
6.3 Density Functional Theory	81
6.3.1 Quantum Espresso	81
6.3.2 Crystal Structure	81
6.3.3 Pseudopotential Selection	81
6.3.4 Qeconverge Python Program	81
6.4 DFT Calculations: Iron and Palladium	83
6.5 High Performance Computing	84
6.5.1 Processing and Memory Requirements	84
6.5.2 HECToR and Archer	84
6.5.3 BlueBEAR	84
6.6 Reference Database	85
6.6.1 DFT Calibration	85
6.6.2 Atomic Configurations for DFT Calculations	85
6.7 Potential Analysis and Fitting Code	86
6.7.1 Tabulated Potentials and Interpolation	86
6.7.2 Simulated Annealing	86
6.7.3 Levenberg-Marquardt Optimisation	86
6.8 Calculating Crystal Properties	87
6.8.1 Introduction	87
6.8.2 Bulk Modulus	87
6.9 DL_POLY Contribution	88
6.9.1 Introduction	88
6.9.2 Modifying DL_POLY: 2BEAM	88
6.9.3 Mailshot Extract	88
6.10 Continuous Optimization	89
6.10.1 Implementation of Optimization Algorithms	89
6.10.2 LMA	89

7 Activity Code Development & Publication	90
7.1 Activation by Ion Irradiation	90
7.2 Computer Package Development	90
7.3 Convergence of Key DFT Settings and Testing	91
7.3.1 Introduction	91
7.3.2 DFT Crystal Property Calculations	91
8 Ab Initio Reference Database	93
8.1 Activation by Ion Irradiation	93
8.2 Computer Package Development	93
9 Interatomic Potential Fitting	94
9.1 Activation by Ion Irradiation	94
9.2 Computer Package Development	94
9.2.1 Cython	94
9.2.2 F2PY	94
9.2.3 OpenMP	94
9.2.4 Development	94
10 Molecular Dynamics	95
10.1 Activation by Ion Irradiation	95
10.2 Computer Package Development	95
11 Conclusions	96
11.1 Restating Objectives	96
11.1.1 Contributions of this Thesis While Answering the Original Question	96
12 Future Work	97
Appendices	98
A Activity Paper published in Computer Physics Communications	99
A.1 Paper available through CPC	99
A.2 Accompanying Manual	107
B Simulated Annealing vs Genetic Algorithm	141
B.1 Introduction	141
B.2 Algorithm Comparison	141
B.2.1 Genetic Algorithm, Pop 32 vs Simulated Annealing	141
B.2.2 Genetic Algorithm, Pop 64 vs Simulated Annealing	141
B.2.3 Genetic Algorithm, Pop 512 vs Simulated Annealing	141

List of Figures

1.1	Graph caption	7
1.2	Electricity in Millions of Tonnes of Oil Equivalent	8
1.3	Graph caption	9
1.4	Fission Spectra	11
2.1	Mn 56 Activity	25
2.2	Steel Irradiation Gamma Dose 2.5e7 Neutrons per cmsquared per second	25
2.3	High Flux Isotope Reactor Neutron Flux	26
2.4	Steel Irradiation Gamma Dose 1.0e14 Neutrons per cmsquared per second	26
2.5	...	27
3.1	A cross section of the HFIR [9]	30
3.2	Electricity in Millions of Tonnes of Oil Equivalent	33
3.3	An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.	35
3.4	One hundred simulated 13MeV proton energy loss curves in Fe simulated with SRIM [11]	36
4.1	Graph caption	46
4.2	Graph caption	58
4.3	Graph caption	59
5.1	An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.	70
5.2	Decay of Po-218: Analytic and Gaver-Stehfest Calculations [10]	74
5.3	Flow chart of major processes in the Activity code	74
B.1	Genetic Algorithm Population Size 32 vs Simulated Annealing	142
B.2	Genetic Algorithm Population Size 32 vs Simulated Annealing	142
B.3	Genetic Algorithm Population Size 32 vs Simulated Annealing	143
B.4	Genetic Algorithm Population Size 64 vs Simulated Annealing	143
B.5	Genetic Algorithm Population Size 64 vs Simulated Annealing	144
B.6	Genetic Algorithm Population Size 64 vs Simulated Annealing	144

B.7	Genetic Algorithm Population Size 512 vs Simulated Annealing	145
B.8	Genetic Algorithm Population Size 512 vs Simulated Annealing	145
B.9	Genetic Algorithm Population Size 512 vs Simulated Annealing	146

List of Tables

2.1	Neutron Categories by Energy Range [5]	20
2.2	DFT Settings	24
3.1	Nbnd settings	32
4.1	Useful Conversion Factors	40
4.2	Seven Crystal Classes	43
6.1	Predicted lattice parameters based on the density, atomic number and type of structure	81
7.1	DFT Settings	91
7.2	DFT Settings	92

Chapter 1

Introduction

Several generation III+ nuclear reactors have been proposed for construction in the UK and generation IV nuclear reactors being researched and developed. New materials are required to withstand the extreme conditions in and around the core of these reactors. Austenitic stainless steels have been an important structural material in the industry, and may continue to be so, providing the problem of Intergranular Stress Corrosion Cracking (IGSCC) can be addressed. Doping these steels with platinum group metals has been seen to reduce IGSCC, but the effects are unknown for these steels in a radiation field.

Notes: brief history of nuclear power discuss neutron spectra focussing on u-235 fission what materials are used in nuclear power - considerations (damage, transmutation, costs, other environmental factors) focus on austenitic stainless steels

1.1 Nuclear Power in the UK

Magnox type reactors were the first used in the United Kingdom. These reactors used natural Uranium as a fuel and were carefully designed to produce energy despite using an unenriched fuel. Graphite was used as a moderator, and the low neutron capture cross section of the Magnox cladding allowed th

1.1.1 Generation 1 Reactors

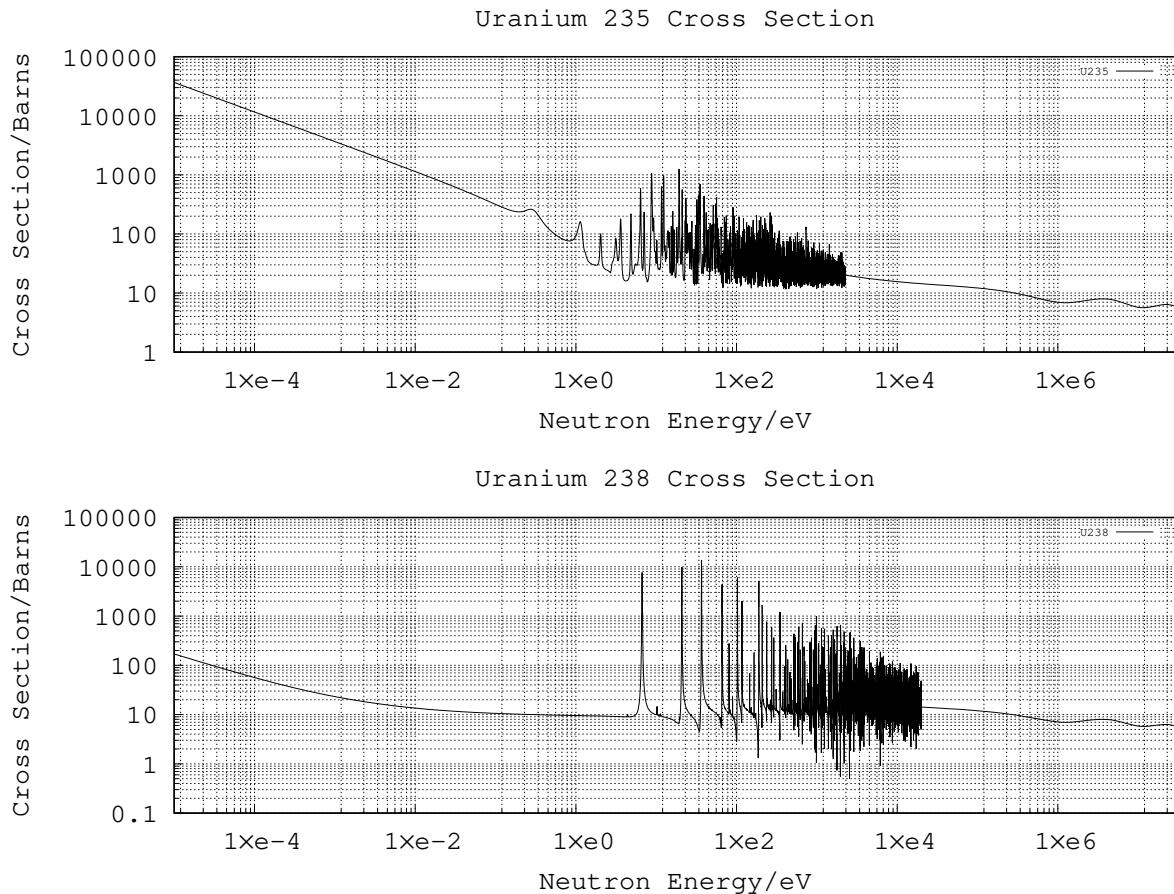


Figure 1.1: Graph caption

1.1.2 Generation 2 Reactors

1.1.3 Generation 3 Reactors

1.2 An Approaching Energy Gap for the UK

Since Calder Hall, the first commercial nuclear power plant, opened in 1956, the demand on electrical power generation in the UK has tripled. There is now a reliance on cheap and clean power from nuclear reactors as these provide a quarter of our electricity. There are sixteen reactors operational in the UK: the Magnox reactor at Wylfa and the fourteen AGR reactors are due to be decommissioned by 2023[1], and the remaining PWR reactor, Sizewell B, is expected to remain operational until 2035[1].

There is an obvious concern that within the next ten years the UK will lose a sizeable proportion of its electricity generation capabilities, however, there are proposals to remedy this. EDF have planned to build two new reactors

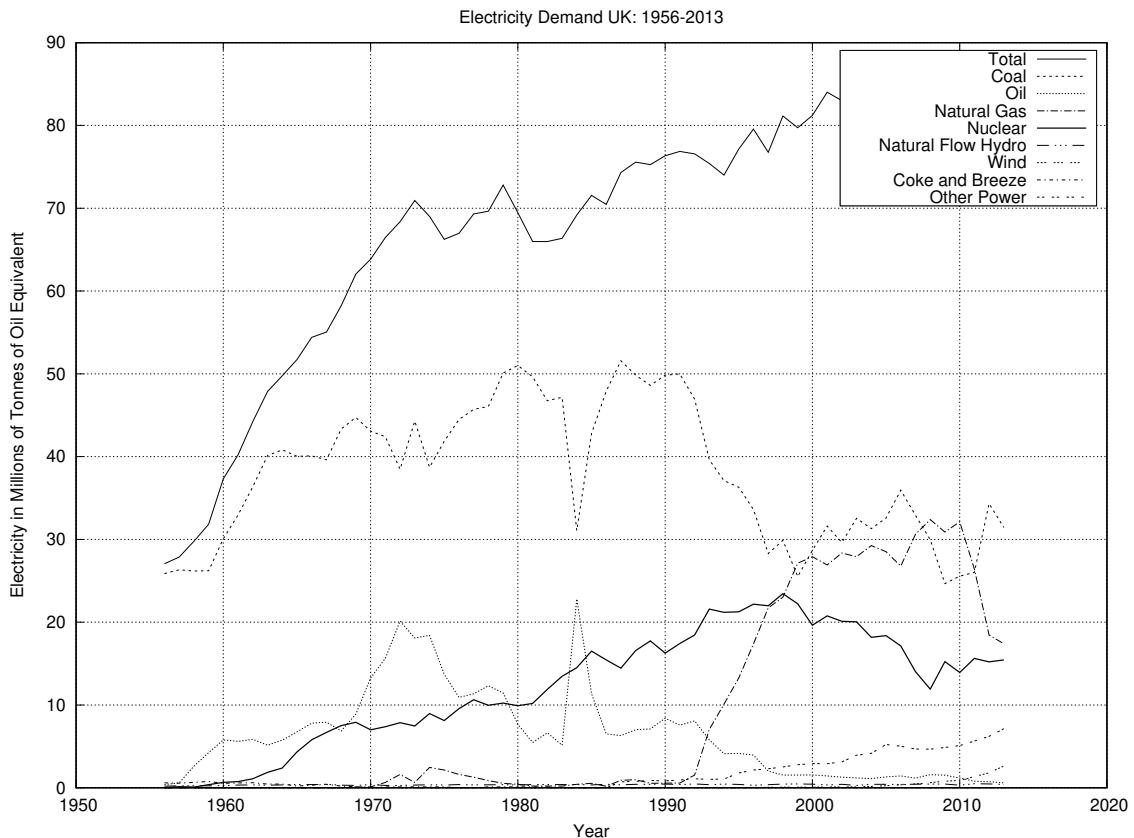


Figure 1.2: Electricity in Millions of Tonnes of Oil Equivalent

at the Hinkley Point site in Somerset. Hinkley Point C will contain two Areva NP designed Gen III+ EPR reactors. Many advanced materials, including a variety of types of Austenitic Stainless Steel, will be used in the construction of these reactors.

The physical stresses subjected to the Gen IV reactors will go beyond those that are currently being built. There will be higher radiation doses and faster, more damaging neutrons. Coolant temperatures will be higher to either give better thermodynamic efficiency or open the route to creating hydrogen directly as a fuel. Novel coolants such as lead and sodium, each with their own challenges to overcome, are also being considered.

1.2.1 Proposed Generation III+ Nuclear Power Plants

Although work has not started, at the time of writing, several sites have been acquired with the aim of building new nuclear power stations. There are five sites and three reactor designs[2]:

- Hinkley Point: two Areva EPRs (EDF Energy)
- Sizewell: two Areva EPRs (EDF Energy)
- Wylfa: 2-3 Hitachi ABWRs (Horizon Nuclear Power)
- Oldbury: 2-3 Hitachi ABWRs (Horizon Nuclear Power)
- Sellafield: 3 Westinghouse AP1000s (NuGeneration) ...

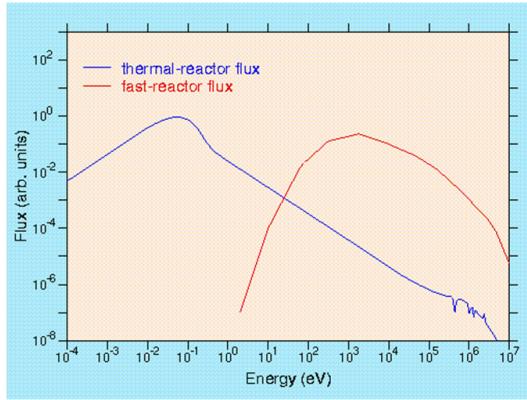


Figure 1.3: Graph caption

EPR

AP1000

1.2.2 Generation IV Proposed Designs

Super-Critical Water Reactors (SCWR)

Supercritical water exists above 374°C and 22.1MPa, and in this state water has a higher thermodynamic efficiency. The design of the nuclear power plant is also simplified as there is no phase change of the water, so a condenser is not needed. The SCWR is the only GEN IV reactor design that uses water as the coolant[1]. The economic benefits have already been seen in SCW fossil fuel power stations, and it is incorporated in GenIV water cooled fast and thermal reactors. The combination of supercritical water chemistry and irradiation damage must be considered, as well as higher temperature and pressure.

PWR 288-325°C 15.2 MPa[2]

BWR 278-287°C 7.1MPa[3]

Lead-cooled Fast Reactors

One example of the next generation reactor designs is ELSY: the European Lead Fast Reactor. It is a fast neutron reactor and this benefits from a lead coolant as lead has a low reaction cross section and the maximum energy lost per neutron-coolant atom collision is low. The Lead coolant will be at a temperature of 400°C to 480°C[4] and will weigh 9,000 tons. The challenge for engineers is to develop materials that can survive such extreme conditions, including the corrosiveness of the liquid lead and the high flux of fast neutrons. (5).

Gas Cooled Fast Reactors

Very high outlet temperatures can be achieved for GFRs, with the temperature of the gas coolant ranging from 490°C to 850°C (6) and this requires advanced materials that can withstand temperatures as high while under high energy neutron flux. Unlike LFRs and SCWRs, the coolant is inert, leaving engineers to overcome high temperatures and irradiation damage.

Experimental Fusion Reactors

Nuclear Fusion is a very attractive technology and could be the answer to all of our energy problems. Much work is being invested in developing this technology and the ITER (International Thermonuclear Experimental Reactor) has been designed to output more energy than is required to start the fusion reaction. The process of fusion combines two isotopes of hydrogen and leaves helium and fast neutrons. As neutrons have no charge, they can penetrate shielding causing damage as they lose energy through nuclear interactions. Any atoms they interact with have a chance to capture the neutron and become unstable.



The fast neutron spectrum for fission reactors ranges from a few eV to a few MeV, whereas the neutrons in a fusion reaction have 2-3 times more energy than the most energetic neutrons from fast fission. Engineers must develop materials to construct components that will be resilient to this damage, while having a low reaction cross section and being able to withstand other extreme conditions within the reactor.

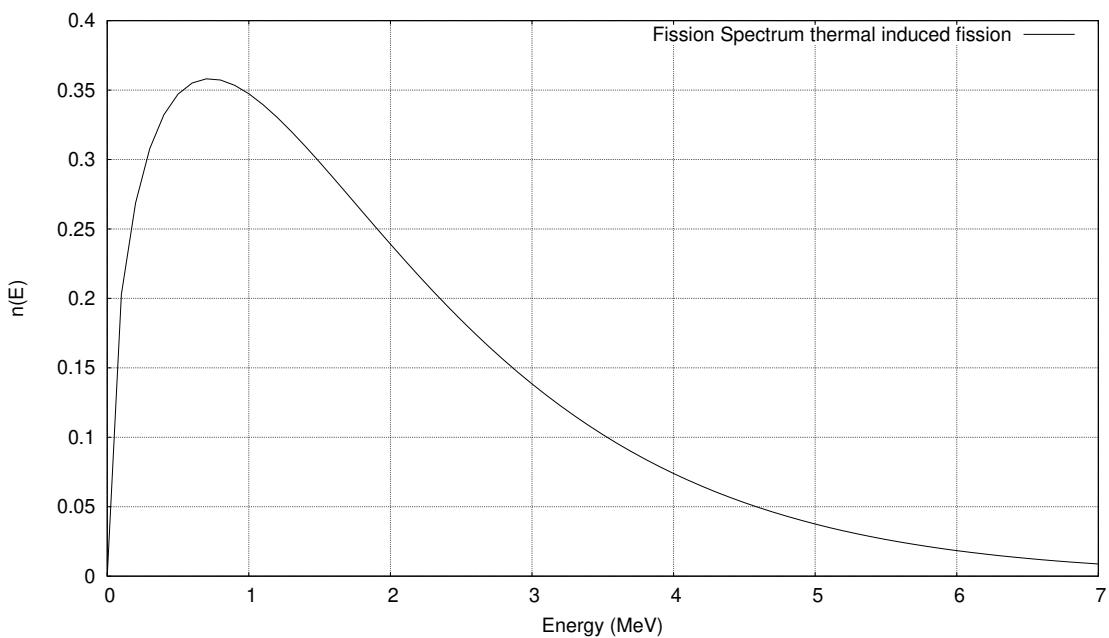


Figure 1.4: Fission Spectra

1.3 Nuclear Fission and Neutron Spectra

Radiation damage in a reactor is caused by several projectiles. Fission fragments are heavy, contain charged particles and lose energy close to their source. Gamma rays ionise may excite atoms and electrons, but do not have the momentum to knock atoms out of place. Fast neutrons, however, may impart a great deal of momentum to a target atom and, as they are neutral, travel much further into a material than fission fragments.

One fuel source for many of the Gen III+ and Gen IV reactors is Uranium-235, whether as enriched Uranium or otherwise. The neutron/s released by the fission of Uranium-235 atoms have a spectra of energy which may roughly be split into four categories: cold (below 0.025eV), thermal (0.025eV), slow and intermediate (above 0.025eV and below 1MeV) and fast (1MeV and above).

Thermal and slow/intermediate neutrons cause damage in their own particular way, as they are captured by and transmute the atoms within the target material. Fast neutrons

reference literature neutron damage

Neutrons

1.4 Materials Developed for Nuclear Power

With temperatures of several hundred degrees or more, pressures exceeding 150 atmospheres and a high levels of irradiation, 1.0×10^{11} neutrons per " cm^2 " per second, the interior of a nuclear reactor is a hostile and damaging environment. The choice of material is further complicated due to a number of factors.

- Isotopes with a high neutron capture cross section that poison the reactor (Xenon-135 and Samarium-149)
- Limiting creation of hazardous radioactive material in the structural material of the reactor (Cobalt-60)
- Absorbtion of neutrons by structural material

The environment within the reactor is due to become more hostile as future reactor designs are developed. A number of reactors will operate at higher temperatures (850 degrees or more) while others are being designed to replace water as a coolant with liquid metal salts.

1.5 Austenitic Stainless Steel

1.5.1 Stainless Steels

Stainless steel is a relatively new material, having first been developed and refined from the 1800s to the early 1900s, then being defined as a steel with at least 10.5

Passive Film Protection of Chromium

Chromium improves the resistance of stainless steel by forming a passive oxide layer Cr₂O₃. Chromium atoms and chromium oxide are similar in size, so they pack well into the crystal structure of the steel. The resistance of austenitic stainless steels can be improved further by added other elements, and removing some impurities (or elements found in other standard compositions). Read: nature of passive films McBee, Kruger

Ferritic Stainless Steel

Ferritic stainless steels have a crystal structure similar to pure Iron at room temperature which is bcc. These steels are magnetic and may be hardened by cold working. They are less corrosion resistant than austenitic stainless steels. Two common examples of this grade of steel are ASME (American Society of Mechanical Engineers) codes 405 and 430.

Martensitic Stainless Steel

Unlike Austenitic and Ferritic, Martensitic stainless steels can be heat treated to harden the steel. These steels are magnetic and have a FCT crystal structure. They contain more than 10.5

Duplex Stainless Steel

Austenitic Stainless Steel

Austenite is a FCC allotrope of Iron, and austenitic stainless steels are useful in many applications, including a structural material for nuclear plant components, due to their resistance to corrosion. In addition to 11 wt

Two examples of such steels are ASME codes 304 and 316. Both have a high Chromium content, in the region of 18-20

1.5.2 Austenitic Stainless Steels in the Nuclear Industry

Components of the Sizewell B PWR were constructed using stainless steels. In particular, all the major parts of the reactor vessel were made from stainless steel, with the reactor coolant piping loop being made from austenitic stainless steel. The majority of the components of the reactor coolant pump are also made from austenitic stainless steel as well as cladding inside the carbon steel pressure vessel. [sizewellbdescription]

[eprreactoroverview]

The Westinghouse AP1000 will also heavily rely on these steels. [ap1000preconstruction]

Low cobalt content

1.5.3 Issues Associated with Austenitic Stainless Steels

1.5.4 Sensitization of Austenitic Stainless Steels

When steels with Chromium content are heated, during processes such as welding, the metal undergoes Chromium sensitization. On heating to temperatures between 600-700°C for many hours Chromium carbides, of the form M₂₃C₆, are created, depleting Chromium at the grain boundary.

Steel held at elevated temperatures Chromium carbides precipitate at grain boundary Depletes the grain boundary of chromium Models of Tedmon et al, Fullman or Stawstrom and Hillert Difficult to model, depends on grain structure, intergranular carbide spacing

1.5.5 Sensitization of Austenitic Stainless Steels

The Chromium in stainless steel only gives the steel its corrosion resistant properties if the content does not drop locally below the threshold percentage required. When austenitic stainless steels are heated, chromium reacts with carbon at the grain boundary to form chromium carbide precipitates. This, coupled with the slow diffusion rate of chromium from within the grain to the surface, removes the protection from the grain boundary. This process is sensitization, and is a problem when welding steels ???.

1.5.6 Intergranular Stress Corrosion Cracking

1.6 Irradiation Damage inside the Reactor Core

1.7 Doping with Palladium, Ruthenium and other Platinum Group Metals

1.7.1 Palladium and Ruthenium at the Grain Boundary

1.8 Doping Alloys with Palladium and Ruthenium

The corrosion resistant benefits of small amounts of palladium (4) and ruthenium (5) have been investigated. Due to the high cost of both palladium and ruthenium, there has been much interest in finding the optimum percentage of both. Ru and Pd doped stainless steels would be worth the additional cost if their corrosive resistant properties are maintained at the grain boundary. In particular, ruthenium has been shown to be beneficial to stainless steels where chloride containing solutions are concerned (5). There have been a number of proposed mechanisms of how PGMs enhance the corrosion resistance of steel.

1.8.1 Cathodic Modification

Cathodic modification is one method that has been known for the last century (6), and the inhibition of anodic dissolution of the stainless steel has also been studied (7). The layer of PGM adatoms, atoms on the crystal surface, block anodic sites within the crystal which stops corrosion attack local to the adatoms (1).

1.8.2 Effect on Intergranular Stress Corrosion Cracking

1.8.3 Other Notable Corrosion Resistance Enhancing Alloys

Molybednum has been added to these steels to improve the resistance of pitting corrosion (4), and it has been shown to improve resistance against chloride containing solutions. Enhancing Resistance to Corrosion

1.9 Ion Irradiation

1.9.1 Emulating Neutron Radiation

reference literature neutron damage

Neutrons

1.9.2 Proton Activation

A major side effect from the process of nuclear fission is the creation of both radioactive fission fragments and radioactive isotopes within the components and structural material of the reactor. Low energy Protons are not captured as low energy Neutrons would be, due to the opposing force between the proton and nucleus of the target atom. Once the proton energies exceed a few MeV, they have sufficient energy to transmute target nuclei.

Evaluated Nuclear Data Files

Engineers and Scientists working for or researching in the nuclear industry need accurate data for a wide range of behaviours and properties of isotopes. There is no magic formula to return the requested data for a given isotope, and this is why there is a need for ENDF files. Experimental data

PADF

The Proton Activation Data File was released in 2007 and contained nuclear reaction data for 2355 target nuclei, ranging from Magnesium (12) to Radon (86) with proton energies up to 150MeV.

TALYS

TALYS is a computer code, written for Linux and Unix systems, that is used to predict and analyse nuclear reactions. It is also used as a tool to generate nuclear data.

TENDL

TENDL is a collection of files, each in the ENDF format, of nuclear reaction data generated by the TALYS code.

1.10 Scope and Objectives

Chapter 2

Background: Radiation Effects and Transport

Chapter Summary

2.1 Radiation Types Relevant to This Work

2.1.1 Introduction

There are three types of radiation that are useful to discuss in this work, and two of these are of particular interest: Neutrons, Ions and Gammas.

2.1.2 Protons and Ions

Charged particles interact with matter through the Coulomb interaction. As a charged particle passes through matter, it may interact with both the nucleus and electrons of an atom. A sufficiently energetic ion may lose kinetic energy to electrons by either raising the electrons to a higher energy levels in the atoms, or by removing electrons from atoms altogether, ionizing atoms.

Ions may also lose kinetic energy to the nucleus of an atom through elastic scattering and, where the atom is in a crystal structure, through knocking atoms in the material out of their lattice positions.

Knock on atoms and electrons with enough kinetic energy that have been removed from atoms (delta rays), continue the irradiation of the material while they have the kinetic energy available to do so.

Ion Activation

A large proportion of the energy of a charged projectile is lost to the electrons of the target material. There is a chance, depending on the energy and type of charged projectile, and the cross section of the target nucleus, that the charged particle will overcome the coulomb potential and be captured by the nucleus.

This may result in a stable nucleus or an unstable nucleus. If it is unstable, there is a probability that it will decay releasing energy in the form of photons, nucleons or electrons. The initial ion irradiation creates sources of futher irradiation within the target material.

2.1.3 Neutrons

Neutrons interact with matter differently to that of protons, ions and other atoms, as the Neutron has no overall charge. Neutrons do have a magnetic moment and experience a weak interaction with electrons, but the dominant interaction is between Neutrons and the Nucleus. There are different methods in which Neutrons interact and these are determined by the kinetic energy, velocity and wavelength of the Neutron.

Name	Energy Range	Velocity/ms ⁻¹	Wavelength Ang
Cold	0-0.025 eV	$0.0 - 2.2 \times 10^3$	> 1.8
Thermal	0.025 eV	2.2×10^3	1.8
Epithermal	0.025-0.4 eV	$2.2 \times 10^3 - 8.8 \times 10^3$	0.5-1.8
Cadmium	0.4-0.6 eV	$8.8 \times 10^3 - 1.1 \times 10^4$	0.4-0.5
Epicadmium	0.6-1.0 eV	$1.1 \times 10^4 - 1.4 \times 10^4$	0.3-0.4
Slow	1-10 eV	$1.4 \times 10^4 - 4.4 \times 10^4$	0.09-0.3
Resonance	10-300 eV	$4.4 \times 10^4 - 2.4 \times 10^5$	0.02-0.09
Intermediate	300 eV - 1 MeV	2.4×10^5	$2.9 \times 10^{-4} - 0.02$
Fast	1-20 MeV	$1.4 \times 10^7 - 6.1 \times 10^7$	$6.5 \times 10^{-5} - 2.9 \times 10^{-4}$
Relativistic	> 20 MeV	$> 6.1 \times 10^7$	$< 6.5 \times 10^{-5}$

Table 2.1: Neutron Categories by Energy Range [5]

Neutron-Neutron

Neutron-Proton

Neutron-Electron

2.1.4 Electrons

Delta Electrons and Delta Rays

2.1.5 High Energy Photons

The electromagnetic spectrum classifies photons based on their energy and/or source, but visible light, x-rays, gamma rays and so on are all the same elementary 'particle'. During the early years of Quantum Mechanics, the relationship between the energy and wavelength of a photon was discovered: the Planck-Einstein relation.

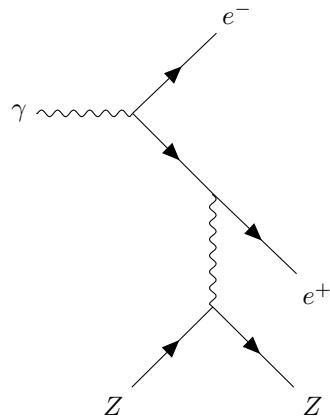
$$E = hf \quad (2.1)$$

Pair Production

The energy of photons in the database used in this work ranges from 1keV up to almost 10MeV. There are several ways high energy photons will interact with the atoms of a target material. The rest mass of an electron is 511keV. If the photon energy is greater than 1.02MeV, i.e. there is at least enough energy to create an electron and positron, there is a chance that the photon will create an electron-proton pair.

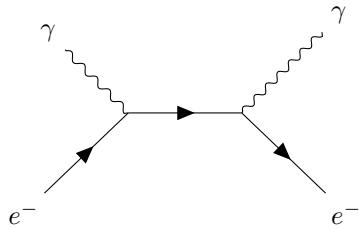
$$hf = (m_e + m_p)c^2 + T_e + T_p \quad (2.2)$$

The creation conserves energy and mass, with excess energy carried away as the kinetic energy of the particle pair. The charge before the creation is zero, as the photon is neutral, and the charge after is also zero, with the -1 of the electron and +1 of the positron cancelling out. Angular momentum is also conserved; the photon is a spin 1 Boson and, as electrons and positrons are Leptons they have half integer spin, adding up to 1. Finally, momentum is not conserved in a vacuum, and this is why pair production occurs in the coulomb field of a nucleus. The nucleus carries away excess momentum, fulfilling this conservation law.

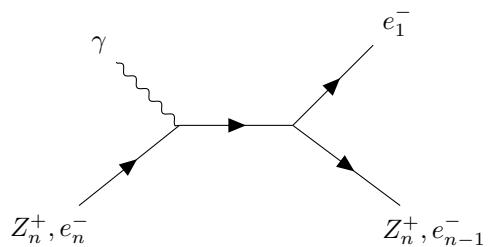


Compton Scattering

An incident photon, with enough energy, may interact with the electron of an atom, and it transfers enough kinetic energy to eject the electron from the atom. A lower energy photon is also created that carries away the remainder of the energy, but also linear momentum, as both energy and momentum must be conserved.

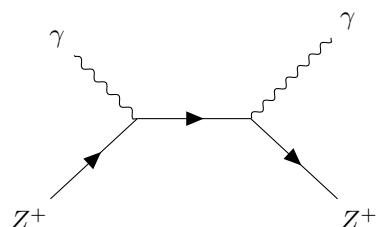


Photoelectric Effect



Coherent Scattering

Lower energy photons



2.2 Material Damage

Gamma and beta radiation damage are important to consider when reviewing biological damage, but the large proportion of damage caused to steel components will be caused by high energy protons, neutrons and their knock on atoms.

2.3 Radioactive Byproducts of Proton and Neutron Irradiation and Associated Dangers

2.3.1 Radioactivity, Absorbed Dose, Dose Equivalent and Exposure

There is a distinction between how radioactive a source is and how much radiation is absorbed by an organic due to that source. There are different measures of radioactivity and it's effects, and they may be summarised very briefly as follows:

- Radioactivity of a source - decay events per second (Bq) (also measured in Curie, 3.7×10^{10} decays per second)
- Absorbed dose of any target - energy absorbed per unit mass (Gy) (equivalent of Joules per Kilo)
- Equivalent dose absorbed by tissue/organ - type of radiation is important (Si)
- Effective dose on tissue/organ - how the equivalent dose affects the specific tissue/organ (Si)

One joule of ionising radiation absorbed by a person may not seem a lot of energy, but the average annual doses are so small they are measured in millisieverts per year. A dose of just 5 Sieverts is enough to kill 50

Exposure	Dose (mSv)
Dental X-ray	0.005
Chest X-ray	0.02
UK average annual dose	2.7
Whole body CT scan	10
Nuclear industry employee annual exposure	20
Acute radiation effects	1,000
50	

Table 2.2: DFT Settings

In an MCNP5 simulated neutron flux environment, with a thermal flux of $2.5 \times 10^7 n/cm^2 s$ a stainless steel cylinder, 20mm diameter and 70mm irradiated for 10 hours[6]. After 10 minutes cooling, it was calculated to have an activity of over $1.0e7$ Bq, predominantly due to the creation of Mn⁵⁶.

The activity in this work seems slightly. The maximum cross section of Mn⁵⁵ (n, γ) Mn⁵⁶ is 2.0 barns and the estimated reaction rate at this flux is just over $2.0e6$. As the half life of Mn⁵⁵ is short compared to the irradiation time (2.5 hours) the maximum number of Mn⁵⁵ atoms are created, with the creation rate equalling the decay rate.

A simple neutron activation code was developed, which will be discussed in more detail in this work, and the predicted Mn⁵⁵ activity vs time was calculated.

The overall gamma dose, for 1 minute of exposure to an 80kg person standing 2m from the source was also calculated.

The flux in the within the flux trap of the HFIR is much higher, with high energy neutrons having a flux of $1.0e14$ neutrons per cmsquared per second.

A second calculation was performed for this increased flux, also irradiated over 10 hours. The overall dose per minute was much higher, between 40 and 50 milligrey per minute over the first hour of cooling.

Activity vs Time Mn 56

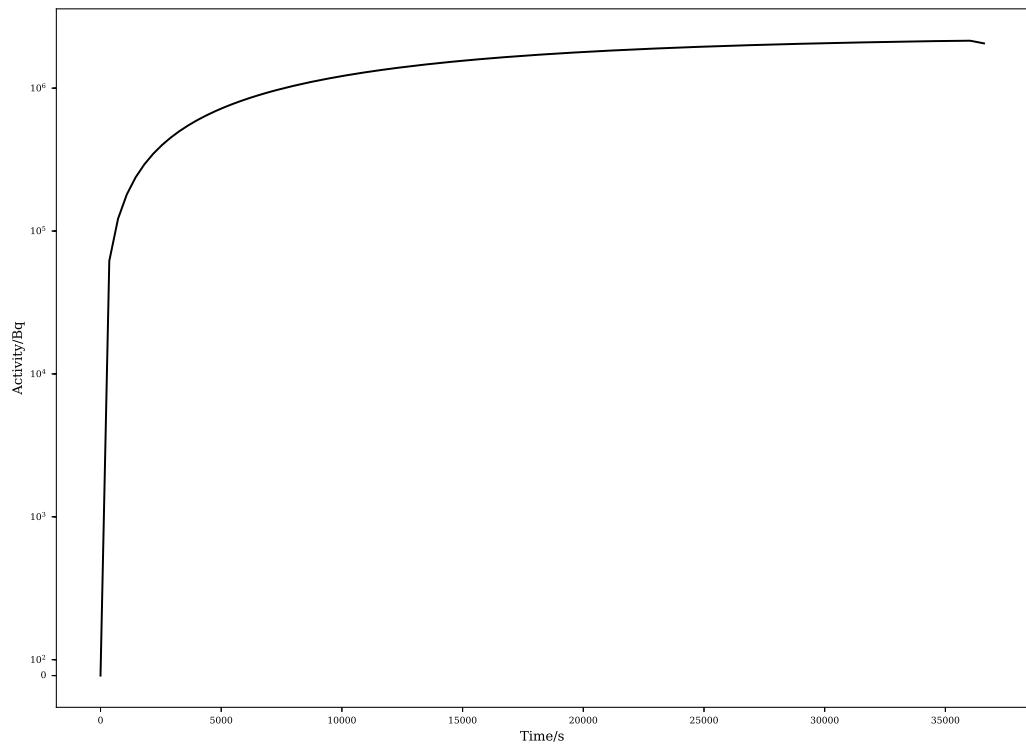


Figure 2.1: Mn 56 Activity

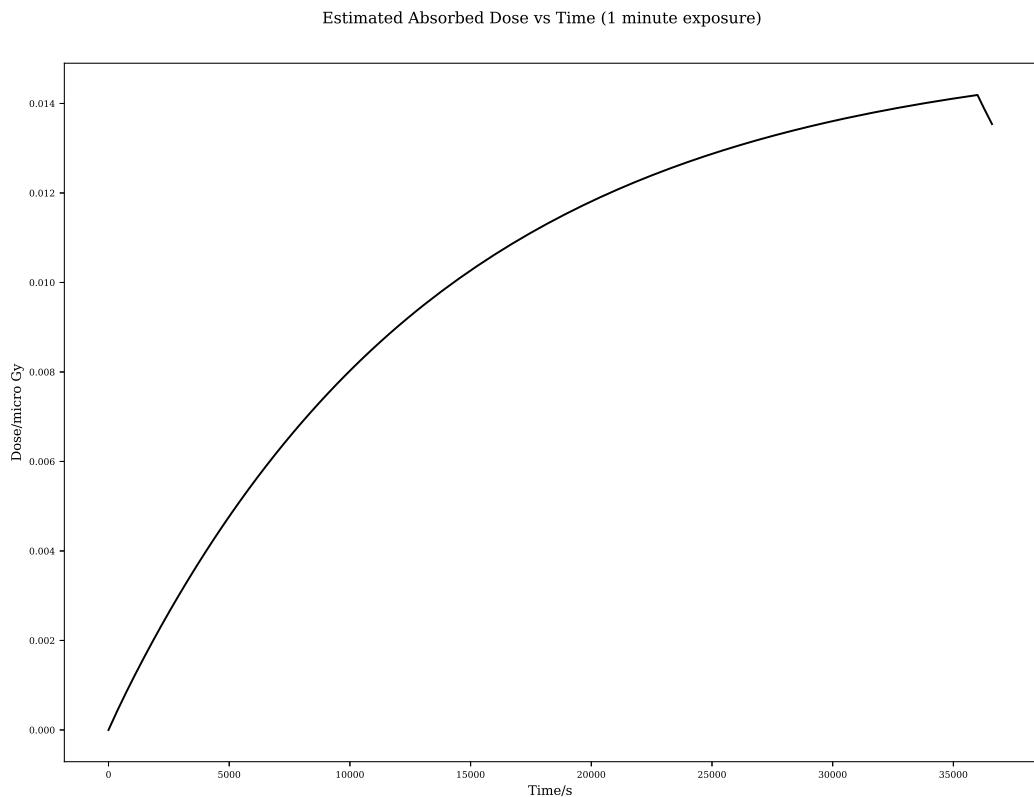


Figure 2.2: Steel Irradiation Gamma Dose $2.5e7$ Neutrons per cmsquared per second

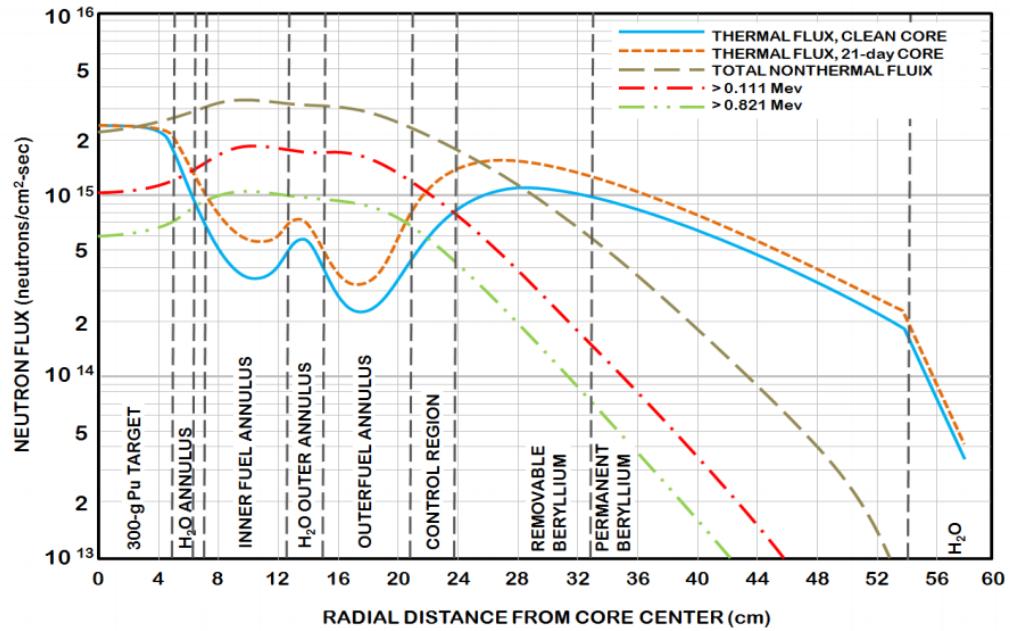


Figure 9: Neutron Flux at the MidPlane of HFIR (at 85MW)

Figure 2.3: High Flux Isotope Reactor Neutron Flux

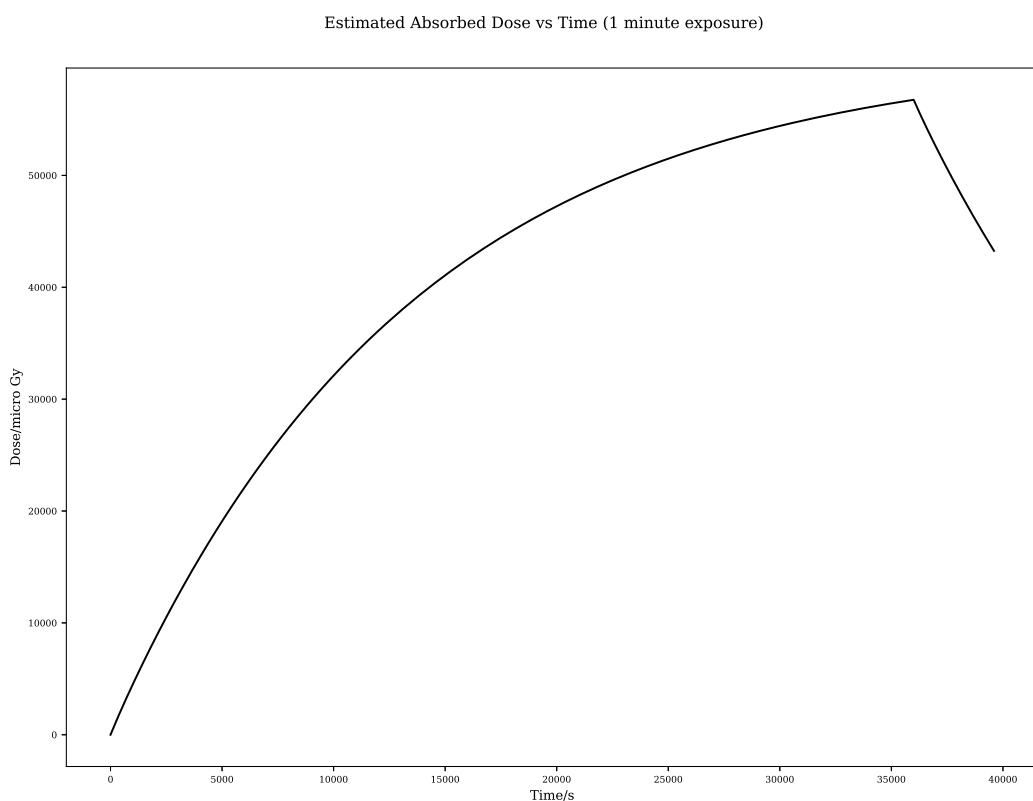


Figure 2.4: Steel Irradiation Gamma Dose $1.0e14$ Neutrons per cmsquared per second

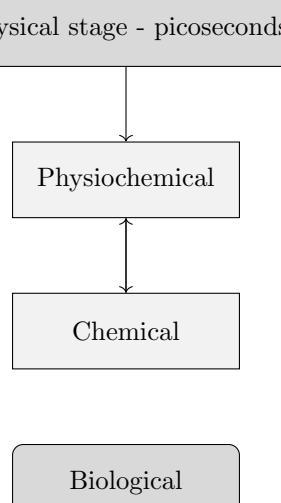


Figure 2.5

2.3.2 Effect of Radiation on Organics

Life evolved on Earth from single cell to multicelled organisms such as humans. Prokaryotic cells are single celled organisms that do not have a nucleus, an example being bacteria. Our cells are eukarytic cells, and these have a nucleus that contains the genetic information. Millions of cells die in our body every second, and our body replaces these by replicating living cells. During the replication stage, the DNA within the nucleus is replicated.

DNA is a polymer that is carefully constructed from six components. Stretched out, it is approximately 2m in length and approximately 25 angstrom wide, and it is neatly contained within the cell nucleus which is on average just 6 microns across. The structure of DNA is the well known double helix. The sides of the helix are made from alternating sugars and phosphates, while the "rungs" of the ladder like structure are pairs of either thymine and adenine or cytosine and guanine. These pairs are covalently bonded to the sugar-phosphate sides, and by hydrogen bonds to each other.

Mitosis is the part of the cell cycle where the DNA within the nucleus is replicated and the cell splits into two. There are repair mechanisms, but if the DNA beyond repair, the cell may die through apoptosis, or it may replicate in an uncontrolled way leading to cancer.

2.3.3 Damage Types

Direct damage is caused when ionising radiation alters or destroys sections of DNA through a direct collision, for example a neutron colliding with and knocking out an atom in the DNA strand.

Our cells are predominantly water, so there is a higher probability of the radiation colliding with water molecules than DNA directly. The chemistry of the contents of the cell is changed, for example by the creation of free radicals, and these lead to damage of DNA as an indirect result of radiation passing through the cell.

2.3.4 Collision Event

The damage event in a material such as Steel takes place in stages, and this is similar to the way the event is handled for biological structures.

Chapter 3

Background: Proton Activation and Radioactive Decay

Chapter Summary

3.1 Proton Accelerators

3.1.1 Linac

Since the development of the first linacs (linear accelerators) in the 1940s, their modern day versions have become some of the most powerful accelerators in the world. The longest of linac, SLAC (), is 3.2km in length and it accelerates electrons and positrons at energies of up to 50GeV. Several linacs for protons include the 800MeV linac component of the ISIS neutron source in Oxfordshire, and the 800MeV linac used by the Spallation Neutron Source at Oak Ridge National Laboratory.

The accelerator is constructed of several tubes, connected alternately to opposite terminals of a high frequency alternating current supply. As protons enter the first tube, a negative voltage is applied. As the protons reach the gap between the first and second tube, the polarity is reversed. The positive charge that is now applied to the first tube pushes the protons forward as the negative charge on the second tube pulls the protons forward. This process is repeated along the length of the accelerator, with the sections increasing in length due to the increase in velocity of the protons.

3.1.2 Cyclotron

Cyclotrons are reasonably compact and cost effective. The largest current cyclotron, TRIUMF, is located in Canada and is able to output protons with energies over 500MeV. It is large, weighing 170 tons, in comparison to the University of Birmingham cyclotron,

3.1.3 Synchrotron

Two of the most well known accelerators are Synchrotrons: the Large Hadron Collider at CERN, and the now retired Tevatron at Fermilab.

3.1.4 Radio Frequency Quadrupole

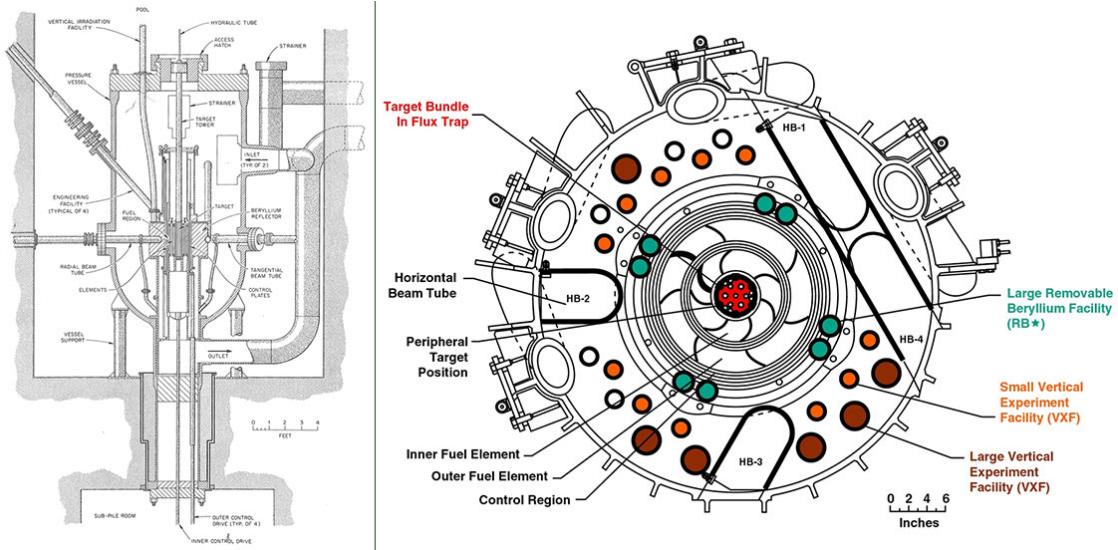


Figure 3.1: A cross section of the HFIR [9]

3.2 Neutron Sources

3.2.1 High-Flux Neutron Reactors

There are 250 or so research reactors in 55 countries [7], and a number of these are used for materials research. In the UK, there is only one remaining research reactor, and this is the Neptune pool type reactor at Rolls Royce[8]. In Cadarache, France, the Jules Horowitz reactor is under construction and this is being built specifically as a materials testing reactor. It will be crucial in researching new materials for use in upcomming Gen IV nuclear power stations [7].

High Flux Isotope Reactor, Oak Ridge

The High Flux Isotope Reactor, at the Oak Ridge National Laboratory in America, is an 85MW research reactor that provides testing space within the reactor as well as a number of neutron beam lines. The reactor uses highly enriched Uranium as its fuel source and is scheduled to operate at 100

NIST Center for Neutron Research

The NBS (National Bureau of Standards) Reactor was designed as a research reactor with a power output of 40MW and a neutron flux of approximately 1.0×10^{15} neutrons $cm^{-2}s^{-1}$. In 2006 it was decided to extend the facility and add beam lines

3.2.2 Spallation

Neutrons cause fission in nuclear reactors, but spallation sources require protons and high mass targets. The energy of the protons required is a magnitude greater than that of the Scanditronix MC-40 Cyclotron at the University of Birmingham, with spallation source accelerators having a range from 500MeV to over 1GeV.

ISIS

Ions are attracted from a plasma source and passed through to a radio frequency quadrupole (RFQ) accelerator. The RFQ supplies 665 KeV protons in batches to the linac every 4.94 ns.

3.3 Source Review

3.3.1 Table

Table 3.1: Nbnd settings

Source	Cost	Projectile	Flux/Current	Energy
Source	Cost	Projectile	Flux/Current	Energy

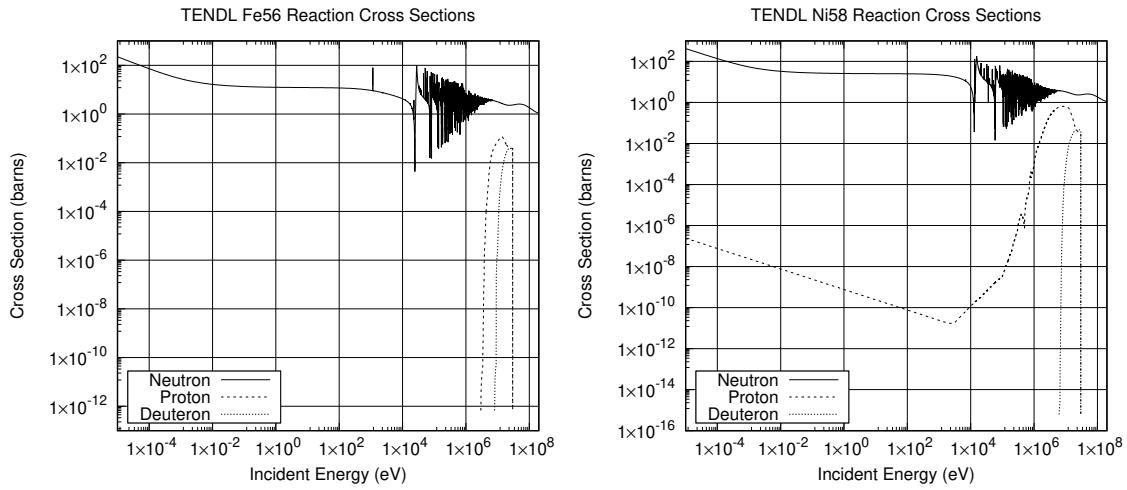


Figure 3.2: Electricity in Millions of Tonnes of Oil Equivalent

3.4 Ion Irradiation to Investigate Neutron Damage

The neutron energy spectra from Uranium-235 neutrons is less than n

3.4.1 Ion Irradiation at the University of Birmingham

The Scanditronix MC-40 Cyclotron is used at the University of Birmingham to create a beam of protons or other light ions. The energies of these ions are typically between 10 MeV and 60 MeV with beam currents ranging up to 50 microamps (3.1×10^{14} protons per second). Target materials are irradiated by this cyclotron for a number of reasons, including purposely creating radioactive isotopes for the nearby Queen Elizabeth Hospital, investigating ion irradiation damage and emulating neutron irradiation.

The Cyclotron is usually used to create radioactive isotopes for medical use, but an additional beam line has been devoted to material science investigations into radiation damage. While the creation of radioactive isotopes is desired in some cases, material being tested for radiation damage should preferably have low levels of radioactivity.

It is expensive to arrange the irradiation of target materials by high energy neutrons sources, whereas it is relatively inexpensive to irradiate using an ion beam on the MC-40 Cyclotron. The energies can be controlled, and a set dose at a single energy, or a range of energies, can be precisely deposited into the target material.

The Activity code discussed here was developed to calculate the activity of a target material irradiated by a proton beam. It has been developed in Fortran and uses data from the TENDL-2013 proton cross section database, SRIM ion transport code and NDS radioactive decay database.

3.4.2 Transmutation of Nuclei by Neutrons and Protons

Neutron Activation

The fission of Uranium-235 atoms results in neutrons with a varied spectrum of energies. The neutrons will bounce around inside the reactor losing energy quickly to light atoms within

Proton Activation

Considering a simplified nuclear potential well, energetic protons approaching a nucleus may overcome the Coulomb potential barrier. They are captured by the nucleus and held within the potential well by the strong nuclear force. This process may leave the nucleus in an excited and unstable state, depending on the input energy of the proton and configuration of nucleons. The process is probabilistic, and the average chance of a reaction (the microscopic cross section) may be measured as a function of the projectile, projectile energy and target, either experimentally or by optical model potential calculations. The reaction rate is calculated from the microscopic cross section using the following equation:

$$R = \frac{J}{e} n_t \sigma \cdot 10^{-28} \delta t \quad (3.1)$$

- R Reaction Rate (reactions per second)
- J Beam current (A)
- n_t Number density of target (atoms per cubic metre)
- σ Microscopic reaction cross section (barns)
- e Elementary charge (1.602177E-19C)
- δT Target thickness (m)

3.4.3 Nuclear Reaction Cross Sections

Reaction Cross Sections

TENDL Data Files

The cross section data for protons and neutrons is available to download in ENDF format files. The data files used by this work are TENDL data files, and these are created by a combination of different nuclear models and experimental data. The TALYS nuclear reaction simulation code provides the calculated data.

The nuclear reaction files are rather large, and they all follow, reasonably well, a standard format.

Each reaction file is itself (confusingly, to begin with) split into multiple "files".

TALYS Models

Relative Amounts of Isotopes

$$p_{m,k} = 100 \times \frac{p_{n,k} \times A_k}{\sum_{i=1,N} (p_{n,i} \times A_i)} \quad (3.2)$$

$$p_{n,k} = 100 \times \frac{\frac{p_{m,k}}{A_k}}{\sum_{i=1,N} (\frac{p_{m,i}}{A_i})} \quad (3.3)$$

3.4.4 Radioactive Decay

Radioactive decay is the random change in nucleons or energy state of an unstable nucleus. It is impossible to predict when a single nucleus will decay, but the decay of a collection of nuclei is statistical in nature. The radioactivity and number of unstable nuclei at time t can be predicted using the decay constant, λ , for the radioactive isotope. This constant is defined as follows:

$$\lambda = -\frac{N'(t)}{N(t)} \quad (3.4)$$

The number of radioactive nuclei $N(t)$ at time t is given by the following equation, where $N(0)$ is the starting number of nuclei:

$$N(t) = N(0) \exp(-t\lambda) \quad (3.5)$$

The activity $A(t)$ of the radioactive nuclei is predicted at time t by using the following equations, where $N'(t)$ is the change in amount of nuclei with respect to time:

$$A(t) = -N'(t) = \lambda N(t) \quad (3.6)$$

$$A(t) = \lambda N(0) \exp(-t\lambda) \quad (3.7)$$

3.4.5 Bateman Equation for Radioactive Decay

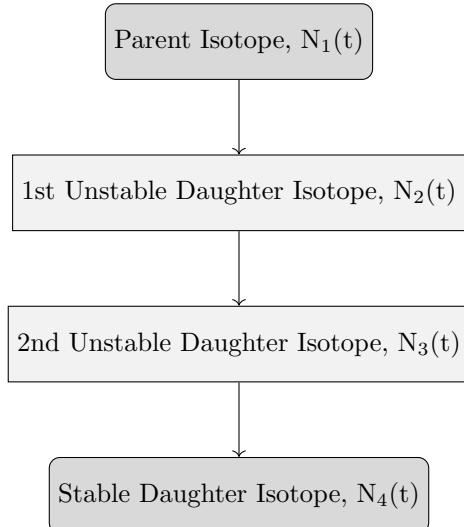


Figure 3.3: An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.

The English mathematician Harry Bateman derived an equation (3.8) to calculate the amount of each isotope in a decay chain, illustrated in Figure 3.3, at time t .

$$N_n(t) = \sum_{i=1}^{i=n} \left(\left(\prod_{j=i}^{j=n-1} \lambda_{(ij+1)} \right) \sum_{j=i}^{j=n} \left(\frac{N_{i0} \exp(-\lambda_j t)}{\prod_{p=i, p \neq j}^{p=n} (\lambda_p - \lambda_j)} \right) \right) \quad (3.8)$$

When a radioactive isotope decays, there may be more than one mode of decay, and this leads to branching factors. Pb-214 only decays via beta decay to Bi-214, giving a branching factor of 1.0, whereas Bi-214 has a

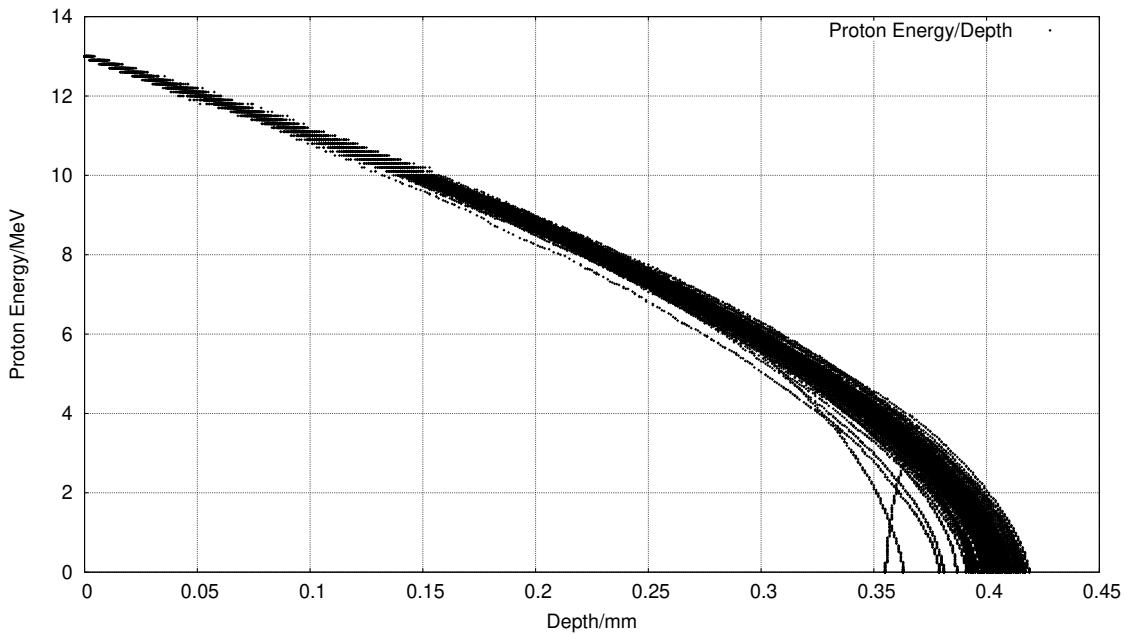


Figure 3.4: One hundred simulated 13MeV proton energy loss curves in Fe simulated with SRIM [11]

99.979% chance of decaying to Po-214 by beta decay and a 0.021% of emitting an alpha particle and decaying to Tl-210 (branching factors of 0.99979 and 0.00021 respectively) [10].

When a target material is irradiated, there is a source term for transmuted nuclei due to the irradiation. The daughter isotopes of these transmuted isotopes will also be affected by the irradiation and will transmute further, giving a source term for each daughter isotope as a result of the irradiation. Sources for each isotope in the decay chain, and branching factors between a parent isotope and its daughter isotope/s must be accounted for.

3.4.6 Simulating Ion Irradiation with SRIM

Move to Method

A package of ion transport codes, SRIM, is freely available to download and use to investigate the transport of ions through matter. SRIM uses the binary collision approximation (BCA) to simulate the passage of ions in a material. It is an approximate method, and one key restriction is that it does not take into account the structure of the material, and this approximation is therefore also imposed on the Activity code.

One file that SRIM creates is of importance to the Activity code, and that is the trajectory file that contains the energy and x,y,z co-ordinate data points for simulated ions moving through matter. Figure 3.4 shows the trajectory of one hundred 13MeV protons entering and passing through an Iron target, and it is this set of data points (together with the cross section database) that the Activity code uses to calculate the reaction rates for the transmutation of nuclei in the target. At higher energies, the ions slow as they lose energy due to electronic stopping, but as the ion energy drops the mechanism of loss through nuclear collisions becomes important. The spreading of ion depths at lower energies is a result of the higher momentum transfer during nuclear collisions, as can be seen in Figure 3.4.

Chapter 4

Background: Interatomic Potential Fitting

It is known that Chromium is

4.1 Experiment, Modelling and Theory

4.1.1 Introduction

Experiment: direct answers from physical reality Limited by technology of the time, also limits due to Heisenberg uncertainty principle Theory: state of the art theories have been replaced numerous times in the past Quantum mechanics is a very accurate theory under certain circumstances Some problems just too hard to solve with these theories Modelling: bridges the gap Experiment and theory have their flaws, so does modelling Helps show what experiment cant, and where theory is too hard to solve Aim: take very accurate DFT calculations based on quantum theory, extrapolate to a larger scale by fitting EAM potentials, open a path to simulations of Pd + Fe and Ru + Fe

Design a size and time scale diagram

DIRAC

Ab Initio: PWscf, Vasp, Siesta hundreds to thousands of atoms

Molecular Dynamics: DL_POLY, LAMMPS

4.1.2 Simulating Materials on a Variety of Scales in Time and Space

Stainless steel grain size less than 100 microns. A 1 micron grain would contain tens of billions of atoms. Given this number of atoms for a very small grain, it is very difficult to simulate just one grain over a short time period.

Certain properties derived ab initio assume the entire material is a single crystal, rather than made from grains of crystals.

4.2 Force Matching

To derive a potential, one may approach the problem from first principles in an attempt to replicate reality. It has been more useful, however, to lose any physical elegance [12] to give potentials that work for specific elements under certain conditions. Force data, gathered experimentally or by first-principles calculations, has been used to develop potentials since the 1990s. The force matching method was developed in 1994 by Ercolessi and Adams [13] to link the more accurate, more processor and memory intensive, world of first-principles calculations to Molecular Dynamics.

The force-matching method uses the difference between the actual force (either measured experimentally or calculated by first-principles calculations)

Given a set of M different atomic configurations, and a potential with a set of L parameters ($\vec{\alpha}$), the function Z_F is a measure of the difference between the forces calculated using the potential for all configurations and the actual (or DFT generated) forces.

$$Z_F(\vec{\alpha}) = \frac{1}{3 \sum_{k=1}^M N_k} \sum_{k=1}^M \sum_{i=1}^{N_k} |\vec{F}_{ki}(\vec{\alpha}) - \vec{F}_{ki}^0|^2 \quad (4.1)$$

$$Z_C(\vec{\alpha}) = \sum_{r=1}^{N_C} W_r [] \quad (4.2)$$

The overall match is measured by adding Z_F and Z_C

$$Z(\alpha) = Z_F(\vec{\alpha}) + Z_C(\vec{\alpha}) \quad (4.3)$$

The additional constraints used in this method are typically the bulk modulus[13], the three elastic constants (C_{11} , C_{12} and C_{44}) for cubic crystals [13] [14], stress tensors and phonons [14].

4.3 Properties of Cubic Crystals

4.3.1 Introduction

Cubic crystals are the most simple class with primitive unit cells having three orthonormal basis vectors. Four of the most common variants of the cubic crystal are the simple cubic, body centered cubic, face centered cubic and zincblende.

The force matching method uses DFT calculated forces and "additional constraints" in order to fit potential functions. These additional constraints include:

- Bulk Modulus
- Equation of State
- Stress Tensor
- Elastic Constants

4.3.2 Bulk Modulus

The Bulk Modulus of a material is defined as the bulk stress of a sample divided by the bulk strain on that sample. It is also the inverse of the compressibility of that material, which means that materials with a higher bulk modulus are less compressible than those with a lower value.

$$B_0 = -V \frac{\partial P}{\partial V} \quad (4.4)$$

$$B_0 = V \frac{\partial^2 E}{\partial V^2} \quad (4.5)$$

Table 4.1: Useful Conversion Factors

Material	Bulk Modulus/GPa
Aluminium	70
Iron (BCC)	110
Stainless Steel 18-8	163

4.3.3 Equation of State

The equation of state of a material relates either the pressure on that material as a function of the volume, or the energy of a sample of a material to the volume. This not only allows one to predict the energy or pressure at a certain volume, but also the minimum energy, relaxed volume and the bulk modulus.

There have been various

Murnaghan Equation of State

Hooke's law implies a linear relationship between stress and strain. In practice, where a pressure is applied to a material, the application of Hooke's law is limited [15]. To improve upon formulae developed in the 1930's, Muraghan derived a new equation (expand).

Birch-Murnaghan

[16]

Choice of Equation

The continued development of the variety of equations discussed is primarily due to high pressure and geoscience research, with change in volume measurements of materials under pressures of 100,000 atmospheres prompting Murnaghan's equation derivation in 1944. During the proposed process of fitting the equation of state in this work, the relaxed

4.3.4 Strain

Strain definition.

4.3.5 Stress

Stress definition . Stress is measured in Pa, although in this work it may also be measured in either $Ry/Bohr^3$ or eV/Ang^3 .

4.3.6 Voigt Notation

Where a tensor is symmetric, Voigt notation is used to simplify how the tensor is written.

$$\vec{A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} = \begin{bmatrix} A_{11} \\ A_{22} \\ A_{33} \\ A_{23} \\ A_{13} \\ A_{12} \end{bmatrix} \quad \text{if } \vec{A} \text{ is symmetric} \quad (4.6)$$

4.3.7 Elastic Constants

The Generalized Hooke's law relates the second order stress and strain tensors using a fourth order stiffness tensor.

$$\vec{\sigma}_{ij} = \vec{C}_{ijkl} \vec{\epsilon}_{kl} \quad (4.7)$$

$$\epsilon_{ij} = \begin{bmatrix} \epsilon_1 = \epsilon_{11} = \epsilon_{11} \\ \epsilon_2 = \epsilon_{22} = \epsilon_{22} \\ \epsilon_3 = \epsilon_{33} = \epsilon_{33} \\ \epsilon_4 = \epsilon_{23} = \epsilon_{32} \\ \epsilon_5 = \epsilon_{13} = \epsilon_{31} \\ \epsilon_6 = \epsilon_{12} = \epsilon_{21} \end{bmatrix} \quad (4.8)$$

elastic constants tensor.

Due to symmetry, a cubic crystal has only three elastic constants.

4.3.8 Calculating Elastic Constants for a Cubic Crystal

Applying strains to a cubic crystal [17] coupled with the calculation of the Bulk Modulus, as already discussed, allows the three independent elastic constants to be calculated.

$$\epsilon_{(C11-C22)} = \begin{bmatrix} \delta & 0 & 0 \\ 0 & -\delta & 0 \\ 0 & 0 & \delta^2/(1-\delta^2) \end{bmatrix} \quad (4.9)$$

$$\epsilon_{(C44)} = \begin{bmatrix} 0 & \frac{\delta}{2} & 0 \\ \frac{\delta}{2} & 0 & 0 \\ 0 & 0 & \delta^2/(4-\delta^2) \end{bmatrix} \quad (4.10)$$

4.4 Properties of Orthorhombic Crystals

4.4.1 Introduction

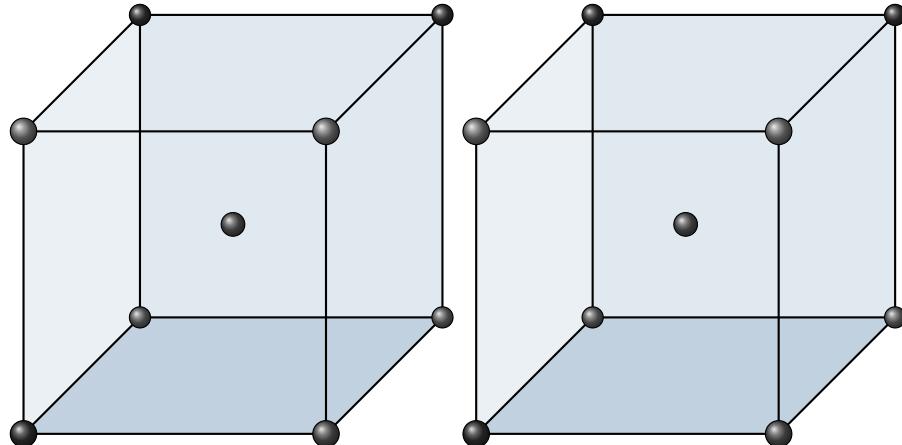
There are seven crystal classes, although this work is interested in only cubic and orthorhombic.

Table 4.2: Seven Crystal Classes

Class	Lengths	Angles
Cubic	$a = b = c$	$\alpha = \beta = \gamma = 90$
Hexagonal	$a = b, c$	$\alpha = \beta, \gamma = 120$
Rhombohedral	$a = b = c$	$\alpha = \beta = \gamma \neq 90$
Tetragonal	$a = b, c$	$\alpha = \beta = \gamma = 90$
Orthorhombic	a, b, c	$\alpha = \beta = \gamma = 90$
Monoclinic	a, b, c	$\alpha = \beta = 90, \gamma \neq 90$
Triclinic	a, b, c	$\alpha, \beta, \gamma,$

The DFT work here includes Palladium and Iron. The natural arrangement of Pd atoms in a pure sample are FCC within a cubic crystal. Pure iron at room temperature is BCC, but this work is interested in austenitic stainless steel where the structure of atoms in the alloy are FCC. When modelling FCC iron using DFT with a non-polarized calculation, the crystal favours a cubic crystal with the atoms fixed in the FCC positions. When a spin-polarized calculation is computed, with magnetization along the x-axis, the crystal becomes tetragonal (once again, the atoms are fixed in FCC positions).

While



$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{11} & C_{12} & 0 & 0 & 0 \\ C_{12} & C_{12} & C_{11} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{44} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{44} \end{bmatrix} \quad (4.11)$$

(3 independent values)

$$C_{ij} = \begin{bmatrix} C_{11} & C_{12} & C_{13} & 0 & 0 & 0 \\ C_{12} & C_{22} & C_{23} & 0 & 0 & 0 \\ C_{13} & C_{23} & C_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & C_{44} & 0 & 0 \\ 0 & 0 & 0 & 0 & C_{55} & 0 \\ 0 & 0 & 0 & 0 & 0 & C_{66} \end{bmatrix} \quad (4.12)$$

(9 independent values)

4.4.2 Calculation of Elastic Constants

Once the optimised parameters have been determined for the orthorhombic crystal, nine strains are applied to the crystal [18] in order to calculate the nine independent elastic constants.

$$E(V, \sigma) = E(V_0, 0) + V_0 \left(\sum_i \tau_i \epsilon_i \sigma_i + \frac{1}{2} \sum_{ij} c_{ij} \sigma_i \epsilon_i \sigma_j \epsilon_j \right) + O(\sigma^3) \quad (4.13)$$

The first three strains applied to the orthorhombic crystal

$$D_1 = \begin{bmatrix} 1 + \delta & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.14)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left(\tau_1 \sigma + \frac{c_{11}}{2} \sigma^2 \right) \quad (4.15)$$

$$D_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 + \delta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (4.16)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left(\tau_2 \sigma + \frac{c_{22}}{2} \sigma^2 \right) \quad (4.17)$$

$$D_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 + \delta \end{bmatrix} \quad (4.18)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left(\tau_3 \sigma + \frac{c_{33}}{2} \sigma^2 \right) \quad (4.19)$$

Volume conserving monoclinic distortions are then applied to the crystal to calculate the C_{11} , C_{22} and C_{33} elastic constants.

$$D_4 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \\ 0 & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (4.20)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left(2\tau_4 \sigma + 2 \frac{c_{44}}{2} \sigma^2 \right) \quad (4.21)$$

$$D_5 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \\ 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (4.22)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left(2\tau_5 \sigma + 2 \frac{c_{55}}{2} \sigma^2 \right) \quad (4.23)$$

$$D_6 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ \frac{\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (4.24)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left(2\tau_6 \sigma + 2 \frac{c_{66}}{2} \sigma^2 \right) \quad (4.25)$$

$$D_7 = \begin{bmatrix} \frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (4.26)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left((\tau_1 - \tau 2\sigma + \frac{1}{2}(c_{11} + c_{22} - 2c_{12})\sigma^2 \right) \quad (4.27)$$

$$D_8 = \begin{bmatrix} \frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (4.28)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left((\tau_1 - \tau 3\sigma + \frac{1}{2}(c_{11} + c_{33} - 2c_{13})\sigma^2 \right) \quad (4.29)$$

$$D_9 = \begin{bmatrix} \frac{1}{(1-\sigma^2)^{\frac{1}{3}}} & 0 & 0 \\ 0 & \frac{1+\sigma}{(1-\sigma^2)^{\frac{1}{3}}} & 0 \\ 0 & 0 & \frac{1-\sigma}{(1-\sigma^2)^{\frac{1}{3}}} \end{bmatrix} \quad (4.30)$$

$$E(V, \sigma) = E(V_0, 0) + V_0 \left((\tau_2 - \tau_3\sigma + \frac{1}{2}(c_{22} + c_{33} - 2c_{23})\sigma^2) \right) \quad (4.31)$$

Correlation of Melting Temperature in Metals with Elastic Constants

There is a correlation between the melting temperature of metals and their elastic constants [19].

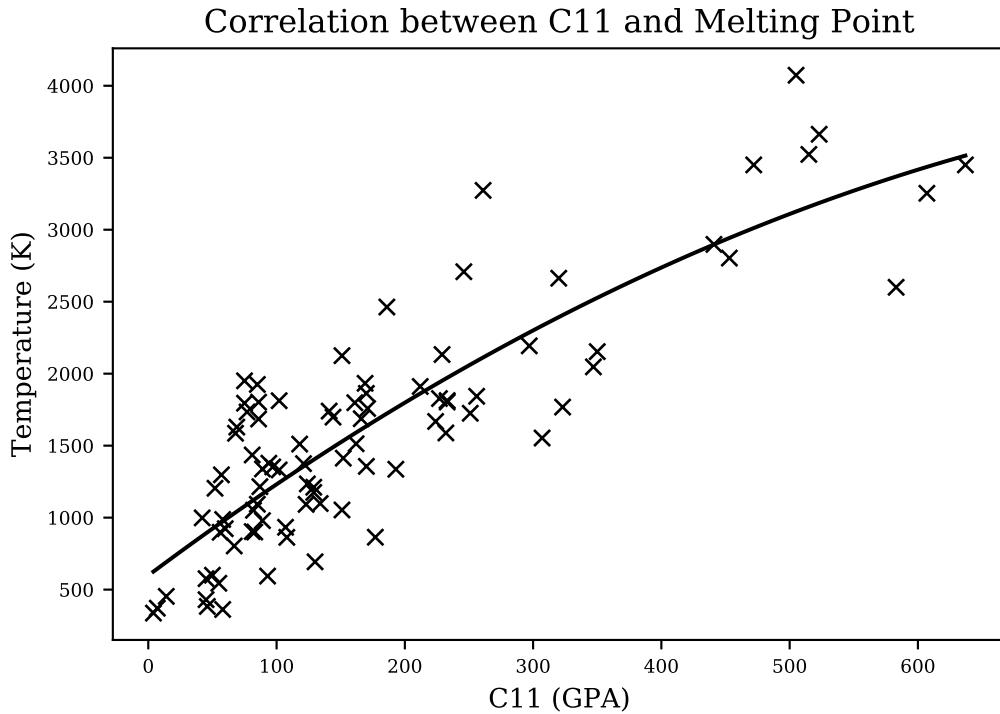


Figure 4.1: Graph caption

4.5 Magnetism

4.5.1 Brief History of Magnetism

Magnetic minerals have been known to and used by civilizations for millenia. Magnetite (lodestone) is a naturally occurring iron mineral.

In the 1800s, the link between electricity and magnetism was explored, beginning with experimental work by Oersted and Faraday, and culminating

4.5.2 Origin of Magnetism

Bohr Magneton

The Bohr magneton is a unit for measuring magnetic moment.

$$1 \text{ Bohr magneton} = 5.79 \text{ eV/T}$$

Electron Motion

The motion of an electric charge causes a magnetic field, and this is a consequence of special relativity.

$$\oint \vec{B} \cdot d\vec{l} = \mu_0 I \quad (4.32)$$

Intrinsic Magnetic Moment

4.5.3 Ferromagnetism and Antiferromagnetism

Hund's Rule

Electrons are arranged in shells, and each shell fills according to Hund's rule. Each subshell contains either zero, one or two electrons, and they fill such that each subshell of that shell contains at least one electron before two electrons occupy the same subshell. Iron has the Argon structure plus a full 4s shell and a partially filled 3d shell: [Ar] 3d7 4s2. The 3d shell contains five subshells and, for Iron, this will mean two paired

4.6 Solid State Physics

4.6.1 Schrodinger Equation

In classical mechanics, the Hamiltonian is the total energy of a system. In quantum mechanics, the Hamiltonian is an operator that is the combination of kinetic and potential energy.

$$\begin{aligned}\hat{H} &= \hat{T} + \hat{V} \\ \hat{H} &= [-\hbar^2/2m\delta^2] + V(\vec{r}, t)\end{aligned}\tag{4.33}$$

$$\hat{H}\psi = E\psi\tag{4.34}$$

4.6.2 Bravais Lattice

A Bravais lattice is a construct used to describe a periodic crystal lattice. It has the following properties:

$$\begin{aligned}\vec{R} &= n_1\vec{a}_1 + n_2\vec{a}_2 + n_3\vec{a}_3 \\ n_1, n_2, n_3 &\in \mathbb{Z} \\ \vec{a}_1, \vec{a}_2, \vec{a}_3 &\text{are independent}\end{aligned}\tag{4.35}$$

4.6.3 Reciprocal Lattice

Reciprocal space, also known as k-space or momentum space, is

$$\begin{aligned}\vec{G} &= m_1\vec{g}_1 + m_2\vec{g}_2 + m_3\vec{g}_3 \\ m_1, m_2, m_3 &\in \mathbb{Z} \\ \vec{a}_1, \vec{a}_2, \vec{a}_3 &\text{are independent}\end{aligned}\tag{4.36}$$

The relationship between the vectors of the primitive cell in real space and the primitive cell in reciprocal space is as follows:

$$\begin{aligned}\vec{g}_1 &= 2\pi/\Omega\vec{r}_2 \times \vec{r}_3 \\ \vec{g}_2 &= 2\pi/\Omega\vec{r}_3 \times \vec{r}_1 \\ \vec{g}_3 &= 2\pi/\Omega\vec{r}_1 \times \vec{r}_2 \\ \Omega &= \vec{r}_1 \cdot (\vec{r}_2 \times \vec{r}_3) \text{ volume of primitive cell in real space}\end{aligned}\tag{4.37}$$

4.6.4 Periodic Potential

4.7 Density Functional Theory

Density Functional Theory (DFT) is a branch of quantum chemistry that approximately solves the Schrödinger equation using electron density, rather than the coordinates of each electron in the system. A number of simplifications are also applied in order for DFT to be practical to use, but despite this calculations are limited to just hundreds or thousands of atoms. A calculation of a hundred or so atoms may take thousands of CPU hours at the time of writing, depending on the type of calculation and complexity of the electron structures of the atoms involved.

It is through DFT that the first principles energy, stress and force calculations will be made, and it's these results that the EAM potentials will be trained and fit to using the force matching method. This will allow much larger scale modelling using the extrapolated behaviour of accurate DFT calculations.

4.7.1 Brief Overview of DFT

Several important theories and approximations are used by DFT with the aim of calculating and minimising energies and forces. The Born-Oppenheimer approximation separates the electron-nucleus wave function. It treats the nuclei as fixed points, and the system of electrons in a fixed potential created by the nuclei.

the DFT of Kohn, Sham and Hohenberg proved that the potential of a system is uniquely determined by its ground state density. This makes solving the Schrödinger equation significantly easier.

4.7.2 Time Independent Schrödinger Equation

The Schrödinger equation is a linear partial differential wave equation, and it was proposed by Erwin Schrödinger in 1925. There is a time-dependent and time independent form of the equation, and as the DFT calculations in this work are static, only the time-independent version will be discussed.

$$\hat{H}|\Psi\rangle = E|\Psi\rangle \quad (4.38)$$

The Hamiltonian \hat{H} is an operator, and it is the total energy in the system. $|\Psi\rangle$ is the wave function (for example, the wavefunction of an electron) and this contains all the measurable information possible about whatever it represents. E is the energy eigen value of this system and this will depend on the eigenstate of the system. The wavefunction is also connected to the probability of a particle being found at a certain point in space, and the integral over all space is equal to 1 i.e. the probability of it being found somewhere in space is equal to 1.

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Psi(x, y, z) dx dy dz \quad (4.39)$$

The Schrödinger equation is set up depending on the system being studied. Starting with the simplest, a free particle, the only non zero energy of the Hamiltonian is kinetic.

$$\begin{aligned} \hat{H} &= \frac{\hbar^2}{2m} \nabla^2 \\ -\frac{\hbar^2}{2m} \nabla^2 \Psi(\vec{r}) &= E \Psi(\vec{r}) \end{aligned} \quad (4.40)$$

If the "particle" is in a potential, it will have both kinetic energy \hat{T} and potential energy \hat{V} .

$$\begin{aligned}\hat{H} &= \hat{T} + \hat{V} = \frac{\hbar^2}{2m} \nabla^2 + V(\vec{r}) \\ \left[-\frac{\hbar^2}{2m} \nabla^2 \right] \Psi(\vec{r}) &= E\Psi(\vec{r})\end{aligned}\quad (4.41)$$

Electronic structure calculations are important and allow the calculation of material properties that may be difficult or impossible to measure with current technology. The next step is to set up the Schrodinger equation (time-independent) for nuclei and electrons in a crystal. Relative to the strong force, the electromagnetic force is 1/137th as strong, but the strong force acts over a range of approximately 1.0×10^{-5} angstrom. In the calculations carried out in this section, the atoms will never be arranged close enough for the strong force to be considered at all. The gravitational force, as with the electromagnetic force, acts over an infinite range. The electromagnetic force is more than 1.0×10^{36} times greater than the gravitational force, so it too can be neglected. Finally, the weak interacting force has a range of approximately 1.0×10^{-8} angstrom which, as with the strong force, is a range small enough that the weak force may be neglected.

The energy operators required are kinetic and electromagnetic potential.

$$\hat{H} = \hat{T}_e + \hat{T}_n + \hat{V}_{e-e} + \hat{V}_{e-n} + \hat{V}_{n-n} \quad (4.42)$$

The first two terms are the kinetic energy of the electrons and nuclei respectively.

$$\begin{aligned}\hat{T}_e &= -\sum_i \frac{\hbar^2}{2m} \nabla^2 \text{Sum of kinetic energy of electrons} \\ \hat{T}_n &= -\sum_k \frac{\hbar^2}{2M} \nabla^2 \text{Sum of kinetic energy of nuclei}\end{aligned}\quad (4.43)$$

The last three terms are potential energy terms due to the electromagnetic force:

$$\begin{aligned}V_{e-e} &= \frac{1}{2} \sum_{i,j, i \neq j} \frac{1}{|\vec{r}_i - \vec{r}_j|} \text{sum of potential energy between electrons} \\ V_{e-n} &= \sum_{i,k} \frac{z_i}{|\vec{r}_i - \vec{r}_l|} \text{sum of potential energy between electrons and nuclei} \\ V_{n-n} &= \frac{1}{2} \sum_{k,l, k \neq l} \frac{z_k z_l}{|\vec{r}_l - \vec{r}_k|} \text{sum of potential energy between nuclei}\end{aligned}\quad (4.44)$$

These operators are now input into the TISE.

$$\left[\left(-\sum_i \frac{\hbar^2}{2m} - \sum_k \frac{\hbar^2}{2M} \right) \nabla^2 + \sum_{i,k} \frac{z_i}{|\vec{r}_i - \vec{r}_l|} + \frac{1}{2} \left(\sum_{i,j, i \neq j} \frac{1}{|\vec{r}_i - \vec{r}_j|} + \sum_{k,l, k \neq l} \frac{z_k z_l}{|\vec{r}_l - \vec{r}_k|} \right) \right] |\Psi\rangle = E|\Psi\rangle \quad (4.45)$$

4.7.3 Born-Oppenheimer

The TISE arrived at is far too complicated to solve, even for the simplest of systems. It represents the many body system of electrons and nuclei, and the variables it takes are the positions of all nuclei and electrons.

$$\hat{H}|\Psi(\vec{r}_e, \vec{r}_n)\rangle = E|\Psi(\vec{r}_e, \vec{r}_n)\rangle$$

where $\vec{r}_e = r_{e,1}, r_{e,2}, \dots, r_{e,i}$ (electron positions) (4.46)

where $\vec{r}_n = r_{n,1}, r_{n,2}, \dots, r_{n,i}$ (electron positions)

In 1927 the Born-Oppenheimer approximation was proposed to separate the electron components from the nuclei in the Hamiltonian.

Protons and Neutrons are almost 2,000 times the mass of electrons. With respect to the electrons, they move much slower and may be considered to be fixed or frozen in place. Simplifying for a moment to a single electron and proton, due to Newton's second law, we can see that the acceleration of the electron would be similarly 2,000 times that of the proton: $f_e = f_p$ and $m_e a_e = m_p a_p$ which leads to $a_e = \frac{m_p}{m_e} a_p$. As the nuclei move, the electrons are assumed to respond instantly, remaining in the ground state and not being promoted into higher energy levels.

The Hamiltonian of the electrons may be written with the electron co-ordinates as a variable, and the nuclei co-ordinates as a parameter.

$$\hat{H}_e(\vec{r}_e; \vec{r}_n) = \hat{T}_e(\vec{r}_e) + \hat{V}_{e-e}(\vec{r}_e) + \hat{V}_e - n(\vec{r}_e; \vec{r}_n) \quad (4.47)$$

The wavefunction and energy for the electrons may be calculated, although if the nuclear coordinates are changed, i.e. by changing the parameter r_n , the wavefunction and energy will need to be recalculated.

$$\hat{H}_e(\vec{r}_e; \vec{r}_n) \psi_e(\vec{r}_e; \vec{r}_n) = E_e(\vec{r}_n) \psi_e(\vec{r}_e; \vec{r}_n) \quad (4.48)$$

$$\Psi(\vec{r}_e, \vec{r}_n) = \chi_{ne}(\vec{r}_e) \psi_e(\vec{r}_e; \vec{r}_n) \quad (4.49)$$

4.7.4 Hohenberg-Kohn Theorem

4.7.5 Born-Oppenheimer

4.7.6 pseudopotentials

Valence electrons are those in the outer shell of an atom, and it is these electrons that are mostly responsible for the bonding of atoms. Iron, for example, has two valence electrons in the 4s shell; however, it is a transition metal and the partially empty 3d shell is also important to consider. The core electrons do not contribute as much to the bond and the model may be simplified using pseudopotentials.

Reduces the number of electrons. Simplifies the potential.

4.8 DFT: Magnetism

4.8.1 Collinear

4.8.2 Non-Collinear

It is simpler to simulate either antiferromagnetic or ferromagnetic configurations where the

4.9 Classical Molecular Dynamics

4.9.1 Verlet Timestep

- Build a neighbourlist
- Calculate the force on each atom
- Start loop with a time step Δt
- Calculate the half time step velocity $\vec{v}(t + 0.5\Delta t) = \vec{v}(t) + 0.5\Delta t \times a(t)$
- Using the half time step velocity, move the atoms to their new position
- Update or rebuild the neighbour list
- Recalculate the forces (and acceleration) at time $t + \Delta t$
- Calculate the end velocity
- ...

4.10 Kinetic Monte Carlo

4.11 Interatomic Potentials

4.11.1 Introduction

Classical molecular dynamics simulations are used to study small volumes of a material, with model sizes typically in the range of 10^5 to 10^6 atoms and time periods on the picosecond scale [20]. Such simulations have been used to model grain boundaries, and within these models irradiation damage may be simulated by initiating atom cascades that result from neutron radiation. Trautt and Mishin used molecular dynamics to study grain boundary migration in copper (12), and a more relevant example is the study by Shibuta et al and their model of grain boundary energy in bcc iron-chromium (13).

The temperature and pressure of the model can also be controlled, and this is important because there may be a temperature dependence on the rate of depletion of PGMs at the grain boundary, if there is depletion at all.

One key ingredient to any classical molecular dynamics simulation is the interatomic potential used to describe the forces between atoms in the model. Initially, simple pair potentials that described the force between two atoms were used.

4.11.2 Morse and Lennard Jones Pair Potentials

Early interatomic potentials were two-body pair potentials, where the force on an atom was determined by summing the pair potentials between that atom and other atoms in its neighbourhood, determined by a cutoff radius.

The Lennard-Jones and Morse potentials are examples of pair-potentials, but they are too simple. They cannot capture the many-body aspect of how atoms in a metal interact with each other. Many-body potentials such as the Finnis-Sinclair or Embedded Atom Model do have a pair potential function, and this can take the form of a LJ potential, Morse potential or other standard Pair Potentials. ... rewrite this part

4.11.3 Finnis Sinclair Potentials

Pair potentials had been used to model metals in simulations prior to EAM type potentials. The Finnis-Sinclair potential was published in 1984 and it introduced both a pair potential and an embedded term to take into account the cohesive energy dependent on the local electron density. The pair term represents the repulsion between the atoms whereas the embedding functional glues the atoms together in the solid. There is no directional term, and this was ignored in the Finnis-Sinclair model; the potentials were fit as well as possible empirically.

The embedding energy is dependent on the density function and it takes the form of a square root.

4.11.4 Embedded Atom Method

$$U_{EAM} = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N V_{ij}(r_{ij}) + \sum_{i=1}^N F[\rho_i] \quad (4.50)$$

where $\rho_i = \sum_{j=i, j \neq i}^N \rho_{ij}(r_{ij})$

Professor Howard Sheng created a website with many EAM potentials, and the plots for the three functions of the Aluminium EAM potential are shown below.

4.11.5 Two Band Embedded Atom Method

There are several variations of the EAM potential, and one of particular interest to us is the two-band model EAM (2BMEAM) that has two electron density and embedding energy terms. This formalism was originally developed to model Caesium (14), and the transition of electrons between S and D bands under pressure, but it has been modified to apply to alloys.

An alloy version of the two-band model was developed by Olsson et al. to investigate the -prime phase formation in Fe-Cr (15). It was further developed by Bonny et al to provide a reliable EAM type potential to model high-chromium ferritic alloys (16). This potential correctly predicts the change of sign in mixing enthalpy as the local concentration of Chromium changes, and the functions take the following form.

$$U_{EAM} = \frac{1}{2} \sum_{i=1}^N \sum_{j \neq i}^N V_{ij}(r_{ij}) + \sum_{i=1}^N F_D[\rho_{d,i}] + \sum_{i=1}^N F_S[\rho_{s,i}] \quad (4.51)$$

where $\rho_{d,i} = \sum_{j=i, j \neq i}^N \rho_{d,ij}(r_{ij})$ and $\rho_{s,i} = \sum_{j=i, j \neq i}^N \rho_{s,ij}(r_{ij})$

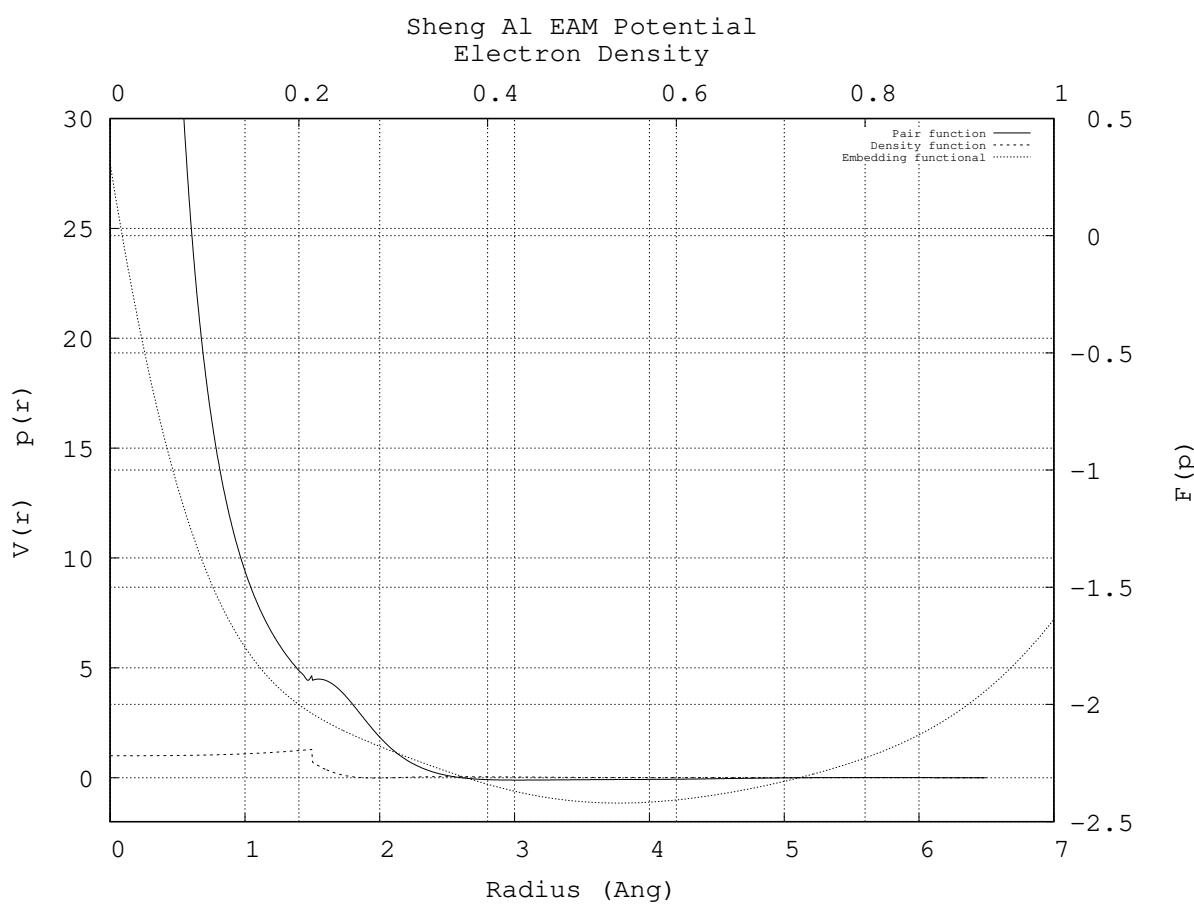


Figure 4.2: Graph caption

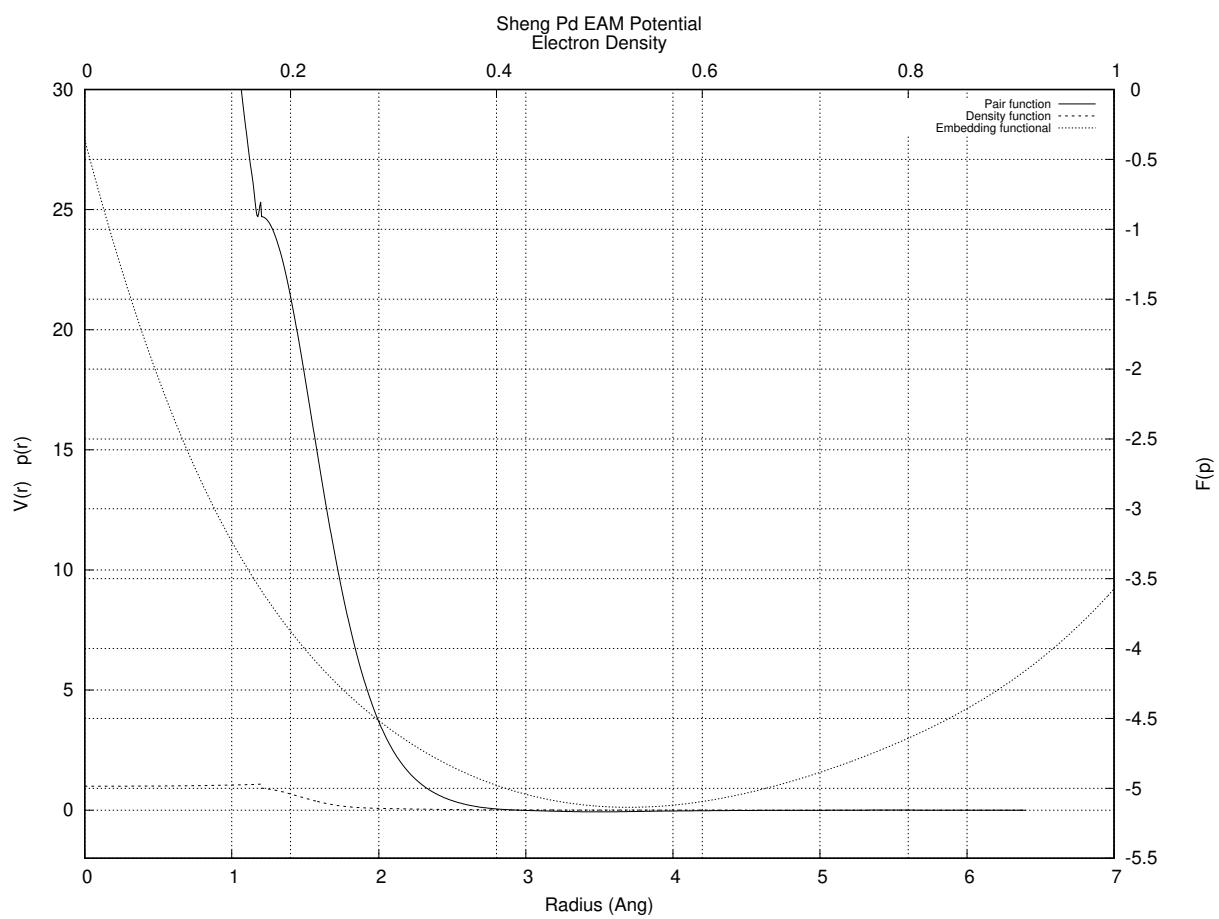


Figure 4.3: Graph caption

This allows a second embedding functional and electron density function to add/subtract energy to an atom when mixed as an alloy, but reverts to the original EAM for that element when in local concentrations of like-atoms.

4.11.6 ZBL Function

It became clear, while experimenting with a number of existing potentials and molecular dynamics programs, that at the very least modifications to those potentials would need to be made for small atom separations. A simulation to model a projectile failed early on due to the projectile's proximity to target atoms, resulting in the MD code returning an error as the separation was out of the range of the potential.

The Ziegler Biersack Littmark (ZBL) potential, as used by the SRIM computer code

Functional Forms of Embedded Atom Method Potentials

The functions used to represent the pair potential, electron density and embedding functional can be tabulated or calculated from analytic functions. By using an analytic function, there is the advantage of being able to produce tabulated versions of the potential functions and functional if required.

In a derivation of an EAM potential for Iron by Mendelev et al (17) the authors use a hybrid function between two exponential functions and a polynomial spline to represent the pair potential, and polynomial splines to represent the electron density function and embedding functional. Similarly, a potential has been derived for Uranium by Smirnova et al (18) that uses polynomial splines exclusively for the pair and density functions, and embedding functional.

Polynomial splines are attractive because they are continuous, have a continuous first order derivative for third order polynomials, a continuous second order derivative for fifth order polynomials, and they are flexible enough to create a potential that reproduces the forces predicted by Ab Initio. Although each segment of the spline is determined by four parameters, the nodes between splines can be adjusted which reduces the number of parameters that are adjusted during the derivation process.

Two forms of polynomial spline have been used in the literature:

$$f(r) = \sum_{i=1}^n A_i(r_i - r)H(r_i - r) \quad (4.52)$$

$$f(r) = \sum_{i=1}^n (a_i r^3 + b_i r^2 + c_i r + d_i) H(r_i^{upper} - r) H(r - r_i^{lower}) \quad (4.53)$$

The first type of spline has fewer parameters, but is a superposition of two or more polynomials up until the spline segment between the final two cutoff points. This makes it tricky to fit the parameters to give a desired starting potential. In the second type of spline, a single polynomial stitches between any two nodes. The boundary conditions at each node are: the output of the functions either side of the node be equal the first and second derivatives of the polynomials either side of the node are equal at the node. While there are more parameters to fit for the second type of spline, the number can be reduced by varying the position of the nodes and calculating the parameters by fitting the polynomials between the nodes. Given that we are deriving potentials for a metal alloy, a many body potential is needed. The EAM potential is a more general form of the Finnis-Sinclair type potential, and it has been used in many molecular dynamics investigations into metals. For these reasons, our aim is to derive EAM potentials for Fe-Pd and Fe-Ru. Two-band method EAMs have additional degrees of freedom and may be useful as they can capture any changes in the potential as the species mix and form the alloy. This is a secondary aim, the primary being standard EAM potentials. Molecular dynamics codes are capable of using tabulated functions. The form of the potential derived will be a spline of many polynomials that will be used to produce the tabulated versions of the potential functions. These are more flexible than other analytical potentials, they can be easily splined to other functions (such as the hard core Ziegler Biersack Littmark [ZBL] hard core potential) and continuous first and second derivatives can be forced. For a fourth order polynomial spline, the functions all take the following form:

$$f(r) = \sum_{i=1}^n (a_i r^3 + b_i r^2 + c_i r + d_i) H(r_i^{upper} - r) H(r - r_i^{lower}) \quad (4.54)$$

4.12 Fitting Interatomic Potentials

4.12.1 Analysing a Potential

To fit a potential, it must be analysed as it is changed by comparing the behaviour of a model using this potential to experimental or DFT derived values. The properties and values being compared in this thesis are:

- total energy of a collection of atoms
- forces between a collection of atoms
- bulk modulus, optimum energy and volume for an FCC crystal structure
- C_{11} , C_{12} and C_{44} elastic constants

4.13 Continuous Optimization

4.13.1 Introduction

Optimization is the process of finding the best solution for a problem that may also need to satisfy a number of imposed constraints. In terms of this work, there are several points in which optimization plays a key role, whether it be in the DFT and MD codes used or the fitting codes developed here.

The mathematical description of optimisation, as written in Numerical Optimization by Nocedal and Wright, is as follows:

Equations here

This work relies on optimization in several stages throughout. The DFT code used (PWScf) employs the Broyden-Fletcher-Goldfarb-Shanno algorithm to relax structures to give the optimum volume and optimal structure of an arrangement of atoms. During the fitting of the variety options of equation of state to the energy-volume points, the Levenberg-Marquart Algorithm (LMA) is used. Finally, a genetic algorithm and simulated annealing algorithm are coupled with LMA to locate the global optimal parameters for Interatomic Potential functions.

4.13.2 Continuous and Discrete Optimization

A discrete optimisation problem has a finite number of solutions, whereas the parameters of a continuous optimisation problem are infinite.

The traveling salesman is a well known optimisation problem where the aim is to calculate the shortest route between a set of cities. This is a discrete optimization problem. Fitting interatomic potentials is a continuous optimisation problem; the parameters are real numbers, with a continuous range for each parameter.

4.13.3 Global and Local Optimization

An example to discuss the difference between local and global optimization will be given in a very literal sense. I am located in Birmingham, and I wish to find the highest point local to me. Using a modern smartphone and map I could quickly find the nearest hill to my current position; this would be the local maxima. I would, however, need to go up and down many peaks and troughs, scouring the entire surface of the Earth, until I found the global maxima, somewhere amongst the many peaks and troughs in the Himalayas.

When optimising the parameters of a function, it is relatively easy to find a local extreme that would give me the optimum parameters locally. It is a much harder task to find the global optimum, especially if very little, or nothing at all, is known about the function for which the parameters are being optimised.

4.13.4 Global Optimization Algorithms

Simulated Annealing

A large amount of computing power is required to problems in Physics. The optimum arrangement of Iron atoms, at various concentrations and temperatures is one such example, but nature "calculates" the correct structure and reacts accordingly as the conditions change. It makes perfect sense to look to nature when designing optimisation algorithms.

Simulated annealing mimics the way a solid cools, with its atoms settling into the optimum relaxed positions as time passes. Initially, the solid is hot and the atoms are able to take non-optimal positions, but as the solid

cools the atoms take their optimal places. The heating allows the atoms to move from non-optimal positions in the initial configuration.

The simulated annealing algorithm takes in a starting set of parameters. These are varied and, if the new parameters give a better solution to the problem, the parameters are updated. However, there is a chance that a worse set of parameters are used, and this depends on how bad they are and the current value of the temperature in the algorithm. By allowing a worse set of parameters to be used, it gives the solution the chance to jump out of a local minimum and explore other solutions that it would otherwise be oblivious to.

Listing 4.1: Simple simulated annealing subroutine

```

1 // Simulated Annealing
2 subroutine simulated_annealing(f, p, pv, x, y, t, p_best)
3   // f -      (IN) the function f(p, x) for which the parameters are being optimised
4   // p -      (IN) array containing parameter/parameters
5   // pv -     (IN) array of maximum parameter/s variance
6   // x -      (IN) array of x points
7   // y -      (IN) array of fy points
8   // t -      (IN) starting temperature
9   // p_best - (OUT) optimised parameters
10  //
11
12  rss = (f(x[:,], p[:,]) - y[:,])**2
13
14  rss_best = rss
15  p_best[:] = p[:]
16
17  // These may be modified as required
18  outer_loops = 100
19  inner_loops = 1000
20  t_decrease = 0.99
21  pv_decrease = 0.99
22
23  for outer = 1, outer_loops
24    for inner = 1, inner_loops
25
26      r[:] = 0.5 - random_float(0.0, 1.0) // array of random floats (same size as p)
27
28      pt[:] = p[:] + r[:] * pv[:]
29      rss = (f(x[:,], pt[:,]) - y[:,])**2
30
31      // If better, always use new parameters
32      if (rss < rss_best) then
33        rss_best = rss
34        p[:] = pt[:]
35        p_best[:] = pt[:]
36
37      // If worse, sometimes use parameters (based on how good they are, and the temperature)
38      else
39        // as t -> 0 exp((rss_best - rss)/t) -> 0
40        // as (rss_best - rss) -> 0 exp((rss_best - rss)/t) -> 1 (where t = 1)
41        rn = random_float(0.0, 1.0)
42        if (rn <= exp((rss_best - rss) / t)) then
43          p[:] = pt[:]
44        end if
45      end if
46
47    end for
48
49    // Decrease temperature
50    t = t_decrease * t

```

```

51
52    // Decrease variance
53    pv[:] = pv_decrease * pv[:]
54
55    end for
56 end subroutine

```

Genetic Algorithm

It is unclear whether life started on this planet, or was brought to this planet, and this idea only removes the problem of how life as we are able to understand it was conceived to a different time and place. However it started, there is evidence of life on this planet approximately 3.5 billion years ago. Prokaryotes were the simple single cell organisms that inhabited our planet for billions of these. Their cell does not have a nucleus and the genetic information is contained in RNA and DNA within the cell.

Approximately 1.5 billion years ago, life made a leap forward in complexity and evolved into the first Eukaryotic cells. These are larger and more complex than Prokaryotes, and the genetic material is stored within a nucleus.

Through processes of inheritance, variation, natural selection and the vast expanse of time, more complex organisms developed.

As simulated annealing takes inspiration from cooling solids, genetic algorithms take inspiration from evolution.

Listing 4.2: Simple simulated annealing subroutine

```

1 // Simulated Annealing
2 subroutine simulated_annealing(f, p, pv, x, y, t, p_best)
3     // f -      (IN) the function f(p, x) for which the parameters are being optimised
4     // p -      (IN) array containing parameter/parameters
5     // pv -     (IN) array of maximum parameter/s variance
6     // x -      (IN) array of x points
7     // y -      (IN) array of y points
8     // t -      (IN) starting temperature
9     // p_best - (OUT) optimised parameters
10    //
11
12    // Set important variables
13    pop_size = 32
14    fresh_size = 8
15    generations = 100
16    chance_to_mutate = 0.1
17    extinction_chance = 0.1
18
19    // Create population array
20    pop = make_pop(pop_size, p, pv) // Some function that initialises the population
21
22    // Calc rss for each parameter set in the pop array
23    for n=1,pop_size
24        pop_rss[n] = sum((y[:] - f(pop[n,:], x[:]))**2)
25    end for
26
27    // Loop through generations
28    for n=1, generations
29
30        // Breed and replace if improvements
31        key = shuffled_list(1,generations) // Make a shuffled list of integers, unique, 1 to generations
32        for m=1, pop_size//2
33            parent_a = pop[key[2 * m - 1]]
34            parent_b = pop[key[2 * m]]
35            child_a, child_b = breed(parent_a, parent_b, chance_to_mutate) // Breed, possibly mutate

```

```

36
37     child_a_rss = sum((y[:] - f(child_a[:] - x[:]))**2)
38     child_b_rss = sum((y[:] - f(child_b[:] - x[:]))**2)
39
40     no_clones(pop, child_a, child_b) // if either already exists in pop, mutate to avoid clones
41     replace(pop) // some subroutine to replace parent/s with child parameters if better
42
43     update(p_best) // some subroutine to update p_best
44 end for
45
46 // Fresh solutions
47 fresh = make_pop(fresh_size, p, pv)
48 key = shuffled_list(1,generations) // Make a shuffled list of integers, unique, 1 to generations
49
50 // Breed with fresh parameters
51 for m=1, fresh_size
52     parent_a = pop[key[m]]
53     parent_b = fresh[m]
54
55     child_a, child_b = breed(parent_a, parent_b) // Breed
56
57     child_a_rss = sum((y[:] - f(child_a[:] - x[:]))**2)
58     child_b_rss = sum((y[:] - f(child_b[:] - x[:]))**2)
59
60     no_clones(pop, child_a, child_b) // if either already exists in pop, mutate to avoid clones
61     replace(pop) // some subroutine to replace parent with best child
62
63     update(p_best) // some subroutine to update p_best
64
65 end for
66
67 // Chance of extinction event to kill off bad parameters
68 extinction_event(pop, extinction_chance) // remove worst performing, replace with slight mutations of best
69 end for
70
71 end subroutine

```

4.13.5 Newton Gauss

The Newton Gauss method is an algorithm that finds the local minimum of a function by approximating the Hessian matrix with the Jacobian and its transpose. By estimating the Hessian, the algorithm only requires the computation of the first order derivatives with respect to each parameter, at each point pair x, y . It is possible to use the function and the analytical derivative, but as this work requires the calculation of functions that have no simple analytical form, the first derivative will be approximated within the algorithm.

$$\vec{J}^T \vec{J} \vec{p} = \vec{J}^T \vec{r} \quad (4.55)$$

$$\vec{r} = \vec{y} - f(\vec{x}, \vec{p})$$

4.13.6 Levenberg Marquardt

$$\left(\vec{J}^T \vec{W} \vec{J} + \lambda \text{diag} \left(\vec{J}^T \vec{W} \vec{J} \right) \right) \vec{p} = \left(\vec{J}^T \vec{W} \vec{r} \right) \quad (4.56)$$

$$\vec{r} = \vec{g} - f(\vec{x}, \vec{p})$$

Chapter 5

Methodology: Proton Activation and Radioactive Decay

Chapter Summary

5.0.1 Introduction

summary of the aim of this section

5.1 Fortran

There are a wide range of programming languages to chose from, each with its own set of advantages and disadvantages. At the time of writing, and out of the choice of Python, C++ and Fortran, the latter was selected.

Fortran is a compiled language that gives the developer direct control over memory allocation. Arrays of a reasonable size, approximately 8MB, may be stored in the stack memory, and for slower operations much larger arrays may be stored in the heap memory. In this respect, it is an unforgiving language as the developer must allocate and free memory correctly or risk segmentation faults.

Many modern computers, and all super computers, are multicore/multiprocessor machines. The OpenMP and OpenMPI libraries (and variants) allow programs to be developed to easily take advantage of parrallel programming. OpenMP allows multiple threads to access the same memory,

5.2 Activation by Ion Irradiation

The Bateman equation was derived using Laplace transforms, and this same method has been used to develop a modified equation that incorporates branching factors and production rates for each isotope in the decay chain, as illustrated by Figure 5.1.

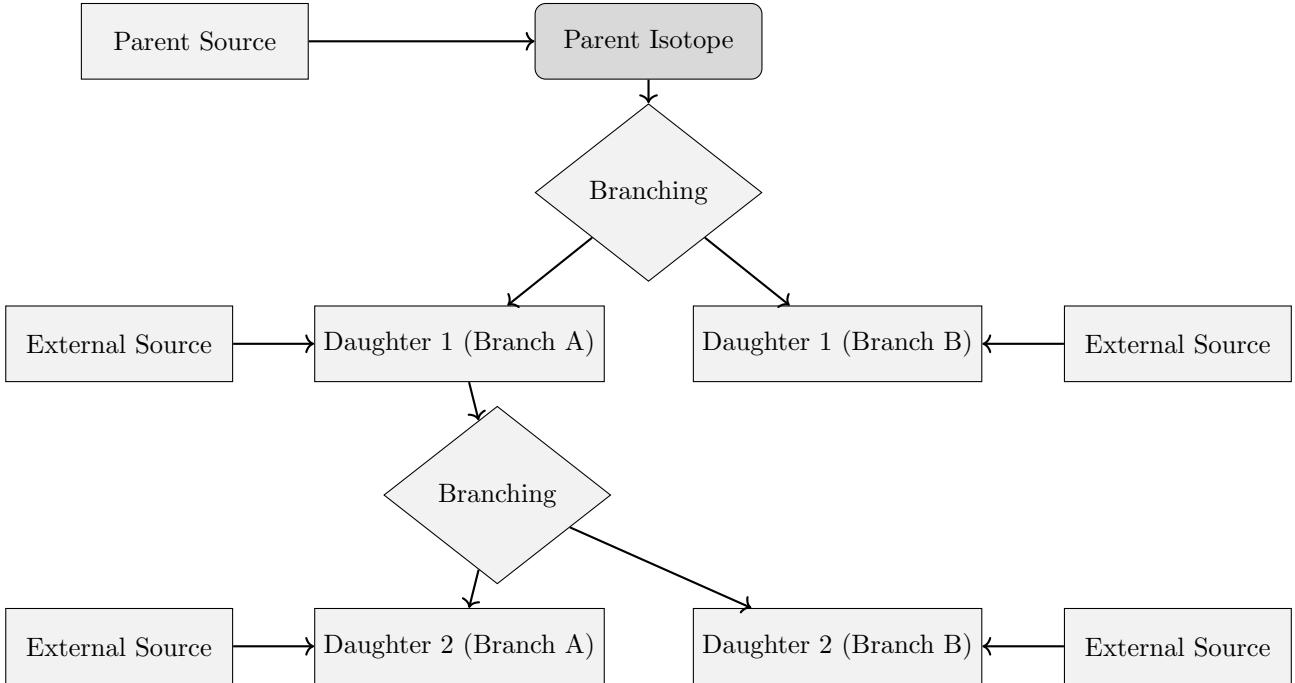


Figure 5.1: An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

5.2.1 Laplace Transform

Laplace Transforms (5.1) are a useful mathematical tool, and allow ordinary differential equations to be solved by simple algebraic manipulation in the s domain. Bateman took advantage of Laplace Transforms in deriving his equation, and this is the method that has been taken here as well.

$$F(s) = \int_0^{\infty} f(t) \exp(-st) dt \quad (5.1)$$

5.2.2 Constructing the Differential Equations

The first step is to set up differential equations for the parent isotope, unstable daughter isotopes and stable daughter isotope. The parent isotope has a source term, due to production, and a loss term, due to decay. The unstable daughter isotopes have two source terms, from the production by irradiation induced transmutation and the decay of preceding isotopes in the decay chain, and a loss term, due to decay. Finally, the stable daughter that finalizes the decay chain has two source terms (the same as the unstable daughters) but no loss term.

The variables (and vectors) used in these equations are defined as follows:

- $\vec{\lambda}$ vector containing isotope decay constants λ_i

- \vec{b} vector containing isotope to isotope branching factors b_i
- \vec{w} vector containing isotope production rates w_i
- t time at which activity/amount of isotope is measured
- $N_i(0)$ starting amount of the i^{th} isotope
- $N_i(t)$ amount of the i^{th} isotope at time t
- $N'_i(t)$ change in amount of the i^{th} isotope, with respect to time, at time t

The differential equations for the parent isotope (first isotope), unstable daughter isotopes (i^{th} isotopes) and stable, final, daughter isotope (z^{th} isotope) in the time domain are as follows:

$$N'_1(t) = \omega_1 - \lambda_1 N_1(t) \quad (5.2)$$

$$N'_i(t) = \omega_i + b_{i-1} \lambda_{i-1} N_{i-1}(t) - \lambda_i N_i(t) \quad (5.3)$$

$$N'_z(t) = \omega_z + b_{z-1} \lambda_{z-1} N_{z-1}(t) \quad (5.4)$$

Applying the Laplace Transform to these three differential equations allows them to be manipulated and solved algebraically in the s-domain.

$$N_1(s) = \frac{1}{s + \lambda_1} N_1(0) + \frac{1}{s(s + \lambda_1)} \omega_1 \quad (5.5)$$

$$N_i(s) = \frac{1}{s(s + \lambda_i)} (\omega_i) + \frac{1}{s + \lambda_i} (b_{i-1} \lambda_{i-1} N_{i-1}(s)) + \frac{1}{s + \lambda_i} N_i(0) \quad (5.6)$$

$$N_z(s) = \frac{1}{s^2} \omega_z + \frac{1}{s} b_{z-1} \lambda_{z-1} N_{z-1}(s) + \frac{1}{s} N_z(0) \quad (5.7)$$

5.2.3 Numerical Inversion of the Laplace Transform

The Gaver-Stehfest[21] algorithm was developed in the 1960s and 1970s and is a method of calculating the inverse of a Laplace Transform in the real number domain. It is an easy to implement and reasonably accurate method, although it is an approximation to the real value. A comparison between an analytic and numeric inversion for the unstable isotope Po-218 is discussed at the end of this section (figure 5.2).

$$f(t) \approx f_n(t) = \frac{\ln(2)}{t} \sum_{k=1}^{2n} a_k(n) F(s) \text{ where } n \geq 1, t > 0 \quad (5.8)$$

$$s = \frac{k \ln(2)}{t} \quad (5.9)$$

$$a_k(n) = \frac{(-1)^{(n+k)}}{n!} \sum_{j=\text{Floor}(\frac{k+1}{2})} j^{n+1} \binom{n}{j} \binom{2j}{j} \binom{j}{k-j} \quad (5.10)$$

The equation for the i^{th} isotope may be calculated by recursively calculating the equations by numeric inversion, starting from the first (parent isotope) and inserting the result into each subsequent recursion until the i^{th} isotope is reached (changing the equations appropriately for the parent, unstable daughter and stable daughter isotopes).

5.2.4 Analytic Solution by Partial Fraction Expansion

The equation for the i^{th} isotope in the s domain can be written in full by substituting the preceding equation until the parent isotope is reached, and this full equation may be rearranged with the production amount of each isotope and starting amount of each isotope in individual terms. Each of these terms is multiplied by a fraction that can be expanded, using partial fractions, and inverted analytically.

This is illustrated with an example unstable isotope, fourth in the decay chain (including the parent isotope):

$$\begin{aligned}
 N_4(s) = & \frac{1}{(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0) \\
 & + \frac{1}{(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\
 & + \frac{1}{(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 N_3(0) \\
 & + \frac{1}{(s + \lambda_4)} N_4(0) \\
 & + \frac{1}{s(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\
 & + \frac{1}{s(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 \omega_2 \\
 & + \frac{1}{s(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 \omega_3 \\
 & + \frac{1}{s(s + \lambda_4)} \omega_4
 \end{aligned} \tag{5.11}$$

An example stable isotope, fourth (last) in the decay chain (including the parent isotope):

$$\begin{aligned}
 N_4(s) = & \frac{1}{s(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0) \\
 & + \frac{1}{s(s + \lambda_2)(s + \lambda_3)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\
 & + \frac{1}{s(s + \lambda_3)} b_4 \lambda_3 N_3(0) \\
 & + N_4(0) \\
 & + \frac{1}{s^2(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\
 & + \frac{1}{s^2(s + \lambda_2)(s + \lambda_3)} b_3 b_4 \lambda_2 \lambda_3 \omega_2 \\
 & + \frac{1}{s^2(s + \lambda_3)} b_4 \lambda_3 \omega_3 \\
 & + \frac{1}{s^2} \omega_4
 \end{aligned} \tag{5.12}$$

By using partial fraction expansion and standard Laplace Transforms, the set of equations below is used to calculate the amount of the m^{th} isotope in the decay chain, providing the m^{th} isotope is unstable.

$$N_m(t; \vec{\lambda}, \vec{b}, \vec{w}) = \sum_{k=1,m} r(k; \vec{\lambda}, \vec{b}) \left[f(t; k, m, \vec{\lambda}) N_k(0) + g(t; k, m, \vec{\lambda}) w_k \right] \quad (5.13)$$

$$r(k, m, \vec{\lambda}) = \begin{cases} \prod_{i=k, m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \quad (5.14)$$

$$f(t; k, m, \vec{\lambda}) = (-1)^{m-k} \sum_{i=k, m} \left[\exp(-\lambda_i t) \prod_{j=k, m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (5.15)$$

$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m} \lambda_i} + (-1)^{m-k+1} \sum_{i=k, m} \left[\frac{1}{\lambda_i} \exp(-\lambda_i t) \prod_{j=k, m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (5.16)$$

The set of equations below is used to calculate the amount of the m^{th} isotope in the decay chain, where the m^{th} isotope is stable.

$$N_m(t; \vec{\lambda}, \vec{b}, \vec{w}) = N_m + w_m t + \sum_{k=1, m-1} r(k; \vec{\lambda}, \vec{b}) \left[f(t; k, m-1, \vec{\lambda}) N_k(0) + g(t; k, m, \vec{\lambda}) w_k \right] \quad (5.17)$$

$$r(k, m, \vec{\lambda}) = \begin{cases} \prod_{i=k, m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \quad (5.18)$$

$$f(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m} \lambda_i} + (-1)^{m-k+1} \sum_{i=k, m} \left[\frac{1}{\lambda_i} \exp(-\lambda_i t) \prod_{j=k, m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (5.19)$$

$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m} \lambda_i} t + \frac{\sum_{i=k, m} \left[\prod_{j=k, m; j \neq i} \lambda_j \right]}{\prod_{i=k, m} \lambda_i^2} + (-1)^{m-k+1} \sum_{i=k, m} \left[\frac{1}{\lambda_i^2} \exp(-\lambda_i t) \prod_{j=k, m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (5.20)$$

5.2.5 Preference: Analytic over Numeric

The numeric solution only requires the equation to be solved in the s-domain; the Gaver-Stehfest algorithm performs the inversion. It is worth the extra effort to derive and implement an analytic solution, as the numeric is only an approximation. Examples of the pitfalls of the numeric solution are that it can give negative amounts of an isotope and the difference between the numeric and analytic calculated amounts can become quite large when the isotope decays away to a very small value. Figure 5.2 shows the predicted decay of a sample of Po-218 irradiated for 1,000s, and sampled until 10,000s. In the region between 4,000s and 9,000s the amount from

Figure 5.2: Decay of Po-218: Analytic and Gaver-Stehfest Calculations [10]

the numeric calculation drops below zero, whereas the analytic calculation remains above zero, as would be expected.

5.3 Computational Methods

The Activity program has been developed in Fortran and takes advantage of MPI (Message Parsing Interface) to speed up calculation times by allowing the use of multiple processes in parallel. It has a self contained maths library, although this could be improved in the future by using optimised maths libraries for certain functions (e.g. matrix operations).

The code was developed on a Debian based distribution of Linux, but it should be supported on other variants of Linux and Unix, and does not require any specialist hardware.

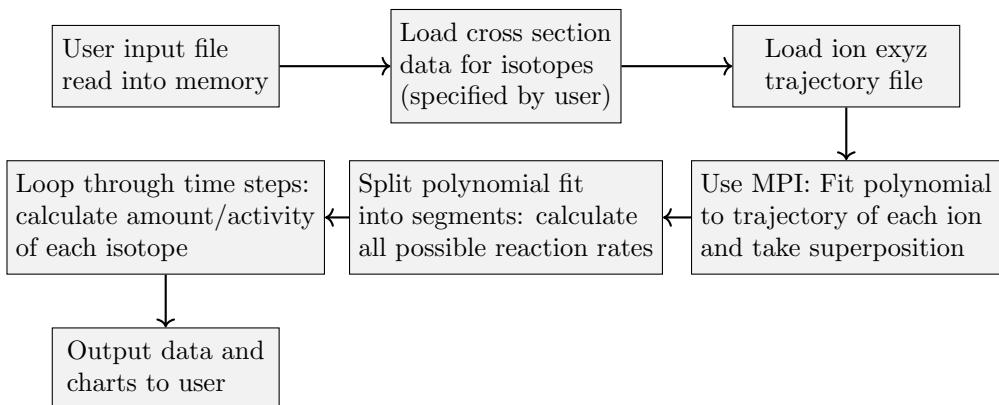


Figure 5.3: Flow chart of major processes in the Activity code

The user is required to prepare an input file that contains the instructions required to perform a calculation. In addition to the input file, the user must provide an EXYZ ion trajectory file output by SRIM. Activity will read in the user input file, and the SRIM and data files listed within, before performing the calculation. Figure 5.3 shows a flowchart of the major steps the code performs.

There are various settings in the user input file, but the main ones relating to the simulated experiment are:

- Element composition of target (percentage by mass).
- Beam flux (current), energy, duration and area on target.
- Activity measurement time (end of the “experiment”).
- Material density.
- Target thickness.

5.4 SRIM

5.5 Activation by Neutron Irradiation

5.5.1 Introduction

A short program was created, based on the existing code developed to calculate ion activation. It is a simple non-transport neutron activation code, used to estimate the activity and subsequent cooling of materials irradiated by neutrons. A single energy neutron flux or Maxwell-Boltzmann distribution may be selected by the user and the TENDL-2019 data file is used to provide cross sections for calculating reaction rates. Finally, the previously derived extended Bateman equations are used to calculate isotopes at time t after irradiation begins.

Chapter 6

Methodology: Interatomic Potential Fitting

Chapter Summary

6.0.1 Introduction

Refresher of the aim of this section - to derive an interatomic potential for Fe-Pd.

6.1 Math Techniques

6.1.1 Lagrange Polynomial Interpolation

The functions and functional that constitute the potential are represented in computer codes as a table of values. The values are discrete points that are evenly spaced, and Lagrange polynomials are used to interpolate values.

$$D = \{(x_0, y_0) (x_1, y_1) (x_2, y_2) \dots (x_n, y_n)\} \quad (6.1)$$

$$p_n(x) = \sum_{i=0}^n y_i L(i, x) \text{ where } l(i, x) = \prod_{j=0, j \neq i}^n (x - x_j) \quad (6.2)$$

Listing 6.1: Add two numbers function

```

1 // lagrange polynomial interpolation
2 function lpinterp_y(x, data)
3   // x - point value being interpolated at to calc f(x)
4   // data - four x,y data points (n by 2 array)
5   n = len(data, 1)
6   y = 0.0
7   for i = 1,n
8     l = 1.0
9     for j = 1,n
10       if (i != j) then
11         l = l * (x - data[j][1]) / (data[i][1] - data[j][1])
12       end if
13     next j
14     y = y + l * data[i][2]
15   next i
16 end function lpinterp_y

```

There are typically hundreds or thousands of data points, so to interpolate, the closest few points are used. Throughout the computer code four point interpolation was the preferred method, to balance computational speed with a well fitting polynomial.

The gradient of the potential functions are also computed using lagrange polynomials. The equation used is given below

$$q_n(x) = \sum_{i=0}^{i=n} y_i g(i, x) \quad (6.3)$$

where $g(i, x) = \left(\prod_{j=0, j \neq i}^{j=n} \frac{1}{x_i - x_j} \right) \times \left(\sum_{k=0, k \neq i}^{k=n} \prod_{j=0, j \neq k, j \neq i}^{j=n} (x - x_j) \right)$

```

1 // lagrange polynomial interpolation
2 function lpinterp_dydx(x, data)
3   // x - point value being interpolated at fo calc f'(x)
4   // data - four x,y data points (n by 2 array)
5   n = len(data, 1)
6   dydx = 0.0
7   for i = 1,n

```

```

8      fx = 1.0
9      gx = 1.0
10     for j = 1, n
11       if (i != j) then
12         fx = fx / (data[i][1] - data[j][1])
13         psum = 1.0
14         for k = 1, n
15           if (i != k and j != k) then
16             psum = psum * (x - data[k][1])
17           end if
18         next k
19         gx = gx + psum
20       end if
21     next j
22   dydx = dydx + fx * gx * data[i][2]
23 next i
24 end function lpinterp_dydx

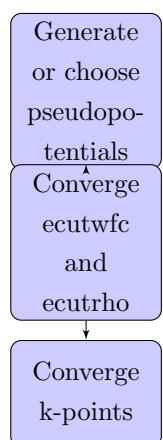
```

6.1.2 Splines

6.1.3 Simulated Annealing

6.1.4 Levenberg-Marquardt Optimisation

6.2 Interatomic Potential Fitting



6.3 Density Functional Theory

6.3.1 Quantum Espresso

Degauss values 0.01Ry Methessel-Paxton [22] Fermi-Dirac 0.01eV (0.00074Ry) [23] Marzari-Vanderbilt 0.01Ry [24] Degauss 0.03 + 0.05 [25] Marzari-Vanderbilt 0.05Ry [26]

6.3.2 Crystal Structure

A large amount of computing power is required to run the DFT calculations. If the input parameters are not accurate enough to begin with, it will either take a long time for the DFT calculation to run, or it will fail to converge completely.

Element	Atomic Mass	Density kg/m ³	Atoms/m ³	FCC (Bohr/Angstrom)	BCC (Bohr/Angstrom)
Al	26.98	2700	6.02×10^{28}	7.66/4.05	6.08/3.22
Fe	55.84	7874	8.47×10^{28}	6.83/3.61	5.42/2.87
Pd	106.42	12023	6.83×10^{28}	7.34/3.88	5.83/3.08
Ni	58.69	8908	9.09×10^{28}	6.67/3.53	5.30/2.80

Table 6.1: Predicted lattice parameters based on the density, atomic number and type of structure

6.3.3 Pseudopotential Selection

6.3.4 Qeconverge Python Program

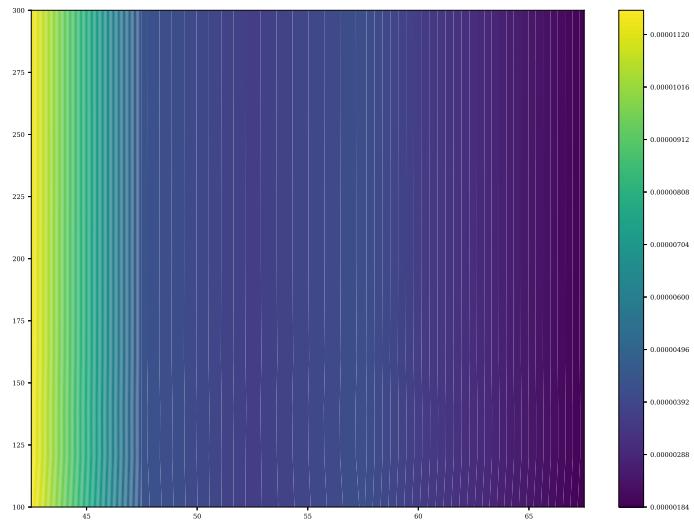
The process of converging cutoff parameters is one that may be automated. A program was written in Python to automatically converge the ecutwfc and ecutrho values within a specified threshold.

The program reads an input file into memory, and this contains the settings for the convergence run. A template PWscf input file is also loaded, and this will have any required PWscf settings, such as the pseudopotentials, atom species and so on.

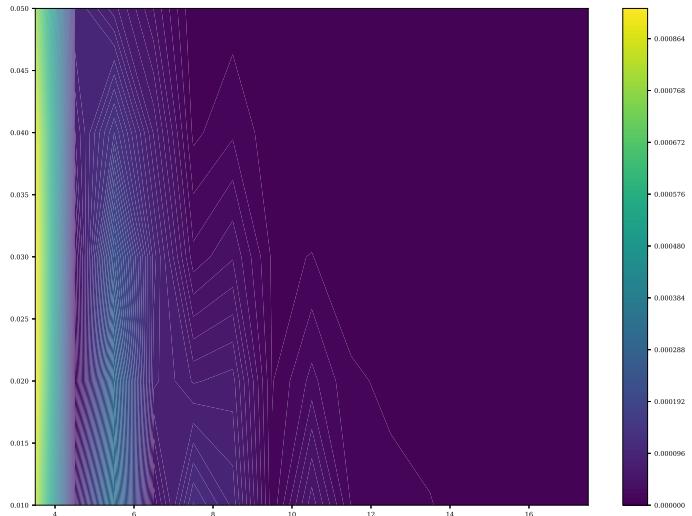
Once these files have been loaded, the first stage of the program runs to determine the ecutwfc and ecutrho values that are within the force and energy convergence thresholds. The program creates a crystal based on the settings, for example an FCC, BCC or SC crystal supercell. The exact atomic positions are likely to be the optimal, relaxed, positions and so the overall forces on each atom will be zero. The program randomly varies the atomic positions, and this results in a configuration with forces between the atoms.

The wfc value starts with the user defined ecutwfc starting value, and this is increased until the change in energy and force both converge to within the specified thresholds. The ecutrho value is increased simultaneously, and is four times the ecutwfc value. In the second stage, the ecutrho value is decreased and the ecutwfc value is held constant while the energy and force remain within their respective convergence thresholds. Finally, the program attempts to decrease the ecutwfc value further.

Following the energy and charge density cutoff, the program then runs a number of calculations in order to produce a number of colour density plots of energy cutoff vs density cutoff, plotting the convergence of total energy and total force. This allows the user to visualise the convergence surface and select cutoff values manually.



The final part of the program is to help the user decide upon degauss and k-point values. The converged `ecutwfc` and `ecutrho` values from the first part of the code are used, and the k-point start, end and increment amounts are set by the user, as is a list of smearing degauss values. 2D colour plots of force and energy convergence as smearing changes and k-point value changes are prepared and saved so the user may study these to determine the best combination for their application.



The randomised atom positions use a seed that may be set in the input file. This means that, although they are pseudo-rng generated, they are repeatable given the same random seed. Every successful PWscf calculation is logged and saved, and if the program runs multiple times, the cached output files are used rather than recalculating every input file.

6.4 DFT Calculations: Iron and Palladium

6.5 High Performance Computing

6.5.1 Processing and Memory Requirements

Although DFT calculations already makes a number of approximations to allow their completion in a reasonable timescale, they are still too complicated and resource intensive to run on even high powered home or business computers. In terms of processing power, what would take weeks to run on a home PC may be completed in a few hours using a high performance computer. In terms of memory, certain problems would be impractical or impossible on a home PC as they require up to or more than a hundred gigabytes of memory, which is rarely available outside of HPC.

6.5.2 HECToR and Archer

HECToR was a supercomputer located in Edinburgh. Manufactured by Cray, it was available for scientists to apply for computing time. Archer replaced HECToR in 2014 and, also manufactured by Cray, is the current leading super computer in the UK.

6.5.3 BlueBEAR

Brief: throughout the process, DFT, potential fitting and future kmc or MD simulations, need high performance computing. Time applied for on Archer and BlueBEAR.

6.6 Reference Database

In order to fit the potentials, it was necessary to build a reference database. This consists of known experimental bulk properties along with forces, energies and stresses calculated using DFT.

32 atom randomised

108 atom defects

6.6.1 DFT Calibration

Choice of Pseudopotentials

Energy and Charge Density Cutoffs

K-Points

nbnd

6.6.2 Atomic Configurations for DFT Calculations

The interatomic potentials are designed to reproduce the forces, energies, stresses and bulk properties

6.7 Potential Analysis and Fitting Code

6.7.1 Tabulated Potentials and Interpolation

Lagrange Interpolation

Lagrange interpolation is a valuable tool when evaluating or optimising potentials. Rather than fit a polynomial that exactly passes through n points, Lagrange interpolation returns the value of $f(x)$ only. This is much faster, computationally, than fitting an n th order polynomial, but it doesn't return enough information in itself to calculate $f'(x)$ and $f''(x)$. By using the Lagrange interpolation algorithm recursively, the first and second derivative values are calculated. This makes it a useful tool for splining between nodes, where the x , $f(x)$, $f'(x)$ and $f''(x)$ are required, and for energy and force calculations from tabulated potentials where the radius and density values of the functions/functionals fall between tabulated points. For a set of data points:

$$D = \{(x_0, y_0)\} \quad (6.4)$$

An alternative method is to set up a system of linear equations and solve this by matrix inversion. Using Fortran, both methods were evaluated for three, four and five point interpolation. The outcome is that Lagrange Interpolation shows a decrease of processing time of up to a factor of ten.

$$P(n, x) = \sum_{i=1}^{i=n} y_n L_n(x) \quad (6.5)$$

6.7.2 Simulated Annealing

6.7.3 Levenberg-Marquardt Optimisation

6.8 Calculating Crystal Properties

6.8.1 Introduction

6.8.2 Bulk Modulus

This method for calculating the bulk-modulus was used with both for DFT calculations and within the potential analysis code.

$$E(V, E_0, V_0, B_0, B'_0) = E_0 + \frac{9}{16} B_0 V_0 (V^{\frac{2}{3}} V_0^{-\frac{2}{3}} - 1)^2 \\ \times (6 + B'_0 (V^{\frac{2}{3}} V_0^{-\frac{2}{3}} - 1) - 4V^{\frac{2}{3}} V_0^{-\frac{2}{3}}) \quad (6.6)$$

A number of homogenous strains (positive and negative) are applied to the relaxed primitive cell. The energy and volume measurements for each strain are recorded, and a second order polynomial is fit to the data points. By rearranging the Birch-Murnaghan and either using Levenberg-Marquardt or some similar optimisation technique, the four parameters are fit to the data. A maximum strain of a few percent was applied throughout this work.

6.9 DL_POLY Contribution

6.9.1 Introduction

DL_POLY is a Molecular Dynamics code developed by W. Smith, T.R. Forester and I.T. Todorov at Daresbury Laboratory in Warrington. It is written in Fortran and, before the modifications, included a number of potential types for metals including

- Finnis Sinclair
- EAM
- EEAM

The Finnis Sinclair is a particular form of the EAM potential, and EEAM is a modification where, if the metal is an alloy, the density and embedding functional for each atom type are treated separately.

6.9.2 Modifying DL_POLY: 2BEAM

A meeting was held with Dr. Todorov at Daresbury Laboratory, where a brief overview of the relevant code was given. After this point, and corresponding by email, the two band EAM type was added (2BEAM). Two major additions to the code were an array used to store the d-band density function and d-band embedding functional, as well as the s-band function/functional (which are stored in the existing two arrays). The calculation of energy and forces on atoms was also altered to make use of these arrays when a two band EAM potential is used.

6.9.3 Mailshot Extract

Listing 6.2: Mailshot Extract

```
1 DL_POLY_4.05: New Release & Events - MAILSHOT 013
2
3 NEW FEATURES \& IMPROVEMENTS
4 -----
5 1. New two band (2B) EAM and EEAM potentials for metals (TEST45 and TEST46).
6
7 Acknowledgements
8 -----
9 Ben Palmer \@ University of Birmingham (UK) for contributing to the
10 development and testing of the 2BEAM for metals;
11
12 Regards,
13
14 Ilian Todorov
15 July 2013
```

6.10 Continuous Optimization

6.10.1 Implementation of Optimization Algorithms

The two main languages used throughout this work were Fortran and Python. While Fortran may be compiled into a shared object library and imported into Python using the Numpy utility f2py, there are several bugs with f2py that need to be worked out to enable calls to external functions defined within Python.

As a result of these issues, the optimization algorithms were written in Python with Numpy leveraged where possible to speed up the optimization process.

6.10.2 LMA

(delete or rewrite)

The LMA is a Newton-Gauss type algorithm that includes the addition of a dampening term that helps to increase the robustness of the algorithm. The particular algorithm here uses a number of other schemes to improve the overall effectiveness of the algorithm.

The starting value for lambda is calculated as a function of the estimate of the Hessian and a cutoff for lambda is introduced to remove dampening altogether. A delayed gratification scheme has also been introduced to increase lambda by 50 percent if the trial solution is worse, and to decrease lambda by a factor of 5 if the trial solution is better. Finally, a diagonal weighting matrix has been included to allow certain parameters to be preferenced during their optimisation.

Chapter 7

Activity Code Development & Publication

Chapter Summary

7.1 Activation by Ion Irradiation

7.2 Computer Package Development

|Intro|

7.3 Convergence of Key DFT Settings and Testing

7.3.1 Introduction

Prior to generating a database of atom configurations, calculated forces, stresses and energies, key DFT settings needed to be selected and tested. The Python code Qeconverge was written and used to converge the ecutwfc, ecutrho, k-points and degauss smearing values for each of the pseudopotentials required.

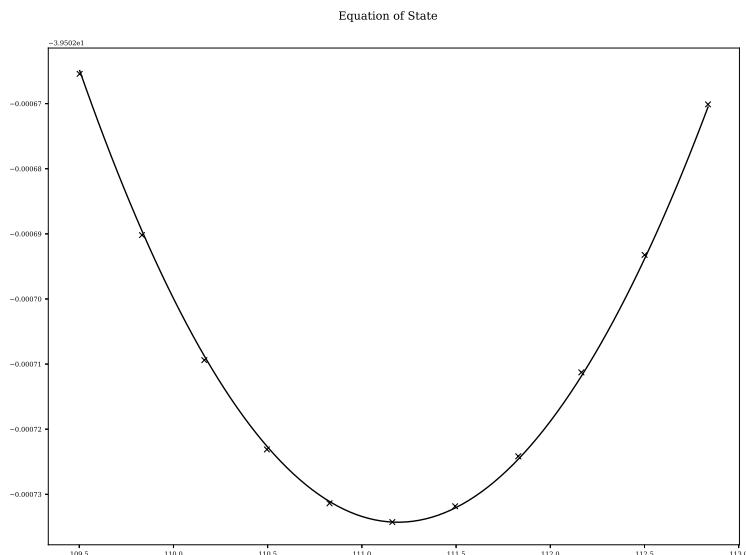
Aluminium is considered here because it was a simpler atom to run trial calculations with, and to compare to experimental results.

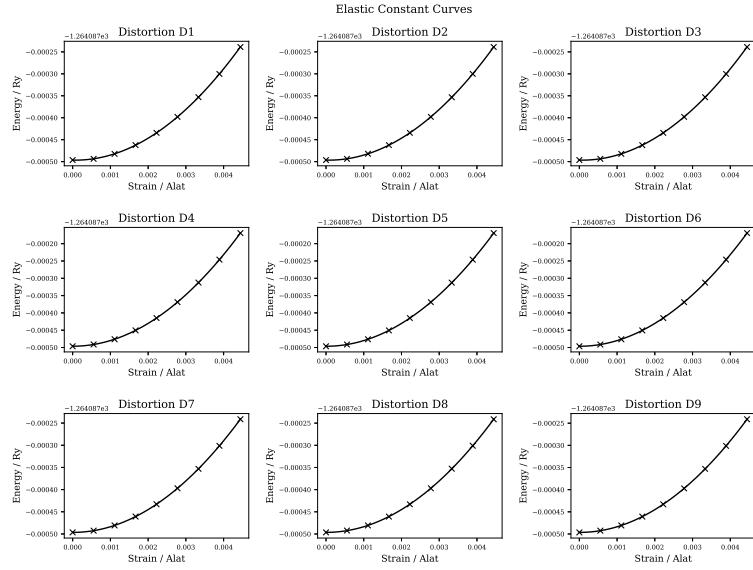
Element	Pseudopotential	Ecutwfc	Ecutrho	K-points	Degauss
Al	Al PBE KJPAW	50	200	11	0.04
Fe					
Pd					

Table 7.1: DFT Settings

7.3.2 DFT Crystal Property Calculations

The Qeeos Python program was used to create the DFT input files and perform the operations required to calculate estimated values of various properties of the material. This enables two things: a check on how good the DFT calculations, with the selected pseudopotentials and settings, compare to the experimental values, and it generates a large number of configurations that are added to the DFT reference database.





Property	Experimental	Previous Work	This Work	RSS
Lattice Parameter (Ang)				
Density (kgm ⁻³)				
alat				

Table 7.2: DFT Settings

Chapter 8

Ab Initio Reference Database

Chapter Summary

8.1 Activation by Ion Irradiation

8.2 Computer Package Development

|Intro;

Chapter 9

Interatomic Potential Fitting

Chapter Summary

9.1 Activation by Ion Irradiation

9.2 Computer Package Development

Fitting a potential requires many repeated calculations of the forces and total energy of configurations of hundreds of atoms using the potential as it is varied. This puts a high demand on memory and central processing unit of a computer. Python is an easy to use high-level programming language that supports object orientated programming. Unfortunately, parallel programming using threading is hampered in Python by the global interpreter lock, and by its very nature as an interpreted language, it is much slower than a compiled language.

There are several tools available to unlock the full potential of a modern multicore processor while writing in Python.

9.2.1 Cython

9.2.2 F2PY

9.2.3 OpenMP

9.2.4 Development

The resulting was developed using python and a shared object library written in Fortran 90 and compiled using F2PY.

Chapter 10

Molecular Dynamics

Chapter Summary

10.1 Activation by Ion Irradiation

10.2 Computer Package Development

|Intro;

Chapter 11

Conclusions

Chapter Summary

|Intro ↴

11.1 Restating Objectives

11.1.1 Contributions of this Thesis While Answering the Original Question

Radioactivity equation including source and sink terms Computer code that uses endf files and the equation

Chapter 12

Future Work

Chapter Summary

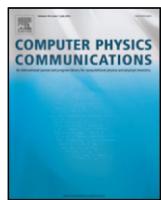
|Intro ↴

Appendices

Appendix A

Activity Paper published in Computer Physics Communications

A.1 Paper available through CPC



Activity computer program for calculating ion irradiation activation[☆]

Ben Palmer ^{*}, Brian Connolly, Mark Read

University of Birmingham, United Kingdom



ARTICLE INFO

Article history:

Received 1 June 2015

Received in revised form 15 February 2017

Accepted 19 February 2017

Available online 29 March 2017

ABSTRACT

A computer program, Activity, was developed to predict the activity and gamma lines of materials irradiated with an ion beam. It uses the TENDL (Koning and Rochman, 2012) [1] proton reaction cross section database, the Stopping and Range of Ions in Matter (SRIM) (Biersack et al., 2010) code, a Nuclear Data Services (NDS) radioactive decay database (Sonzogni, 2006) [2] and an ENDF gamma decay database (Herman and Chadwick, 2006) [3]. An extended version of Bateman's equation is used to calculate the activity at time t, and this equation is solved analytically, with the option to also solve by numeric inverse Laplace Transform as a failsafe. The program outputs the expected activity and gamma lines of the activated material.

Program summary

Program title: Activity

Catalogue identifier: AFBS_v1_0

Program summary URL: http://cpc.cs.qub.ac.uk/summaries/AFBS_v1_0.html

Program obtainable from: CPC Program Library, Queen's University, Belfast, N. Ireland

Licensing provisions: GNU GPL v3

No. of lines in distributed program, including test data, etc.: 688828

No. of bytes in distributed program, including test data, etc.: 71056048

Distribution format: tar.gz

Programming language: Fortran.

Computer: PCs or HPCs.

Operating system: Linux (tested on Debian).

Has the code been vectorized or parallelized?: OpenMPI

RAM: 250MB per process + 200MB overhead

Classification: 2.2, 17.8.

Nature of problem: To calculate the predicted activity of an ion irradiated target. The expected range of ion energies is between 1MeV and 200MeV; this is the range of the available ion cross section data.

Solution method: The program loads cross section data from the TENDL database and trajectory data from a SRIM [1] simulation xyz data file. It uses this data to calculate the production/loss rate of each isotope in the simulated target. Radioactive decay equations are used to calculate the amounts and activity of each radioactive isotope at the set time.

Running time: Typically the Activity program runs each input from seconds to no more than several minutes.

References:

- [1] SRIM — The stopping and range of ions in matter (2010). Ziegler, James F., Ziegler, M.D. and Biersack, J.P. 2010, Nuclear Instruments and Methods in Physics Research Section B, Vol. 268, pp. 1818–1823.

© 2017 Elsevier B.V. All rights reserved.

[☆] This paper and its associated computer program are available via the Computer Physics Communication homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author.

E-mail address: benpalmer1983@gmail.com (B. Palmer).

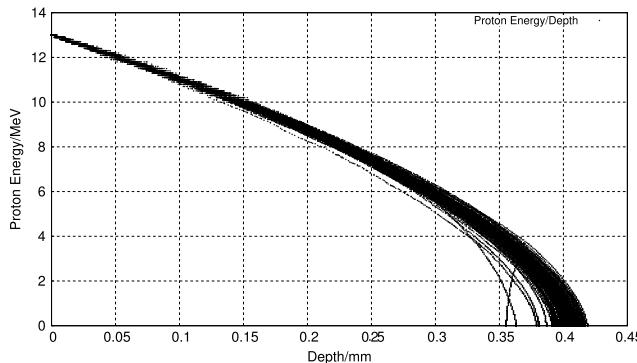


Fig. 1. One hundred simulated 13 MeV proton energy loss curves in Fe simulated with SRIM [4].

1. Background

1.1. Ion irradiation at the University of Birmingham

The Scanditronix MC-40 Cyclotron is used at the University of Birmingham to create a beam of protons or other light ions. The energies of these ions are typically between 10 MeV and 60 MeV with beam currents ranging up to 50 μA (3.1×10^{14} protons per second). Target materials are irradiated by this cyclotron for a number of reasons, including purposely creating radioactive isotopes for the nearby Queen Elizabeth Hospital, investigating ion irradiation damage and emulating neutron irradiation.

The cyclotron is usually used to create radioactive isotopes for medical use, but an additional beam line has been devoted to material science investigations into radiation damage. While the creation of radioactive isotopes is desired in some cases, material being tested for radiation damage should preferably have low levels of radioactivity.

It is expensive to arrange the irradiation of target materials by high energy neutrons sources, whereas it is relatively inexpensive to irradiate using an ion beam on the MC-40 Cyclotron. The energies can be controlled, and a set dose at a single energy, or a range of energies, can be precisely deposited into the target material.

The Activity code discussed here was developed to calculate the activity of a target material irradiated by a proton beam. It has been developed in Fortran and uses data from the TENDL-2013 proton cross section database, SRIM ion transport code and NDS radioactive decay database.

1.2. Simulating ion irradiation with SRIM

A package of ion transport codes, SRIM, is freely available to download and use to investigate the transport of ions through matter. SRIM uses the binary collision approximation (BCA) to simulate the passage of ions in a material. It is an approximate method, and one key restriction is that it does not take into account the structure of the material, and this approximation is therefore also imposed on the Activity code.

One file that SRIM creates is of importance to the Activity code, and that is the trajectory file that contains the energy and x , y , z co-ordinate data points for simulated ions moving through matter. Fig. 1 shows the trajectory of one hundred 13 MeV protons entering and passing through an Iron target, and it is this set of data points (together with the cross section database) that the Activity code uses to calculate the reaction rates for the transmutation of nuclei in the target. At higher energies, the ions slow as they lose energy due to electronic stopping, but as the ion energy drops the mechanism of loss through nuclear collisions becomes important.

The spreading of ion depths at lower energies is a result of the higher momentum transfer during nuclear collisions, as can be seen in Fig. 1.

1.3. Transmutation of nuclei by ion irradiation

Considering a simplified nuclear potential well, energetic protons approaching a nucleus may overcome the Coulomb potential barrier. They are captured by the nucleus and held within the potential well by the strong nuclear force. This process may leave the nucleus in an excited and unstable state, depending on the input energy of the proton and configuration of nucleons. The process is probabilistic, and the average chance of a reaction (the microscopic cross section) may be measured as a function of the projectile, projectile energy and target, either experimentally or by optical model potential calculations. The reaction rate is calculated from the microscopic cross section using the following equation:

$$R = \frac{J}{e} n_t \sigma \cdot 10^{-28} \delta t \quad (1)$$

- R Reaction Rate (reactions per second)
- J Beam current (A)
- n_t Number density of target (atoms per cubic meter)
- σ Microscopic reaction cross section (barns)
- e Elementary charge (1.602177E-19C)
- δt Target thickness (m).

1.4. Radioactive decay

Radioactive decay is the random change in nucleons or energy state of an unstable nucleus. It is impossible to predict when a single nucleus will decay, but the decay of a collection of nuclei is statistical in nature. The radioactivity and number of unstable nuclei at time t can be predicted using the decay constant, λ , for the radioactive isotope. This constant is defined as follows:

$$\lambda = -\frac{N'(t)}{N(t)}. \quad (2)$$

The number of radioactive nuclei $N(t)$ at time t is given by the following equation, where $N(0)$ is the starting number of nuclei:

$$N(t) = N(0) \exp(-t\lambda). \quad (3)$$

The activity $A(t)$ of the radioactive nuclei is predicted at time t by using the following equations, where $N'(t)$ is the change in amount of nuclei with respect to time:

$$A(t) = -N'(t) = \lambda N(t) \quad (4)$$

$$A(t) = \lambda N(0) \exp(-t\lambda). \quad (5)$$

1.5. Bateman equation for radioactive decay

The English mathematician Harry Bateman derived an Eq. (6) to calculate the amount of each isotope in a decay chain, illustrated in Fig. 2, at time t .

$$N_n(t) = \sum_{i=1}^{i=n} \left(\left(\prod_{j=i}^{j=n-1} \lambda_{(ij+1)} \right) \sum_{j=i}^{j=n} \left(\frac{N_{i0} \exp(-\lambda_j t)}{\prod_{p=i, p \neq j}^{p=n} (\lambda_p - \lambda_j)} \right) \right). \quad (6)$$

When a radioactive isotope decays, there may be more than one mode of decay, and this leads to branching factors. Pb-214 only decays via beta decay to Bi-214, giving a branching factor of 1.0, whereas Bi-214 has a 99.979% chance of decaying to Po-214 by beta decay and a 0.021% of emitting an alpha particle and decaying to Tl-210 (branching factors of 0.99979 and 0.00021 respectively) [5].

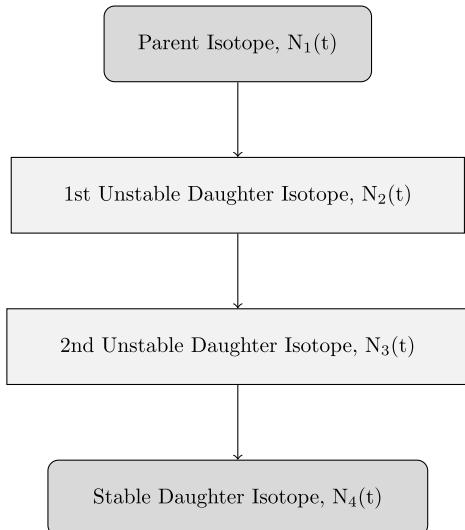


Fig. 2. An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.

When a target material is irradiated, there is a source term for transmuted nuclei due to the irradiation. The daughter isotopes of these transmuted isotopes will also be affected by the irradiation and will transmute further, giving a source term for each daughter isotope as a result of the irradiation. Sources for each isotope in the decay chain, and branching factors between a parent isotope and its daughter isotope/s must be accounted for.

The Bateman equation was derived using Laplace transforms, and this same method has been used to develop a modified equation that incorporates branching factors and production rates for each isotope in the decay chain, as illustrated by Fig. 3.

1.6. Laplace transform

Laplace Transforms (7) are a useful mathematical tool, and allow ordinary differential equations to be solved by simple algebraic manipulation in the s domain. Bateman took advantage of Laplace Transforms in deriving his equation, and this is the method that has been taken here as well.

$$F(s) = \int_0^\infty f(t) \exp(-st) dt. \quad (7)$$

1.7. Constructing the differential equations

The first step is to set up differential equations for the parent isotope, unstable daughter isotopes and stable daughter isotope. The parent isotope has a source term, due to production, and a loss term, due to decay. The unstable daughter isotopes have two source terms, from the production by irradiation induced transmutation and the decay of preceding isotopes in the decay chain, and a loss term, due to decay. Finally, the stable daughter that finalizes the decay chain has two source terms (the same as the unstable daughters) but no loss term.

The variables (and vectors) used in these equations are defined as follows:

- $\vec{\lambda}$ vector containing isotope decay constants λ_i
- \vec{b} vector containing isotope to isotope branching factors b_i
- \vec{w} vector containing isotope production rates w_i
- t time at which activity/amount of isotope is measured
- $N_i(0)$ starting amount of the i th isotope

- $N_i(t)$ amount of the i th isotope at time t
- $N'_i(t)$ change in amount of the i th isotope, with respect to time, at time t .

The differential equations for the parent isotope (first isotope), unstable daughter isotopes (i th isotopes) and stable, final, daughter isotope (z th isotope) in the time domain are as follows:

$$N'_1(t) = \omega_1 - \lambda_1 N_1(t) \quad (8)$$

$$N'_i(t) = \omega_i + b_{i-1} \lambda_{i-1} N_{i-1}(t) - \lambda_i N_i(t) \quad (9)$$

$$N'_z(t) = \omega_z + b_{z-1} \lambda_{z-1} N_{z-1}(t). \quad (10)$$

Applying the Laplace Transform to these three differential equations allows them to be manipulated and solved algebraically in the s-domain.

$$N_1(s) = \frac{1}{s + \lambda_1} N_1(0) + \frac{1}{s(s + \lambda_1)} \omega_1 \quad (11)$$

$$N_i(s) = \frac{1}{s(s + \lambda_i)} (\omega_i) + \frac{1}{s + \lambda_i} (b_{i-1} \lambda_{i-1} N_{i-1}(s)) \\ + \frac{1}{s + \lambda_i} N_i(0) \quad (12)$$

$$N_z(s) = \frac{1}{s^2} \omega_z + \frac{1}{s} b_{z-1} \lambda_{z-1} N_{z-1}(s) + \frac{1}{s} N_z(0). \quad (13)$$

1.8. Numerical inversion of the Laplace Transform

The Gaver–Stehfest [6] algorithm was developed in the 1960s and 1970s and is a method of calculating the inverse of a Laplace Transform in the real number domain. It is an easy to implement and reasonably accurate method, although it is an approximation to the real value. A comparison between an analytic and numeric inversion for the unstable isotope Po-218 is discussed at the end of this section (Fig. 4).

$$f(t) \approx f_n(t) = \frac{\ln(2)}{t} \sum_{k=1}^{2n} a_k(n) F(s) \text{ where } n \geq 1, t > 0 \quad (14)$$

$$s = \frac{k \ln(2)}{t} \quad (15)$$

$$a_k(n) = \frac{(-1)^{(n+k)}}{n!} \sum_{j=\text{Floor}(\frac{k+1}{2})} j^{n+1} \binom{n}{j} \binom{2j}{j} \binom{j}{k-j}. \quad (16)$$

The equation for the i th isotope may be calculated by recursively calculating the equations by numeric inversion, starting from the first (parent isotope) and inserting the result into each subsequent recursion until the i th isotope is reached (changing the equations appropriately for the parent, unstable daughter and stable daughter isotopes).

1.9. Analytic solution by partial fraction expansion

The equation for the i th isotope in the s domain can be written in full by substituting the preceding equation until the parent isotope is reached, and this full equation may be rearranged with the production amount of each isotope and starting amount of each isotope in individual terms. Each of these terms is multiplied by a fraction that can be expanded, using partial fractions, and inverted analytically.

This is illustrated with an example unstable isotope, fourth in the decay chain (including the parent isotope):

$$N_4(s) = \frac{1}{(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0)$$

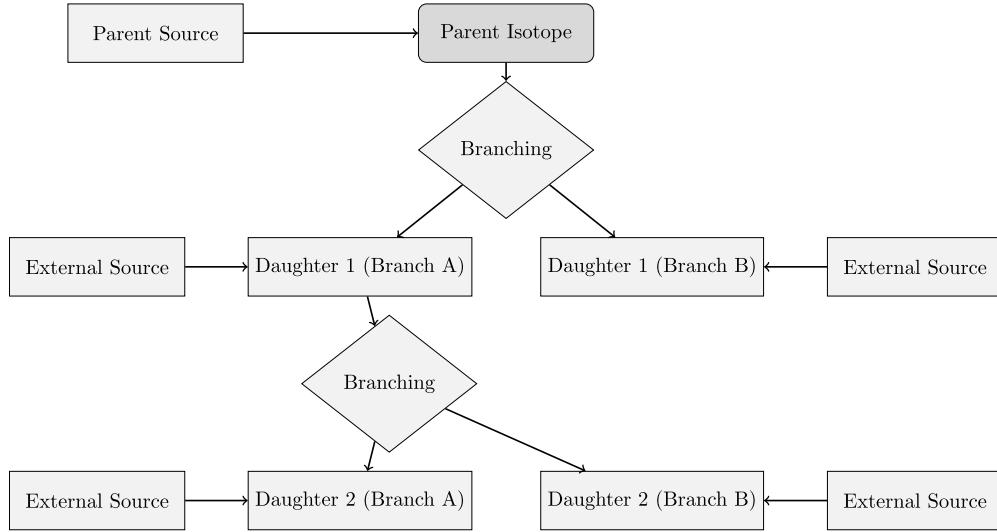


Fig. 3. An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

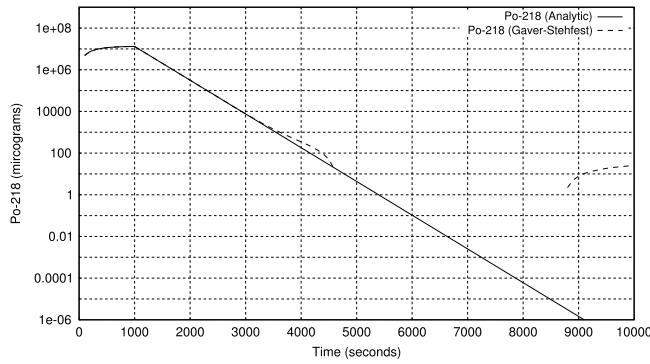


Fig. 4. Decay of Po-218: Analytic and Gaver–Stehfest Calculation [5].

$$\begin{aligned}
 & + \frac{1}{(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\
 & + \frac{1}{(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 N_3(0) + \frac{1}{(s + \lambda_4)} N_4(0) \\
 & + \frac{1}{s(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\
 & + \frac{1}{s(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 \omega_2 \\
 & + \frac{1}{s(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 \omega_3 + \frac{1}{s(s + \lambda_4)} \omega_4. \quad (17)
 \end{aligned}$$

An example stable isotope, fourth (last) in the decay chain (including the parent isotope):

$$\begin{aligned}
 N_4(s) = & \frac{1}{s(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0) \\
 & + \frac{1}{s(s + \lambda_2)(s + \lambda_3)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\
 & + \frac{1}{s(s + \lambda_3)} b_4 \lambda_3 N_3(0) + N_4(0) \\
 & + \frac{1}{s^2(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\
 & + \frac{1}{s^2(s + \lambda_2)(s + \lambda_3)} b_3 b_4 \lambda_2 \lambda_3 \omega_2
 \end{aligned}$$

$$+ \frac{1}{s^2(s + \lambda_3)} b_4 \lambda_3 \omega_3 + \frac{1}{s^2} \omega_4. \quad (18)$$

By using partial fraction expansion and standard Laplace Transforms, the set of equations below is used to calculate the amount of the m th isotope in the decay chain, providing the m th isotope is unstable.

$$\begin{aligned}
 N_m(t; \vec{\lambda}, \vec{b}, \vec{w}) \\
 = \sum_{k=1,m} r(k; \vec{\lambda}, \vec{b}) [f(t; k, m, \vec{\lambda}) N_k(0) + g(t; k, m, \vec{\lambda}) w_k] \quad (19)
 \end{aligned}$$

$$r(k, m, \vec{\lambda}) = \begin{cases} \prod_{i=k,m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \quad (20)$$

$$\begin{aligned}
 f(t; k, m, \vec{\lambda}) \\
 = (-1)^{m-k} \sum_{i=k,m} \left[\exp(-\lambda_i t) \prod_{j=k,m;j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (21)
 \end{aligned}$$

$$\begin{aligned}
 g(t; k, m, \vec{\lambda}) \\
 = \frac{1}{\prod_{i=k,m} \lambda_i} + (-1)^{m-k+1} \\
 \times \sum_{i=k,m} \left[\frac{1}{\lambda_i} \exp(-\lambda_i t) \prod_{j=k,m;j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right]. \quad (22)
 \end{aligned}$$

The set of equations below is used to calculate the amount of the m th isotope in the decay chain, where the m th isotope is stable.

$$\begin{aligned}
 N_m(t; \vec{\lambda}, \vec{b}, \vec{w}) = N_m + w_m t + \sum_{k=1,m-1} r(k; \vec{\lambda}, \vec{b}) \\
 \times [f(t; k, m-1, \vec{\lambda}) N_k(0) + g(t; k, m, \vec{\lambda}) w_k] \quad (23)
 \end{aligned}$$

$$r(k, m, \vec{\lambda}) = \begin{cases} \prod_{i=k,m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \quad (24)$$

$$\begin{aligned}
 f(t; k, m, \vec{\lambda}) \\
 = \frac{1}{\prod_{i=k,m} \lambda_i} + (-1)^{m-k+1} \\
 \times \sum_{i=k,m} \left[\frac{1}{\lambda_i} \exp(-\lambda_i t) \prod_{j=k,m;j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (25)
 \end{aligned}$$

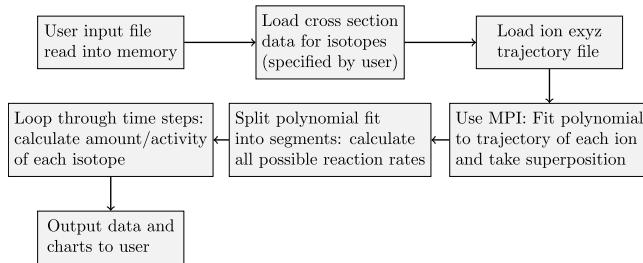


Fig. 5. Flow chart of major processes in the Activity code.

$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k,m} \lambda_i} t + \frac{\sum_{i=k,m} \left[\prod_{j=k,m;j \neq i} \lambda_j \right]}{\prod_{i=k,m} \lambda_i^2} + (-1)^{m-k+1} \sum_{i=k,m} \left[\frac{1}{\lambda_i^2} \exp(-\lambda_i t) \prod_{j=k,m;j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right]. \quad (26)$$

1.10. Preference: analytic over numeric

The numeric solution only requires the equation to be solved in the s-domain; the Gaver–Stehfest algorithm performs the inversion. It is worth the extra effort to derive and implement an analytic solution, as the numeric is only an approximation. Examples of the pitfalls of the numeric solution are that it can give negative amounts of an isotope and the difference between the numeric and analytic calculated amounts can become quite large when the isotope decays away to a very small value. Fig. 4 shows the predicted decay of a sample of Po-218 irradiated for 1000 s, and sampled until 10,000 s. In the region between 4000 s and 9000 s the amount from the numeric calculation drops below zero, whereas the analytic calculation remains above zero, as would be expected.

2. Computational methods

The Activity program has been developed in Fortran and takes advantage of MPI (Message Passing Interface) to speed up calculation times by allowing the use of multiple processes in parallel. It has a self contained maths library, although this could be improved in the future by using optimized maths libraries for certain functions (e.g. matrix operations).

The code was developed on a Debian based distribution of Linux, but it should be supported on other variants of Linux and Unix, and does not require any specialist hardware.

The user is required to prepare an input file that contains the instructions required to perform a calculation. In addition to the input file, the user must provide an EXYZ ion trajectory file output by SRIM. Activity will read in the user input file, and the SRIM and data files listed within, before performing the calculation. Fig. 5 shows a flowchart of the major steps the code performs.

There are various settings in the user input file, but the main ones relating to the simulated experiment are:

- Element composition of target (percentage by mass).
- Beam flux (current), energy, duration and area on target.
- Activity measurement time (end of the “experiment”).
- Material density.
- Target thickness.

Several data files are generated by Activity and, if the user has matplotlib [7], charts will be created too. The most relevant to the user are:

- gammaLines.dat – tally into discrete bins of predicted gamma counts.
- ionTraj.dat – the averaged ion trajectory used in the calculation.
- isotopeActivityFileG.dat – a large data file detailing the activity of every predicted radioactive isotope in the target at user specified times following irradiation.

The charts include:

- activityTop5.png – activity of the top 5 active isotopes as a function of time after irradiation starts.
- gammaLines.png – predicted gamma spectrum expected at the “experiment end time”.

The Activity code uses the equations derived above to calculate the amount and activity of each isotope in the calculation. One problem with the original Bateman equation that also exists in the set of modified Bateman equations is that two different isotopes with the same decay constant will cause a singularity and a halt in the calculation. The activity code loops through all the decay constants in use before it attempts to run the calculation. If any isotope decay constants match they are varied by a small amount relative to the decay constant. It repeats this process until all decay constants are unique before proceeding.

3. Approximations

The accuracy of the Activity code is dependent on the input files provided by the user and the method used to calculate the reaction rates and resulting activity. The TENDL proton database consists of experimental measured cross sections as well as values calculated using the optical model potential. Using the latest database is recommended.

SRIM uses the binary collision approximation to simulate ion transport. It is a well tested code that has been used for many years. One limitation is that the structure of the material is not taken into account. This would have an impact on a user of the Activity program if they were trying to calculate, for example, whether a FCC (face centered cubic) steel would be irradiated differently when compared to a BCC (body centered cubic) steel. The Activity code would determine the activity of the steel as a function of the ion current, ion type and the density, thickness and composition of the steel, not its structure.

This version of the Activity code averages the path of all the SRIM simulated ions, rather than treating each ion differently. This may or may not have an impact on the results. If a new version of the code is developed there would be an option to calculate reaction rates for each individual simulated ion, and a comparison could then be made to the calculations using the averaged path of a set of ions.

The final approximation would be to use the numeric solution to the activity equations, although the analytic solution is forced within the code unless it returns a failed result.

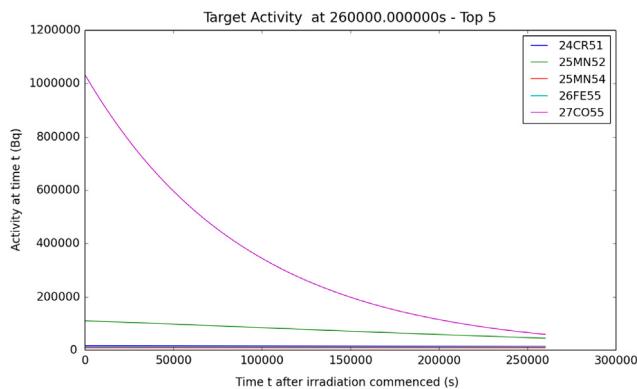
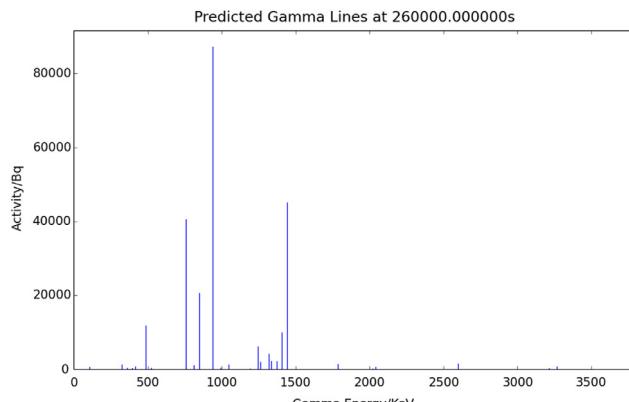
4. Results

A target of high purity Iron was irradiated with 36 MeV protons by the University of Birmingham Scanditronix MC-40 Cyclotron. The target was 0.5 mm thick and was irradiated at a current of 0.5 μ A for 300 s, irradiating approximately 0.25 g of Iron. A high purity Germanium detector was used to measure the gamma peaks three days after irradiation.

The peak that dominated the readings was the 931 keV Cobalt 55 line. After calibrating the detector and adjusting the readings, this peak was measured at 44,300 Bq+/−2000 Bq. The activity of this peak as predicted by the Activity code was 44,565 Bq.

Table 1

Gamma energy (keV)	Predicted activity (Bq)	Experimental activity (Bq)
766	4.45E6	5.11E5+/-2.5E4
778	6.14E6	1.36E6+/-6.8E4
812	5.04E6	1.15E6+/-5.8E4
850	6.00E6	1.39E6+/-7.0E4
126	9.33E5	2.10E5+/-1.1E4

**Fig. 6.** Sample Activity code output chart for the top five most active isotopes for Iron irradiated by 36 MeV protons.**Fig. 7.** Sample Activity code output chart for the expected gamma lines to be measured for Iron irradiated by 36 MeV protons.

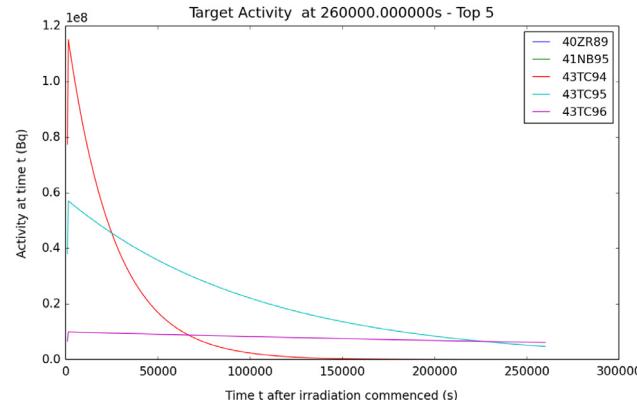
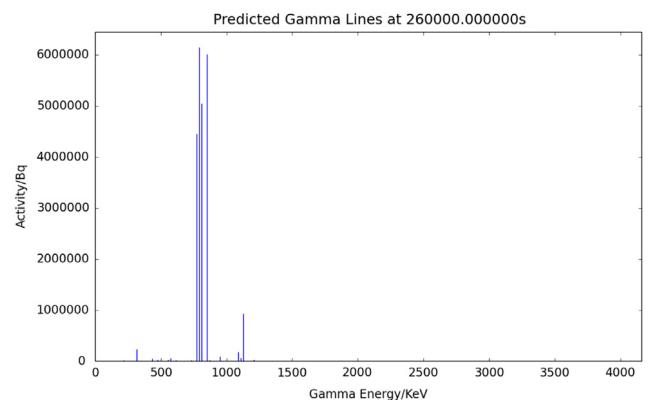
A target of high purity Molybdenum was irradiated with 13 MeV protons by the University of Birmingham Scanditronix MC-40 Cyclotron. The target was 0.5 mm thick and was irradiated at a current of 5 μ A for 1500 s, irradiating approximately 0.3 g of Molybdenum. A high purity Germanium detector was used to measure the gamma peaks three days after irradiation.

Five peaks were of interest, and these are listed in Table 1.

There was the possibility of the high purity Germanium detector introducing errors due to detector dead time, where by a source that is too active floods the device with gammas. The samples were safe to handle, and did not appear to flood the detector in any way, but there was still the possibility of counts being missed due to dead time. The probabilistic nature of radioactive decay also introduced inherent errors to the experimental activity measurements (see Figs. 6–9).

5. Conclusions

The Activity program is an easy to use Fortran compiled executable that can be built and run, for free, on a Linux computer. It

**Fig. 8.** Sample Activity code output chart for the top five most active isotopes for Molybdenum irradiated by 13 MeV protons.**Fig. 9.** Sample Activity code output chart for the expected gamma lines to be measured for Molybdenum irradiated by 13 MeV protons.

takes advantage of multi-core processors, and typical calculations take from seconds to minutes.

Using the SRIM code and TENDL database, the code has been used to predict the activity of Iron and Molybdenum targets that have been irradiated with a proton beam. The prediction of the 931 (keV) Cobalt 55 gamma activity of irradiated Iron was very close to the measured value. The five predicted gamma activities for ion irradiated Molybdenum were up to a factor of 5–10 away from the measured activities.

There are a number of improvements that would be considered in a future version of the Activity code. These improvements would include the following:

- A more readable output with less information printed, as too much may confuse the user.
- A file containing the top five radioactive isotopes with their top five gamma lines.
- Individual ion trajectories used to calculate the reaction rates, rather than the average path.
- Experimental ion activation data for a wider range of elements to test the Activity code against.
- Expand to include deuterons (this would require the TENDL deuteron cross section database).

Acknowledgments

We would like to thank and acknowledge the input and advice of the following:

- Dr Chris Cooper and John Hewett for irradiation activation data points.

- The University of Birmingham for providing the funding for this project.

References

- [1] A.J. Koning, D. Rochman, *Nucl. Data Sheets* 113 (2012).
- [2] A.A. Sonzogni, Nuclear Data Services. www-nds.iaea.org. 2006. URL: <http://www-nds.iaea.org/ndspub/download-endf/ENDF-B-VII.0/decay-index.htm> (visited on 08/14/2015).
- [3] M. Herman, M.B. Chadwick, *Nucl. Data Sheets* 107 (2006) 2931.
- [4] J.P. Biersack, J.F. Ziegler, M.D. Ziegler, *Nuclear Instrum. Methods Phys. Res. B* 268 (2010) 1818–1823.
- [5] P. Blaise, M. Coste, A. Courcelle, T.D. Huynh, C. Jouanne, P. Leconte, O. Litaize, S. Mengelle, G. Nogure, J.-M. Ruggiri, O. Srot, J. Tommasi, C. Vaglio, J.-F. Vidal, A. Santamarina, D. Bernard, The JEFF-3.1.1 Nuclear Data Library. 2009. ISBN: 978-92-64-99074-6.
- [6] H. Stehfest, *Commun. ACM* 13 (1970) 47–49.
- [7] J.D. Hunter, *Comput. Sci. Eng.* 9 (2007) 90–95.

A.2 Accompanying Manual

UNIVERSITY OF BIRMINGHAM



Department of Metallurgy & Materials

Activity Manual

Contents

1	Background	4
1.1	Ion Irradiation	4
1.1.1	Ion Irradiation at the University of Birmingham	4
1.1.2	Simulating Ion Irradiation with SRIM	4
1.1.3	Transmutation of Nuclei by Ion Irradiation	5
1.2	Decay and Activity Equations	6
1.2.1	Radioactive Decay	6
1.2.2	Bateman Equation for Radioactive Decay	6
1.2.3	Laplace Transform	7
1.2.4	Constructing the Differential Equations	7
1.2.5	Numerical Inversion of the Laplace Transform	8
1.2.6	Analytic Solution by Partial Fraction Expansion	9
1.2.7	Preference: Analytic over Numeric	11
1.3	Computational Methods	12
1.3.1	Activity Code	12
1.3.2	Approximations	13
1.3.3	Results	13
2	Installation and Using the Activity Code	17
2.1	Getting Started	17
2.1.1	Prerequisites	17
2.2	Installing Activity	17
2.2.1	Download Source Code	17
2.2.2	Compile Source Code	18
2.3	Input File	18
2.4	Acknowledgements	21
Appendices		22

A Example Input File	23
A.1 Iron 36MeV Proton Beam	23
B Fortran 90 Code	25
B.1 Fortran 90 Implementation of Analytic Method	25

List of Figures

1.1	Proton energy loss in Fe simulated with SRIM [2]	5
1.2	An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.	6
1.3	An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.	7
1.4	Decay of Po-218: Analytic and Gaver-Stehfest Calculations [5]	11
1.5	Flow chart of major processes in the Activity code	12
1.6	Sample Activity code output chart for the top five most active isotopes for Iron irradiated by 36MeV protons.	14
1.7	Sample Activity code output chart for the expected gamma lines to be measured for Iron irradiated by 36MeV protons.	15
1.8	Sample Activity code output chart for the top five most active isotopes for Molybdenum irradiated by 13MeV protons.	15
1.9	Sample Activity code output chart for the expected gamma lines to be measured for Molybdenum irradiated by 13MeV protons.	16

Chapter 1

Background

1.1 Ion Irradiation

A computer program, Activity, was developed to predict the activity and gamma lines of materials irradiated with an ion beam. It uses the TENDL[1] proton reaction cross section database, the Stopping and Range of Ions in Matter (SRIM)[2] code, a Nuclear Data Services (NDS) radioactive decay database [3] and an ENDF gamma decay database [4]. An extended version of Bateman's equation is used to calculate the activity at time t, and this equation is solved analytically, with the option to also solve by numeric inverse Laplace transform as a failsafe. The program outputs the expected activity and gamma lines of the activated material.

1.1.1 Ion Irradiation at the University of Birmingham

The Scanditronix MC-40 Cyclotron is used at the University of Birmingham to create a beam of protons or other light ions. The energies of these ions are typically between 10 MeV and 60 MeV with beam currents ranging up to 50 microamps (3.1×10^{14} protons per second). Target materials are irradiated by this cyclotron for a number of reasons, including purposely creating radioactive isotopes for the nearby Queen Elizabeth Hospital, investigating ion irradiation damage and emulating neutron irradiation.

The Cyclotron is usually used to create radioactive isotopes for medical use, but an additional beam line has been devoted to material science investigations into radiation damage. While the creation of radioactive isotopes is desired in some cases, material being tested for radiation damage should preferably have low levels of radioactivity.

It is expensive to arrange the irradiation of target materials by high energy neutrons sources, whereas it is relatively inexpensive to irradiate using an ion beam on the MC-40 Cyclotron. The energies can be controlled, and a set dose at a single energy, or a range of energies, can be precisely deposited in the target material.

The Activity code discussed here was developed to calculate the activity of a target material irradiated by a proton beam. It has been developed in Fortran and uses data from the TENDL-2013 proton cross section database, SRIM ion transport code and NDS radioactive decay database.

1.1.2 Simulating Ion Irradiation with SRIM

A package of ion transport codes, SRIM, is freely available to download and use to investigate the transport of ions through matter. SRIM uses the binary collision approximation (BCA) to simulate the passage of ions in a material. It is an approximate method, and one key restriction is that it does not take into account the structure of the material, and this approximation is therefore also imposed on the Activity code.

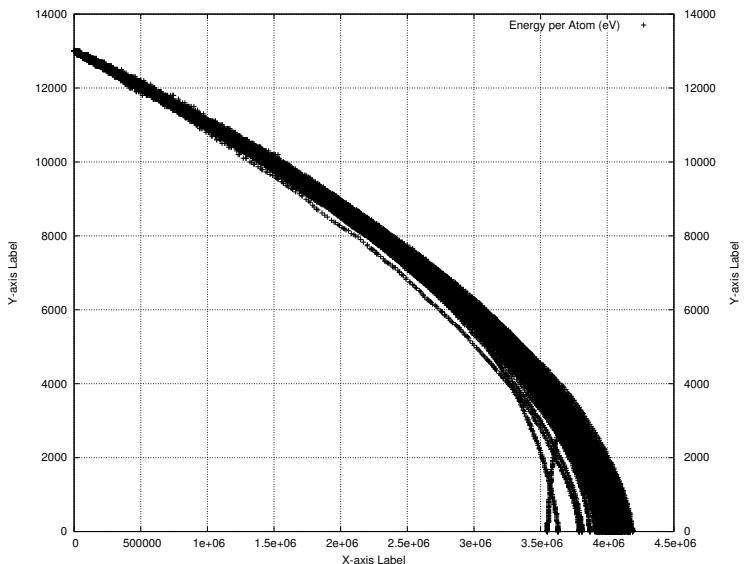


Figure 1.1: Proton energy loss in Fe simulated with SRIM [2]

One file that SRIM creates is of importance to the Activity code, and that is the trajectory file that contains the energy and x,y,z co-ordinate data points for simulated ions moving through matter. Figure 1.1 shows the trajectory of protons passing through an Iron target, and it is this set of data points (together with the cross section database) that the Activity code uses to calculate the reaction rates for the transmutation of nuclei in the target. At higher energies, the ions slow as they lose energy due to electronic stopping, but as the ion energy drops the mechanism of loss through nuclear collisions becomes important. The spreading of ion depths at lower energies is a result of the higher momentum transfer during nuclear collisions, as can be seen in Figure 1.1.

1.1.3 Transmutation of Nuclei by Ion Irradiation

Considering a simplified nuclear potential well, energetic protons approaching a nucleus may overcome the coulomb potential barrier. They are captured by the nucleus and held within the potential well by the strong nuclear force. This process may leave the nucleus in an excited and unstable state, depending on the input energy of the proton and configuration of nucleons. The process is probabilistic, and the average chance of a reaction (the microscopic cross section) may be measured as a function of the projectile, projectile energy and target, either experimentally or by optical model potential calculations. The reaction rate is calculated from the microscopic cross section using the following equation:

$$R = \frac{J}{e} n_t \sigma \cdot 10^{28} \delta t \quad (1.1)$$

- R Reaction Rate (reactions per second)
- J Beam current (A)
- n_t Number density of target (atoms per cubic metre)
- σ Microscopic reaction cross section (barns)
- e Elementary charge (1.602177E-19C)
- δT Target thickness (m)

1.2 Decay and Activity Equations

1.2.1 Radioactive Decay

Radioactive decay is the random change in nucleons or energy state of an unstable nucleus. It is impossible to predict when a nucleus will decay, but the decay of a collection of nuclei is statistical in nature. The radioactivity and number of unstable nuclei at time t can be predicted using the decay constant λ for the radioactive isotope. This constant is defined as follows:

$$\lambda = -\frac{N'(t)}{N(t)} \quad (1.2)$$

The number of radioactive nuclei $N(t)$ at time t is given by the following equation, where $N(0)$ is the starting number of nuclei:

$$N(t) = N(0) \exp(-t\lambda) \quad (1.3)$$

The activity $A(t)$ of the radioactive nuclei is predicted at time t by using the following equations, where $N'(t)$ is the change in amount of nuclei with respect to time:

$$A(t) = -N'(t) = \lambda N(t) \quad (1.4)$$

$$A(t) = \lambda N(0) \exp(-t\lambda) \quad (1.5)$$

1.2.2 Bateman Equation for Radioactive Decay

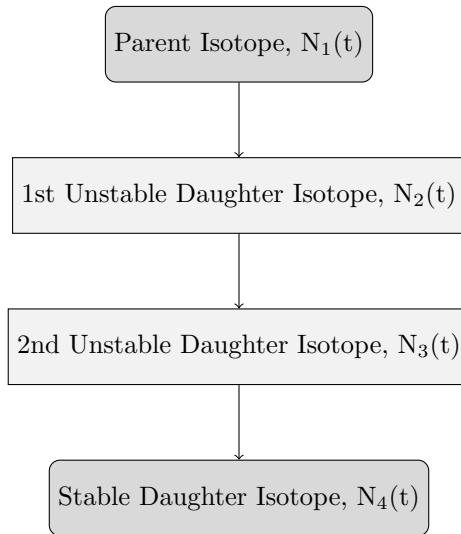


Figure 1.2: An example decay chain from an unstable parent isotope, through unstable daughter isotopes ending with a stable daughter isotope.

The English mathematician Harry Bateman derived an equation to calculate the amount of each isotope in a decay chain, illustrated in Figure 1.2, at time t .

$$N_n(t) = \sum_{i=1}^{i=n} \left(\left(\prod_{j=i}^{j=n-1} \lambda_{(ij+1)} \right) \sum_{j=i}^{j=n} \left(\frac{N_{i0} \exp(-\lambda_j t)}{\prod_{p=i, p \neq j}^{p=n} (\lambda_p - \lambda_j)} \right) \right) \quad (1.6)$$

When a radioactive isotope decays, there may be more than one mode of decay, and this leads to branching factors. Pb-214 only decays via beta decay to Bi-214, giving a branching factor of 1.0, whereas Bi-214 has a 99.979% chance of decaying to Po-214 by beta decay and a 0.021% of emitting an alpha particle and decaying to Tl-210 (branching factors of 0.99979 and 0.00021 respectively) [5].

When a target material is irradiated, there is a source term for transmuted nuclei due to the irradiation. The daughter isotopes of these transmuted isotopes will also be affected by the irradiation and will transmute further, giving a source term for each daughter isotope as a result of the irradiation. Sources for each isotope in the decay chain, and branching factors between a parent isotope and its daughter isotope/s must be accounted for.

The Bateman equation was derived using Laplace transforms, and this same method has been used to develop a modified equation that incorporates branching factors and production rates for each isotope in the decay chain, as illustrated by Figure 1.3.

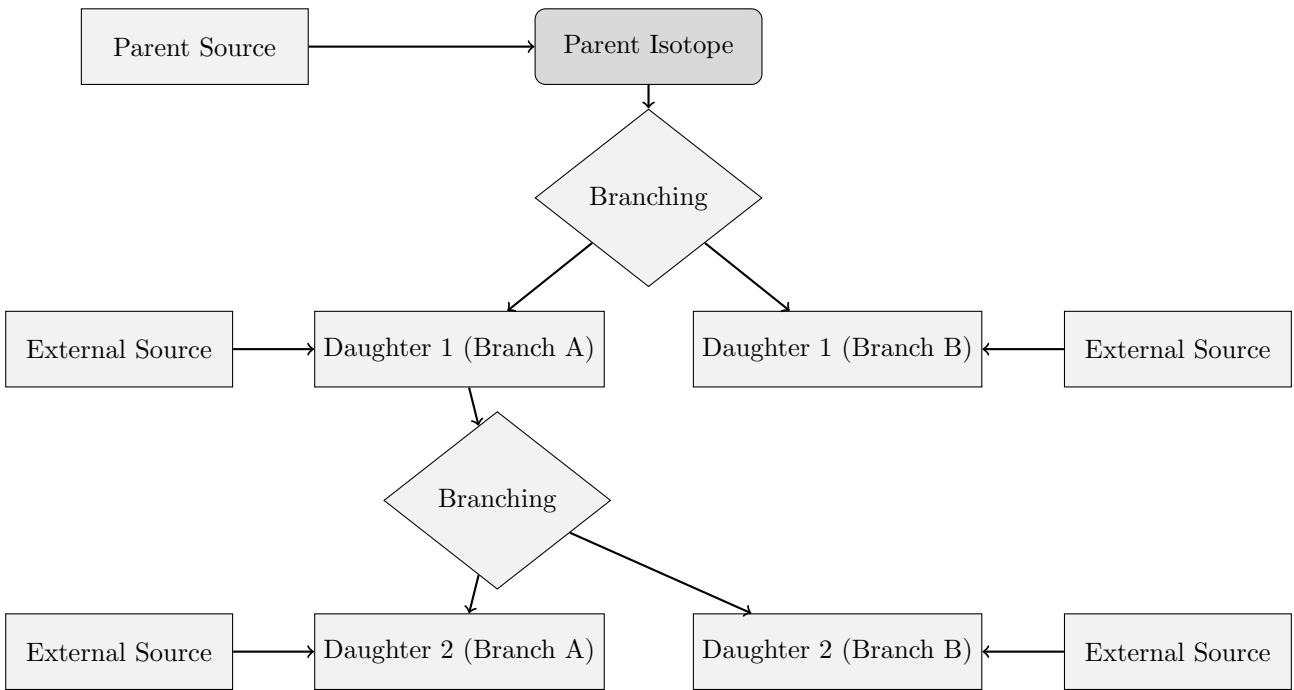


Figure 1.3: An example of several decay chains including branching factors and possible external source terms for each isotope on each chain.

1.2.3 Laplace Transform

Laplace Transforms are a useful mathematical tool, and allow ordinary differential equations to be solved by simple algebraic manipulation in the s domain. Bateman took advantage of Laplace Transforms in deriving his equation, and this is the method that has been taken here as well.

$$F(s) = \int_0^{\infty} f(t) \exp(-st) dt \quad (1.7)$$

1.2.4 Constructing the Differential Equations

The first step is to set up differential equations for the parent isotope, unstable daughter isotopes and stable daughter isotope. The parent isotope has a source term, due to production, and a loss term, due to decay. The unstable daughter isotopes have two source terms, from the production by irradiation induced transmutation

and the decay of preceding isotopes in the decay chain, and a loss term, due to decay. Finally, the stable daughter that finalizes the decay chain has two source terms (the same as the unstable daughters) but no loss term.

The variables (and vectors) used in these equations are defined as follows:

- $\vec{\lambda}$ vector containing isotope decay constants λ_i
- \vec{b} vector containing isotope to isotope branching factors b_i
- \vec{w} vector containing isotope production rates w_i
- t time at which activity/amount of isotope is measured
- $N_i(0)$ starting amount of the i th isotope
- $N_i(t)$ amount of the i th isotope at time t
- $N'_i(t)$ change in amount of the i th isotope, with respect to time, at time t

The differential equations for the parent isotope (first isotope), unstable daughter isotopes (i th isotopes) and stable, final, daughter isotope (z th isotope) in the time domain are as follows:

$$N'_1(t) = \omega_1 - \lambda_1 N_1(t) \quad (1.8)$$

$$N'_i(t) = \omega_i + b_{i-1} \lambda_{i-1} N_{i-1}(t) - \lambda_i N_i(t) \quad (1.9)$$

$$N'_z(t) = \omega_z + b_{z-1} \lambda_{z-1} N_{z-1}(t) \quad (1.10)$$

Applying the Laplace Transform to these three differential equations allows them to be manipulated and solved algebraically in the s -domain.

$$N_1(s) = \frac{1}{s + \lambda_1} N_1(0) + \frac{1}{s(s + \lambda_1)} \omega_1 \quad (1.11)$$

$$N_i(s) = \frac{1}{s(s + \lambda_i)} (\omega_i) + \frac{1}{s + \lambda_i} (b_{i-1} \lambda_{i-1} N_{i-1}(s)) + \frac{1}{s + \lambda_i} N_i(0) \quad (1.12)$$

$$N_z(s) = \frac{1}{s^2} \omega_z + \frac{1}{s} b_{z-1} \lambda_{z-1} N_{z-1}(s) + \frac{1}{s} N_z(0) \quad (1.13)$$

1.2.5 Numerical Inversion of the Laplace Transform

The Gaver-Stehfest[6] algorithm was developed in the 1960s and 1970s and is a method of calculating the inverse of a Laplace Transform in the real number domain. It is an easy to implement and reasonably accurate method, although it is an approximation to the real value. A comparison between an analytic and numeric inversion for the unstable isotope Po-218 is discussed at the end of this section.

$$f(t) \approx f_n(t) = \frac{\ln(2)}{t} \sum_{k=1}^{2n} a_k(n) F(s) \text{ where } n \geq 1, t > 0 \quad (1.14)$$

$$s = \frac{k \ln(2)}{t} \quad (1.15)$$

$$a_k(n) = \frac{(-1)^{(n+k)}}{n!} \sum_{j=\text{Floor}(\frac{k+1}{2})} j^{n+1} \binom{n}{j} \binom{2j}{j} \binom{j}{k-j} \quad (1.16)$$

The equation for the i th isotope may be calculated by recursively calculating the equations by numeric inversion, starting from the first (parent isotope) and inserting the result into each subsequent recursion until the i th isotope is reached (changing the equations appropriately for the parent, unstable daughter and stable daughter isotopes).

1.2.6 Analytic Solution by Partial Fraction Expansion

The equation for the i th isotope in the s domain can be written in full by substituting the preceding equation until the parent isotope is reached, and this full eqaution may be rearranged with the production amount of each isotope and starting amount of each isotope in individual terms. Each of these terms is multiplied by a fraction that can be expanded, using partial fractions, and inverted analytically.

This is illustrated with an example unstable isotope, fourth in the decay chain (including the parent isotope):

$$\begin{aligned} N_4(s) = & \frac{1}{(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0) \\ & + \frac{1}{(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\ & + \frac{1}{(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 N_3(0) \\ & + \frac{1}{(s + \lambda_4)} N_4(0) \\ & + \frac{1}{s(s + \lambda_1)(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\ & + \frac{1}{s(s + \lambda_2)(s + \lambda_3)(s + \lambda_4)} b_3 b_4 \lambda_2 \lambda_3 \omega_2 \\ & + \frac{1}{s(s + \lambda_3)(s + \lambda_4)} b_4 \lambda_3 \omega_3 \\ & + \frac{1}{s(s + \lambda_4)} \omega_4 \end{aligned} \quad (1.17)$$

An example stable isotope, fourth (last) in the decay chain (including the parent isotope):

$$\begin{aligned}
N_4(s) = & \frac{1}{s(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 N_1(0) \\
& + \frac{1}{s(s+\lambda_2)(s+\lambda_3)} b_3 b_4 \lambda_2 \lambda_3 N_2(0) \\
& + \frac{1}{s(s+\lambda_3)} b_4 \lambda_3 N_3(0) \\
& + N_4(0) \\
& + \frac{1}{s^2(s+\lambda_1)(s+\lambda_2)(s+\lambda_3)} b_2 b_3 b_4 \lambda_1 \lambda_2 \lambda_3 \omega_1 \\
& + \frac{1}{s^2(s+\lambda_2)(s+\lambda_3)} b_3 b_4 \lambda_2 \lambda_3 \omega_2 \\
& + \frac{1}{s^2(s+\lambda_3)} b_4 \lambda_3 \omega_3 \\
& + \frac{1}{s^2} \omega_4
\end{aligned} \tag{1.18}$$

By using partial fraction expansion and standard Laplace Transforms, the set of equations below is used to calculate the amount of the m th isotope in the decay chain, providing the m th isotope is unstable.

$$N_m(t; \vec{\lambda}, \vec{b}, \vec{w}) = \sum_{k=1,m} r(k; \vec{\lambda}, \vec{b}) [f(t; k, m, \vec{\lambda}) N_k(0) + g(t; k, m, \vec{\lambda}) w_k] \tag{1.19}$$

$$r(k, m, \vec{\lambda}) = \begin{cases} \prod_{i=k, m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \tag{1.20}$$

$$f(t; k, m, \vec{\lambda}) = (-1)^{m-k} \sum_{i=k, m} \left[\exp(-\lambda_i t) \prod_{j=k, m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \tag{1.21}$$

$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k, m} \lambda_i} + (-1)^{m-k+1} \sum_{i=k, m} \left[\frac{1}{\lambda_i} \exp(-\lambda_i t) \prod_{j=k, m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \tag{1.22}$$

The set of equations below is used to calculate the amount of the m th isotope in the decay chain, where the m th isotope is stable.

$$N_m(t; \vec{\lambda}, \vec{b}, \vec{w}) = N_m + w_m t + \sum_{k=1, m-1} r(k; \vec{\lambda}, \vec{b}) [f(t; k, m-1, \vec{\lambda}) N_k(0) + g(t; k, m, \vec{\lambda}) w_k] \tag{1.23}$$

$$r(k, m, \vec{\lambda}) = \begin{cases} \prod_{i=k, m-1} (b_{i+1} \lambda_i), & \text{if } k < m \\ 1, & \text{if } k = m \end{cases} \tag{1.24}$$

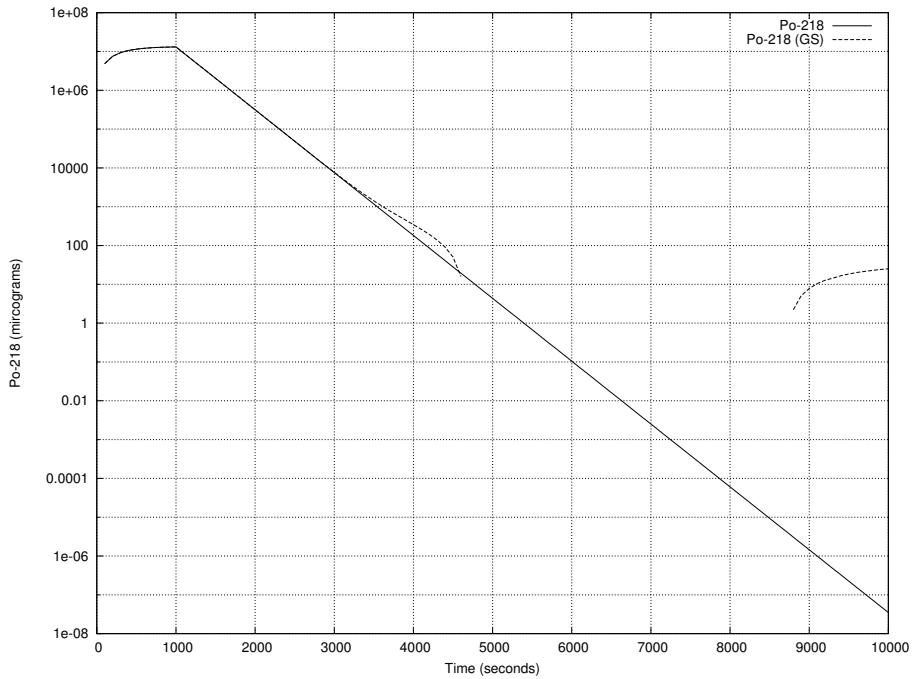


Figure 1.4: Decay of Po-218: Analytic and Gaver-Stehfest Calculations [5]

$$f(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k,m} \lambda_i} + (-1)^{m-k+1} \sum_{i=k,m} \left[\frac{1}{\lambda_i} \exp(-\lambda_i t) \prod_{j=k,m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (1.25)$$

$$g(t; k, m, \vec{\lambda}) = \frac{1}{\prod_{i=k,m} \lambda_i} t + \frac{\sum_{i=k,m} \left[\prod_{j=k,m; j \neq i} \lambda_j \right]}{\prod_{i=k,m} \lambda_i^2} + (-1)^{m-k+1} \sum_{i=k,m} \left[\frac{1}{\lambda_i^2} \exp(-\lambda_i t) \prod_{j=k,m; j \neq i} \left(\frac{1}{\lambda_i - \lambda_j} \right) \right] \quad (1.26)$$

1.2.7 Preference: Analytic over Numeric

The numeric solution only requires the equation to be solved in the s-domain; the Gaver-Stehfest algorithm performs the inversion. It is worth the extra effort to derive and implement an analytic solution, as the numeric is only an approximation. Examples of the pitfalls of the numeric solution are that it can give negative amounts of an isotope and the difference between the numeric and analytic calculated amounts can become quite large when the isotope decays away to a very small value. Figure 1.4 shows the predicted decay of a sample of Po-218 irradiated for 1,000s, and sampled until 10,000s. In the region between 4,000s and 9,000s the amount from the numeric calculation drops below zero, whereas the analytic calculation remains above zero, as would be expected.

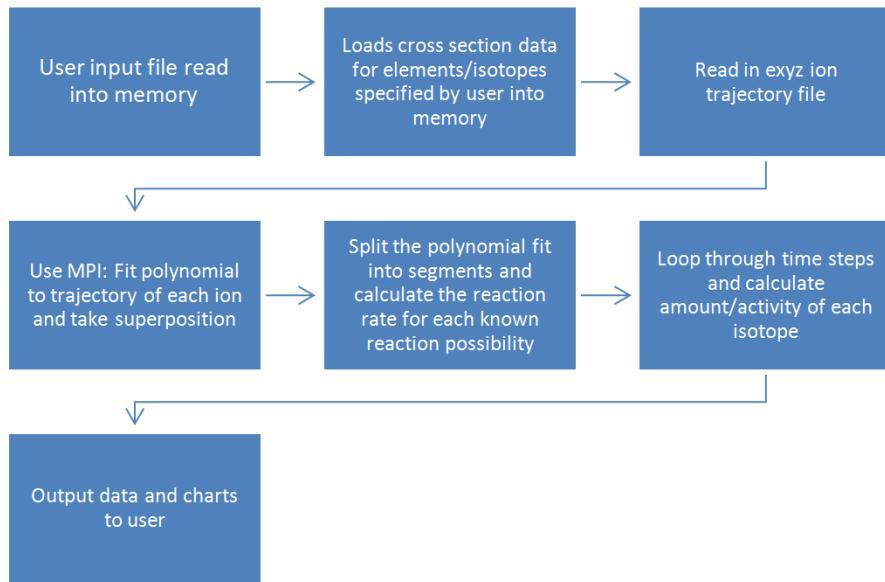


Figure 1.5: Flow chart of major processes in the Activity code

1.3 Computational Methods

1.3.1 Activity Code

The Activity program has been developed in Fortran and takes advantage of MPI (Message Parsing Interface) to speed up calculation times by allowing the use of multiple processes in parallel. It has a self contained maths library, although this could be improved in the future by using optimised maths libraries for certain functions (e.g. matrix operations).

The code was developed on a Debian version of Linux, but it should be supported on other variants of Linux and Unix, and does not require any specialist hardware.

The user is required to prepare an input file that contains the instructions required to perform a calculation. In addition to the input file, the user must provide an EXYZ ion trajectory file output by SRIM. Activity will read in the user input file, and the SRIM and data files listed within, before performing the calculation. Figure 1.5 shows a flowchart of the major steps the code performs.

There are various settings in the user input file, but the main ones relating to the simulated experiment are:

- element composition of target (percentage by mass)
- beam flux (current), energy, duration and area on target
- activity measurement time (end of the “experiment”)
- material density
- target thickness

Several data files are generated by Activity and, if the user has matplotlib [7], charts will be created too. The most relevant to the user are:

- gammaLines.dat - tally into discrete bins of predicted gamma counts

- ionTraj.dat - the averaged ion trajectory used in the calculation
- isotopeActivityFileG.dat - a large data file detailing the activity of every predicted radioactive isotope in the target at user specified times following irradiation

The charts include:

- activityTop5.png - activity of the top 5 active isotopes as a function of time after irradiation starts
- gammaLines.png - predicted gamma spectrum expected at the “experiment end time”

The Activity code uses the equations derived above to calculate the amount and activity of each isotope in the calculation. One problem with the original Bateman equation that also exists in the set of modified Bateman equations is that two different isotopes with the same decay constant will cause a singularity and a halt in the calculation. The activity code loops through all the decay constants in use before it attempts to run the calculation. If any isotope decay constants match they are varied by a small amount relative to the decay constant. It repeats this process until all decay constants are unique before proceeding.

1.3.2 Approximations

The accuracy of the Activity code is dependant on the input files provided by the user and the method used to calculate the reaction rates and resulting activity. The TENDL proton database consists of experimental measured cross sections as well as values calculated using the optical model potential. Using the latest database is recommended.

SRIM uses the binary collision approximation to simulate ion transport. It is a well tested code that has been used for many years. One limitation is that the structure of the material is not taken into account. This would have an impact on a user of the Activity program if they were trying to calculate, for example, whether a FCC (face centered cubic) steel would be irradiated differently when compared to a BCC (body centered cubic) steel. The Activity code would determine the activity of the steel as a function of the ion current, ion type and the density, thickness and composition of the steel, not its structure.

This version of the Activity code averages the path of all the SRIM simulated ions, rather than treating each ion differently. This may or may not have an impact on the results. If a new version of the code is developed there would be an option to calculate reaction rates for each individual simulated ion, and a comparison could then be made to the calculations using the averaged path of a set of ions.

The final approximation would be to use the numeric solution to the activity equations, although the analytic solution is forced within the code unless it returns a failed result.

1.3.3 Results

A target of high purity Iron was irradiated with 36 MeV protons by the University of Birmingham Scanditronix MC-40 Cyclotron. The target was 0.5mm thick and was irradiated at a current of 0.5 micro Amps for 300 seconds, irradiating approximately 0.25g of Iron. A high purity Germanium detector was used to measure the gamma peaks three days after irradiation.

The peak that dominated the readings was the 931 keV Cobalt 55 line. After calibrating the detector and adjusting the readings, this peak was measured at 44,300Bq +/- 2,000Bq. The activity of this peak as predicted by the Activity code was 44,565Bq.

A target of high purity Molybdenum was irradiated with 13 MeV protons by the University of Birmingham Scanditronix MC-40 Cyclotron. The target was 0.5mm thick and was irradiated at a current of 5 micro Amps

Table 1.1: Gamma peaks predicted and measured for a 13 MeV ion irradiated sample of Mo

Gamma Energy (keV)	Predicted Activity (Bq)	Experimental Activity (Bq)
766	4.45E6	5.11E5 +/- 2.5E4
778	6.14E6	1.36E6 +/- 6.8E4
812	5.04E6	1.15E6 +/- 5.8E4
850	6.00E6	1.39E6 +/- 7.0E4
126	9.33E5	2.10E5 +/- 1.1E4

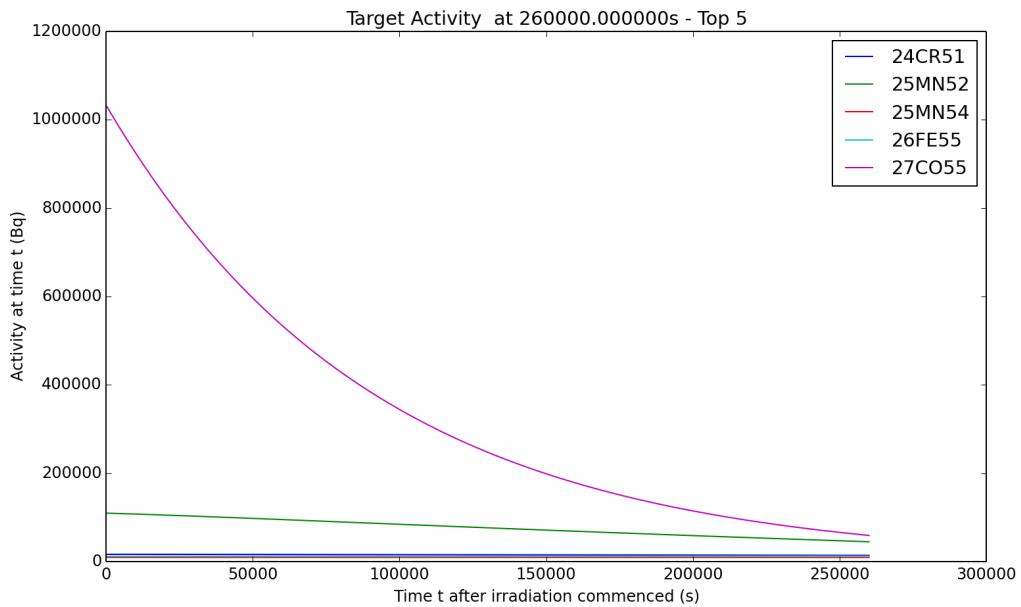


Figure 1.6: Sample Activity code output chart for the top five most active isotopes for Iron irradiated by 36MeV protons.

for 1500 seconds, irradiating approximately 0.3g of Molybdenum. A high purity Germanium detector was used to measure the gamma peaks three days after irradiation.

Five peaks were of interest, and these are listed in Table 1.1.

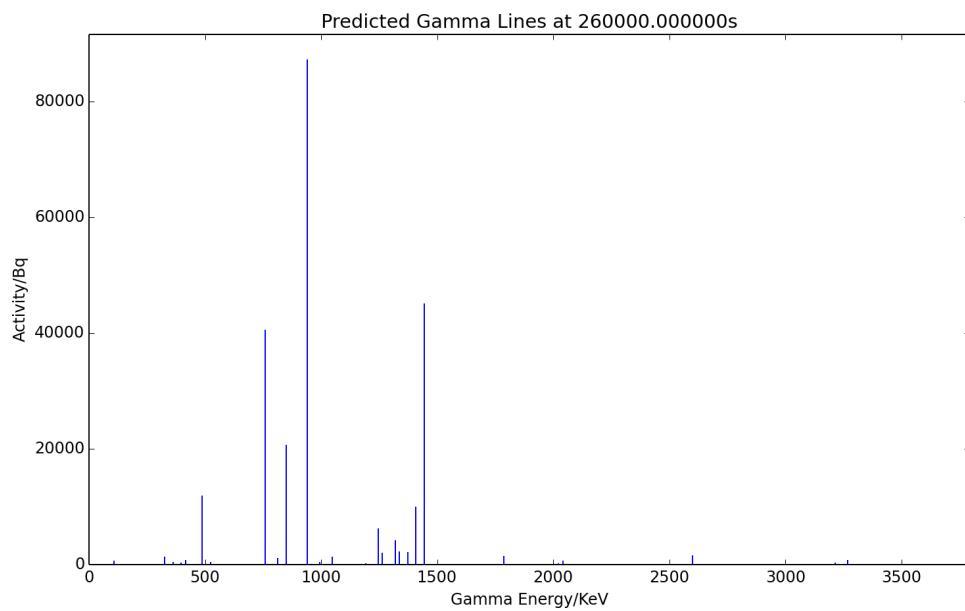


Figure 1.7: Sample Activity code output chart for the expected gamma lines to be measured for Iron irradiated by 36MeV protons.

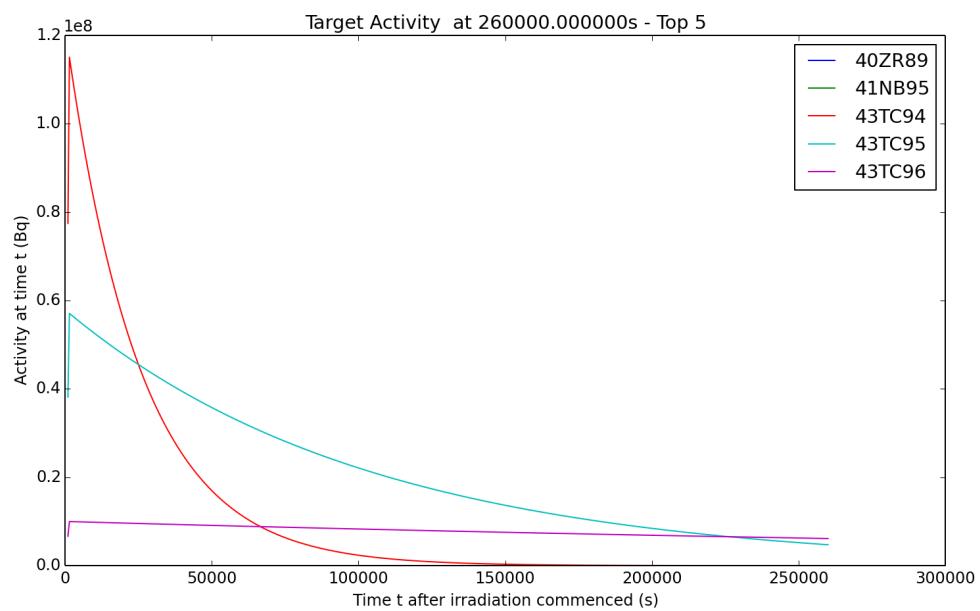


Figure 1.8: Sample Activity code output chart for the top five most active isotopes for Molybdenum irradiated by 13MeV protons.

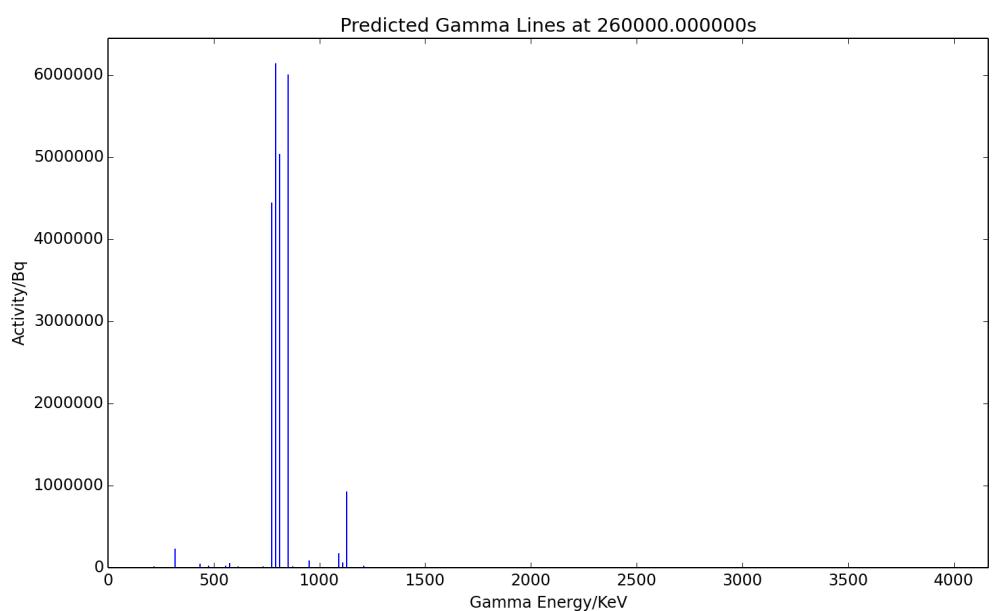


Figure 1.9: Sample Activity code output chart for the expected gamma lines to be measured for Molybdenum irradiated by 13MeV protons.

Chapter 2

Installation and Using the Activity Code

2.1 Getting Started

2.1.1 Prerequisites

This code was designed to run on Linux and has been developed and tested on the Debian based distribution Mint. It has not been tested on any other versions of Linux or Unix, nor has it been tested on cygwin.

The minimal requirements of the code are:

- 250MB per process + 200MB overhead
- 1GB Hard Drive space
- Ideally a modern multicore processor
- Fortran90
- OpenMPI

Optional:

- GNUPlot
- Python & MatPlotLib

2.2 Installing Activity

2.2.1 Download Source Code

The most recent source code is available for download from github. It may be downloaded and extracted using the terminal and the commands in listing 1.

Listing 2.1: bash

```
1 cd ~
2 mkdir -p ~/activity/tar
3 cd ~/activity/tar
4 wget https://github.com/BenPalmer1983/activity/raw/master/activity.tar.gz
5 tar -xzvf activity.tar.gz
```

2.2.2 Compile Source Code

Once the source files have been downloaded and extracted, run the install script then remove the downloaded files (if you need to free up space). The terminal commands are given in listing 2. Providing the prerequisites are available, the process will only take a few minutes (with a reasonable Internet connection).

Listing 2.2: bash

```
1 cd ~/activity/tar
2 ./install.sh #choose installation directory when prompted; default is ~/activity
3 rm -R ~/activity/tar
```

The activity code is now installed and the activity.x binary may be executed on the terminal. If the terminal fails to find the binary, the .bashrc or .bash_profile file must be edited, or a symlink created from the bin file to a valid and accessible bin directory.

2.3 Input File

The input file sets the target starting composition, the beam and simulated experiment settings as well as the paths to the required data files.

#elements

The elements keyword instructs the code to read in the composition of the target material. The elements and their percentage by weight are listed under the keyword.

Listing 2.3: Input File

```
1 #elements
2 Fe 72
3 Cr 18
4 Ni 8
5 Mg 2
```

#isotopes #decaymodes #gammaenergies #xsfiles

Four sets of data files are distributed with the code, and the paths to these files are defined underneath each keyword. N.B. the path for the xsfiles is to the directory that holds all the cross section data files, rather than each individual file.

Listing 2.4: Input File

```
1 #isotopes
2 "/home/ben/activity/data/isotopes.txt"
3 #decaymodes
4 "/home/ben/activity/data/decaymodes.txt"
```

```
5 #gammaenergies
6 "/home/ben/activity/data/gammaenergies.txt"
7 #xsfiles
8 "/home/ben/activity/data/xs"
```

```
#trajfile
```

The SRIM exyz file used for the calculation is pointed to under this keyword.

Listing 2.5: Input File

```
1 #trajfile
2 "/home/ben/activity/examples/Fe36MeV.exyz"
```

```
#polyfitorder #integrationgranularity
```

The code calculates the reaction rate of each projectile and target nucleus by first fitting a polynomial to the energy/depth data file. This polynomial gives an average path for the projectiles travelling through the target, allowing $E(x)$ to be calculated quickly. The order of the polynomial is selected here, and a 5th order shold give a reasonable fit. The integration granularity keyword determines how many sections the polynomial is split up into. The energy, cross section for each section and the beam settings are used to calculate the reaction rate for each section, and these are all summed up to give the overall reaction rate.

Listing 2.6: Input File

```
1 #polyfitorder
2 5
3 #integrationgranularity
4 10
```

```
#beamflux #beamenergy #beamduration #beamarea
```

These keywords are self explanatory and control the beam settings used in the calculation.

Listing 2.7: Input File

```
1 #beamflux
2 0.5 uA
3 #beamenergy
4 36 MeV
5 #beamduration
6 300 s
7 #beamarea
8 100 mm2
```

```
#amtime #timestep
```

The time at which the activity of the sample is measured is set by the #amtime keyword. The Activity code calculates the activity of each radioactive isotope in the calculation from the time the beam starts until the activity measured time; and the intervals at which these calculations are made is determined by the #timestep keyword.

Listing 2.8: Input File

```
1 #amtime
2 260000 s
```

```
3 #timestep
4 1000 s
```

#projectile

The current version only supports protons as the projectile, therefore the only possible atomic and mass number combination for a projectile is 1 1.

Listing 2.9: Input File

```
1 #projectile
2 1 1
```

#targetthickness #materialdensity

Both the target thickness and density of the material it is made from are input with these keywords.

Listing 2.10: Input File

```
1 #targetthickness
2 0.5 mm
3 #materialdensity
4 8000 kgm3
```

#vpi

This keyword is only being used for a feature currently under testing, so the keyword can be omitted from the input file.

Listing 2.11: Input File

```
1 #vpi
2 60.2
```

#individualisotopeactivity #verboseterminal

If the #individualisotopeactivity keyword is followed by yes; additional data files are output containing the activity of each individual isotope in the calculation. #verboseterminal followed by yes will output more verbosely to the terminal.

Listing 2.12: Input File

```
1 #individualisotopeactivity
2 yes
3 #verboseterminal
4 yes
```

#targetdpa

This keyword is only being used for a feature currently under testing, so the keyword can be omitted from the input file.

Listing 2.13: Input File

```
1 #targetdpa
2 0.0
```

```
#gammachartresolution
```

The gamma chart is created by tallying the gamma values into bins, and this figure specifies the resolution(number of bins) used to create the output chart.

Listing 2.14: Input File

```
1 #gammachartresolution  
2 200
```

2.4 Acknowledgements

We would like to thank and acknowledge the input and advice of the following:

- Dr Chris Cooper and John Hewett for irradiation activation data points.
- The University of Birmingham for providing the funding for this project.

Appendices

Appendix A

Example Input File

A.1 Iron 36MeV Proton Beam

Listing A.1: Fe36MeV.in

```
1 #elements
2 Fe 100
3 #isotopes
4 "/home/ben/activity/data/isotopes.txt"
5 #decaymodes
6 "/home/ben/activity/data/decaymodes.txt"
7 #gammaenergies
8 "/home/ben/activity/data/gammaenergies.txt"
9 #xsfiles
10 "/home/ben/activity/data/xs"
11 #trajfile
12 "/home/ben/activity/examples/Fe36MeV.exyz"
13 #polyfitorder
14 5
15 #integrationgranularity
16 10
17 #beamflux
18 0.5 uA
19 #beamenergy
20 36 MeV
21 #beanduration
22 300 s
23 #beamarea
24 100 mm2
25 #amtime
26 260000 s
27 #timestep
28 1000 s
29 #projectile
30 1 1
31 #targetthickness
32 0.5 mm
33 #materialdensity
34 8000 kgm3
```

```
35 #vpi
36 60.2
37 #individualisotopeactivity
38 yes
39 #verboseterminal
40 yes
41 #targetdpa
42 0.0
43 #gammachartresolution
44 200
```

Appendix B

Fortran 90 Code

B.1 Fortran 90 Implementation of Analytic Method

Listing B.1: Fortran 90

```
1 Type :: decayChainObj
2   Real(kind=DoubleReal) :: time = 0.0D0
3   !Real(kind=DoubleReal) :: productionRate = 0.0D0
4   Integer(kind=StandardInteger) :: isotopes
5   Character(len=16), Dimension(1:100) :: label
6   Real(kind=DoubleReal), Dimension(1:100) :: productionRate = 0.0D0
7   Real(kind=DoubleReal), Dimension(1:100) :: branchFactor = 1.0D0 ! from isotope parent
8   Real(kind=DoubleReal), Dimension(1:100) :: decayConstant = -1.0D0 ! negative for stable
9   Real(kind=DoubleReal), Dimension(1:100) :: halfLife = -1.0D0 ! negative for stable
10  Real(kind=DoubleReal), Dimension(1:100) :: amountStart = 0.0D0
11  Real(kind=DoubleReal), Dimension(1:100) :: amountEnd = 0.0D0
12 End Type
13
14 Subroutine CalcIsotopeChain(decayChain)
15 ! Uses inverse laplace transform to calculate isotope amounts at time t (after time = 0)
16 ! t time in seconds after t=0
17 ! w production rate of parent isotope
18 ! isotope chain data
19   Implicit None ! Force declaration of all variables
20 ! Vars In/Out
21   Type(decayChainObj) :: decayChain
22 ! Vars Private
23   Integer(kind=StandardInteger) :: i
24   Real(kind=DoubleReal) :: t, nEnd
25   Real(kind=DoubleReal), Dimension(1:100) :: W ! Production Rate
26   Real(kind=DoubleReal), Dimension(1:100) :: L ! Lambda
27   Real(kind=DoubleReal), Dimension(1:100) :: N ! Starting number of atoms
28   Real(kind=DoubleReal), Dimension(1:100) :: B ! Exp
29 ! Complete decay chain data
30   Call decayChainComplete(decayChain)
31 ! store input in shortned name arrays to make equations clearer
32   t = decayChain%time
33   Do i=1,decayChain%isotopes
34     W(i) = decayChain%productionRate(i)
```

```

35      L(i) = decayChain%decayConstant(i)
36      B(i) = decayChain%branchFactor(i)
37      N(i) = decayChain%amountStart(i)
38  End Do
39 ! Break infinities
40  Call DecayBreakInfinities(L,decayChain%isotopes)
41 ! Run analytic calculations
42 ! Loop through isotopes
43 ! Using L-1(1/(q+ps) = (1/p)*exp(-1*(q/p)*t) and partial fractions
44  Do i=1,decayChain%isotopes
45      nEnd = CalcIsotopeChainCalc(t,i,W,L,N,B)
46      decayChain%amountEnd(i) = nEnd
47  End Do
48 End Subroutine CalcIsotopeChain
49
50 Subroutine decayChainComplete(decayChain)
51 ! Completes the decay chain object
52 Implicit None ! Force declaration of all variables
53 ! Vars In/Out
54 Type(decayChainObj) :: decayChain
55 ! Vars Private
56 Integer(kind=StandardInteger) :: i, n
57 n = 0
58 Do i=1,100
59     n = n + 1
60     If(decayChain%decayConstant(i).eq.-1.0D0.and.decayChain%halfLife(i).gt.0.0D0)Then
61 ! complete decay constant from half life
62         decayChain%decayConstant(i) = lnTwo/decayChain%halfLife(i)
63     End If
64     If(decayChain%halfLife(i).le.0.0D0.and.decayChain%decayConstant(i).gt.0.0D0)Then
65 ! complete decay constant from half life
66         decayChain%halfLife(i) = lnTwo/decayChain%decayConstant(i)
67     End If
68 ! Adjust for stable isotope
69     If(decayChain%decayConstant(i).lt.0.0D0)Then
70         decayChain%halfLife(i) = -1.0D0
71         decayChain%decayConstant(i) = 0.0D0
72     End If
73     If(decayChain%halfLife(i).lt.0.0D0)Then
74         decayChain%halfLife(i) = -1.0D0
75         decayChain%decayConstant(i) = 0.0D0
76     End If
77 ! Break out if stable
78     If(decayChain%decayConstant(i).eq.0.0D0)Then
79         Exit
80     End If
81 End Do
82 decayChain%isotopes = n
83 End Subroutine decayChainComplete
84
85 Subroutine DecayBreakInfinities(L,n)
86 !
87 Implicit None ! Force declaration of all variables

```

```

88 ! Vars In/Out
89   Real(kind=DoubleReal), Dimension(:) :: L ! Lambda
90   Integer(kind=StandardInteger) :: n
91 ! Vars Private
92   Integer(kind=StandardInteger) :: i,j
93   Logical :: breaking
94 ! Loop and alter matching decay constants slightly
95   breaking = .true.
96   Do While(breaking)
97     breaking = .false.
98     Do i=1,n-1
99       Do j=i+1,n
100         If(L(i).eq.L(j))Then
101           breaking = .true.
102           L(i) = L(i)*1.0000001D0 ! Vary by 0.00001%
103         End If
104       End Do
105     End Do
106   End Do
107 End Subroutine DecayBreakInfinities
108
109 Function CalcIsotopeChainCalc(t,m,W,L,N,B) Result (nEnd)
110   Implicit None ! Force declaration of all variables
111 ! Vars In
112   Real(kind=DoubleReal) :: t
113   Integer(kind=StandardInteger) :: m
114   Real(kind=DoubleReal), Dimension(1:100) :: W ! Production Rate
115   Real(kind=DoubleReal), Dimension(1:100) :: L ! Lambda
116   Real(kind=DoubleReal), Dimension(1:100) :: N ! Starting number of atoms
117   Real(kind=DoubleReal), Dimension(1:100) :: B ! Branch factor
118 ! Vars Out
119   Real(kind=DoubleReal) :: nEnd
120 ! Vars Private
121   Integer(kind=StandardInteger) :: i, j, k, z
122   Real(kind=DoubleReal) :: mult, multR
123   Real(kind=DoubleReal) :: nChange
124 ! Init
125   nEnd = 0.0D0
126 ! -----
127 ! UNSTABLE Isotopes
128 ! -----
129   If(L(m).gt.0.0D0)Then
130 ! Loop through terms
131   Do k=1,m
132     multR = CalcIsotopeChainMultR(k,m,L,B)
133     mult = multR * N(k)
134   ! decay of starting matter
135     nChange = CalcIsotopeChainF_Unstable(k,m,t,mult,L)
136     nEnd = nEnd + nChange
137   ! production term
138     mult = multR * W(k)
139     nChange = CalcIsotopeChainG_Unstable(k,m,t,mult,L)
140     nEnd = nEnd + nChange

```

```

141      print *,k,nEnd
142      End Do
143      Else
144 ! -----
145 ! STABLE Isotopes
146 ! -----
147 ! Loop through terms
148     nEnd = N(m)+t*W(m)
149     Do k=1,m-1
150         multR = CalcIsotopeChainMultR(k,m,L,B)
151         mult = multR * N(k)
152 ! decay of starting matter
153         nChange = CalcIsotopeChainF_Stable(k,m,t,mult,L)
154         nEnd = nEnd + nChange
155 ! production term
156         mult = multR * W(k)
157         nChange = CalcIsotopeChainG_Stable(k,m,t,mult,L)
158         nEnd = nEnd + nChange
159         print *,k,nEnd
160     End Do
161     End If
162 End Function CalcIsotopeChainCalc
163
164 Function CalcIsotopeChainMultR(k,m,L,B) Result (multR)
165 ! Vars In
166     Integer(kind=StandardInteger) :: k, m
167     Real(kind=DoubleReal), Dimension(1:100) :: L ! Lambda
168     Real(kind=DoubleReal), Dimension(1:100) :: B ! Branching Factor
169 ! Vars Out
170     Real(kind=DoubleReal) :: multR
171 ! Private
172     Integer(kind=StandardInteger) :: i
173 ! Result
174     multR = 1.0D0
175     If(k.lt.m)Then
176         Do i=k,m-1
177             multR = multR * B(i+1) * L(i)
178         End Do
179     End If
180 End Function CalcIsotopeChainMultR
181
182 ! -----
183 ! Unstable Isotope Functions
184 ! -----
185
186 Function CalcIsotopeChainF_Unstable(k,m,t,mult,L) Result (nChange)
187 ! Vars In
188     Integer(kind=StandardInteger) :: k, m
189     Real(kind=DoubleReal) :: t, mult
190     Real(kind=DoubleReal), Dimension(1:100) :: L ! Lambda
191 ! Vars Out
192     Real(kind=DoubleReal) :: nChange
193 ! Private

```

```

194     Real(kind=DoubleReal) :: multP
195     Real(kind=DoubleReal) :: p, q, r, s
196     Integer(kind=StandardInteger) :: i, j
197 ! Calculate isotope amount change
198     nChange = 0.0D0
199     multP = (-1.0D0)**(m-k)
200 ! Loop through pfrac
201     Do i=k,m
202         r = 1.0D0
203         Do j=k,m
204             If(j.ne.i)Then
205                 r = r * (L(i)-L(j))
206             End If
207         End Do
208         nChange = nChange + (1.0D0/r)*exp(-1.0D0*L(i)*t)*multP*mult
209     End Do
210 End Function CalcIsotopeChainF_Unstable
211
212 Function CalcIsotopeChainG_Unstable(k,m,t,mult,L) Result (nChange)
213 ! Vars In
214     Integer(kind=StandardInteger) :: k, m
215     Real(kind=DoubleReal) :: t, mult
216     Real(kind=DoubleReal), Dimension(1:100) :: L ! Lambda
217 ! Vars Out
218     Real(kind=DoubleReal) :: nChange
219 ! Private
220     Real(kind=DoubleReal) :: multP
221     Real(kind=DoubleReal) :: p, q, r, s
222     Integer(kind=StandardInteger) :: i, j
223 ! Calculate isotope amount change
224     nChange = 0.0D0
225     multP = (-1.0D0)**(m-k+1)
226 ! term A
227     r = 1.0D0
228     Do i=k,m
229         r = r * L(i)
230     End Do
231     nChange = nChange + (1.0D0/r)*mult
232 ! Loop through pfrac
233     Do i=k,m
234         r = 1.0D0*L(i)
235         Do j=k,m
236             If(j.ne.i)Then
237                 r = r * (L(i)-L(j))
238             End If
239         End Do
240         nChange = nChange + (1.0D0/r)*exp(-1.0D0*L(i)*t)*multP*mult
241     End Do
242 End Function CalcIsotopeChainG_Unstable
243
244 ! -----
245 ! Stable Isotope Functions
246 ! -----

```

```

247
248 Function CalcIsotopeChainF_Stable(k,mIn,t,mult,L) Result (nChange)
249 ! Vars In
250   Integer(kind=StandardInteger) :: k, mIn
251   Real(kind=DoubleReal) :: t, mult
252   Real(kind=DoubleReal), Dimension(1:100) :: L ! Lambda
253 ! Vars Out
254   Real(kind=DoubleReal) :: nChange
255 ! Private
256   Integer(kind=StandardInteger) :: m
257   Real(kind=DoubleReal) :: multP
258   Real(kind=DoubleReal) :: p, q, r, s
259   Integer(kind=StandardInteger) :: i, j
260 ! In
261   m = mIn-1
262 ! Calculate isotope amount change
263   nChange = 0.0D0
264   multP = (-1.0D0)**(m-k+1)
265 ! term A
266   r = 1.0D0
267   Do i=k,m
268     r = r * L(i)
269   End Do
270   nChange = nChange + (1.0D0/r)*mult
271 ! Loop through pfrac
272   Do i=k,m
273     r = 1.0D0*L(i)
274     Do j=k,m
275       If(j.ne.i)Then
276         r = r * (L(i)-L(j))
277       End If
278     End Do
279     nChange = nChange + (1.0D0/r)*exp(-1.0D0*L(i)*t)*multP*mult
280   End Do
281 End Function CalcIsotopeChainF_Stable
282
283
284 Function CalcIsotopeChainG_Stable(k,mIn,t,mult,L) Result (nChange)
285 ! Vars In
286   Integer(kind=StandardInteger) :: k, mIn
287   Real(kind=DoubleReal) :: t, mult
288   Real(kind=DoubleReal), Dimension(1:100) :: L ! Lambda
289 ! Vars Out
290   Real(kind=DoubleReal) :: nChange
291 ! Private
292   Integer(kind=StandardInteger) :: m
293   Real(kind=DoubleReal) :: multP
294   Real(kind=DoubleReal) :: p, q, r, s
295   Integer(kind=StandardInteger) :: i, j
296 ! In
297   m = mIn-1
298 ! Calculate isotope amount change
299   nChange = 0.0D0

```

```

300     multP = (-1.0D0)**(m-k)
301 ! term A
302     r = 1.0D0
303     Do i=k,m
304         r = r * L(i)
305     End Do
306     nChange = nChange + (1.0D0/r)*t*mult
307 ! term B
308     p = CalcIsotopeChainC(L,k,m)
309     q = 1.0D0
310     Do i=k,m
311         q = q*L(i)*L(i)
312     End Do
313     r = (-1.0D0)*(p/q)
314     nChange = nChange + r*mult
315 ! Loop through pfrac
316     Do i=k,m
317         r = 1.0D0*L(i)*L(i)
318         Do j=k,m
319             If(j.ne.i)Then
320                 r = r * (L(i)-L(j))
321             End If
322         End Do
323         nChange = nChange + (1.0D0/r)*exp(-1.0D0*L(i)*t)*multP*mult
324     End Do
325 End Function CalcIsotopeChainG_Stable
326
327 Function CalcIsotopeChainC(L,k,m) Result (numerator)
328 ! Calculates numerator in isotope activity function
329 Implicit None ! Force declaration of all variables
330 ! Vars In
331 Real(kind=DoubleReal), Dimension(:) :: L
332 Integer(kind=StandardInteger) :: k, m
333 ! Vars Out
334 Real(kind=DoubleReal) :: numerator
335 ! Vars Private
336 Integer(kind=StandardInteger) :: i, j
337 Real(kind=DoubleReal) :: tempMult
338 numerator = 0.0D0
339 Do i=k,m
340     tempMult = 1.0D0
341     Do j=k,m
342         If(j.ne.i)Then
343             tempMult = tempMult * L(j)
344         End If
345     End Do
346     numerator = numerator + tempMult
347 End Do
348 End Function CalcIsotopeChainC

```

Bibliography

- [1] A. J. Koning D. Rochman. “Modern Nuclear Data Evaluation With The TALYS Code System”. In: *Nuclear Data Sheets* 113 (2012).
- [2] J. P. Biersack James F. Ziegler M. D. Ziegler. “SRIM - The stopping and range of ions in matter”. In: *Nuclear Instruments and Methods in Physics Research Section B* 268 (2010), pp. 1818–1823.
- [3] A. A. Sonzogni. *Nuclear Data Services*. www-nds.iaea.org. 2006. URL: <http://www-nds.iaea.org/ndspub/download-endf/ENDF-B-VII.0/decay-index.htm> (visited on 08/14/2015).
- [4] M. Herman M. B. Chadwick. “ENDF/B-VII.0: Next generation evaluated nuclear data library for nuclear science and technology.” In: *Nucl. Data Sheets* 107 (2006), p. 2931.
- [5] P. Blaise M. Coste A. Courcelle T.D. Huynh C. Jouanne P. Leconte O. Litaize S. Mengelle G. Nogure J-M. Ruggiri O. Srot J. Tommasi C. Vaglio J-F. Vidal A. Santamarina D. Bernard. *The JEFF-3.1.1 Nuclear Data Library*. 2009. ISBN: 978-92-64-99074-6.
- [6] H. Stehfest. “Algorithm 368: Numerical Inversion of Laplace Transform”. In: *Communications of the ACM* 13 (1970), pp. 47–49.
- [7] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing In Science & Engineering* 9 (2007), pp. 90–95.

Appendix B

Simulated Annealing vs Genetic Algorithm

B.1 Introduction

Simulated annealing and genetic algorithms are used to find the global minimum of a given problem.

B.2 Algorithm Comparison

Brief comparisons fits were run on the same fitting problem for both algorithms. For each test, the optimisation time was limited, and the rss for each fit attempt at the end of that time period was recorded. The fitting problem was a double exponential $f(x) = a\exp(bx) + c\exp(dx)$ with four parameters, and the initial parameters were the same for both algorithms.

B.2.1 Genetic Algorithm, Pop 32 vs Simulated Annealing

B.2.2 Genetic Algorithm, Pop 64 vs Simulated Annealing

B.2.3 Genetic Algorithm, Pop 512 vs Simulated Annealing

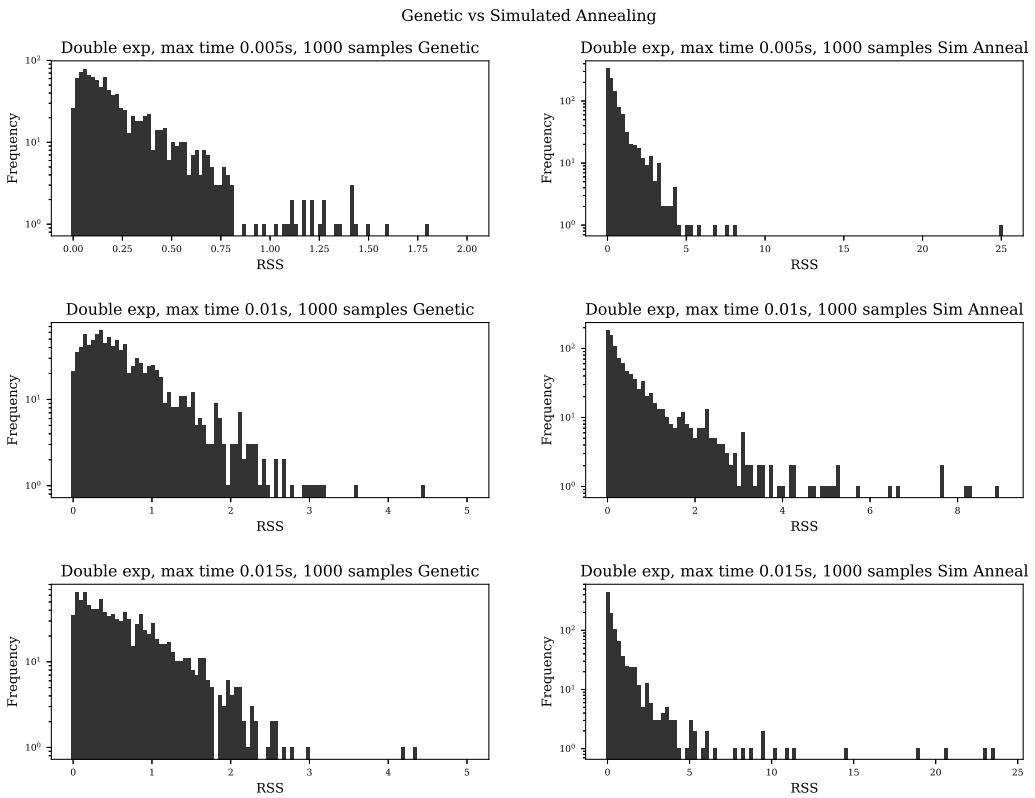


Figure B.1: Genetic Algorithm Population Size 32 vs Simulated Annealing

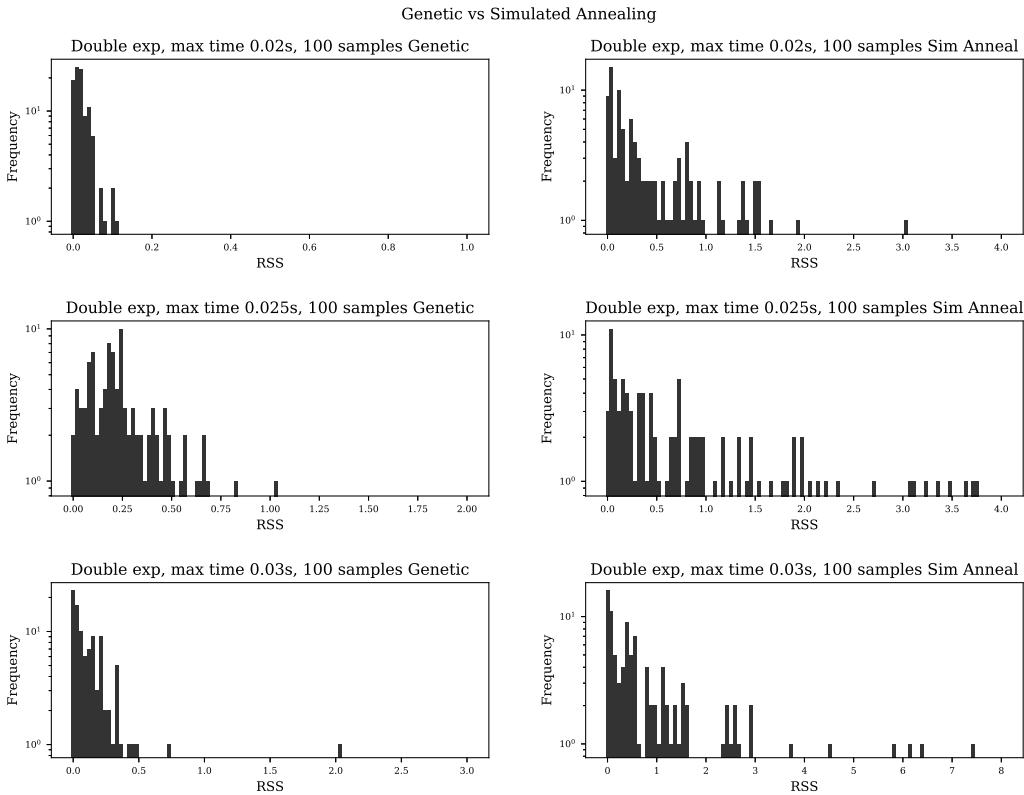


Figure B.2: Genetic Algorithm Population Size 32 vs Simulated Annealing

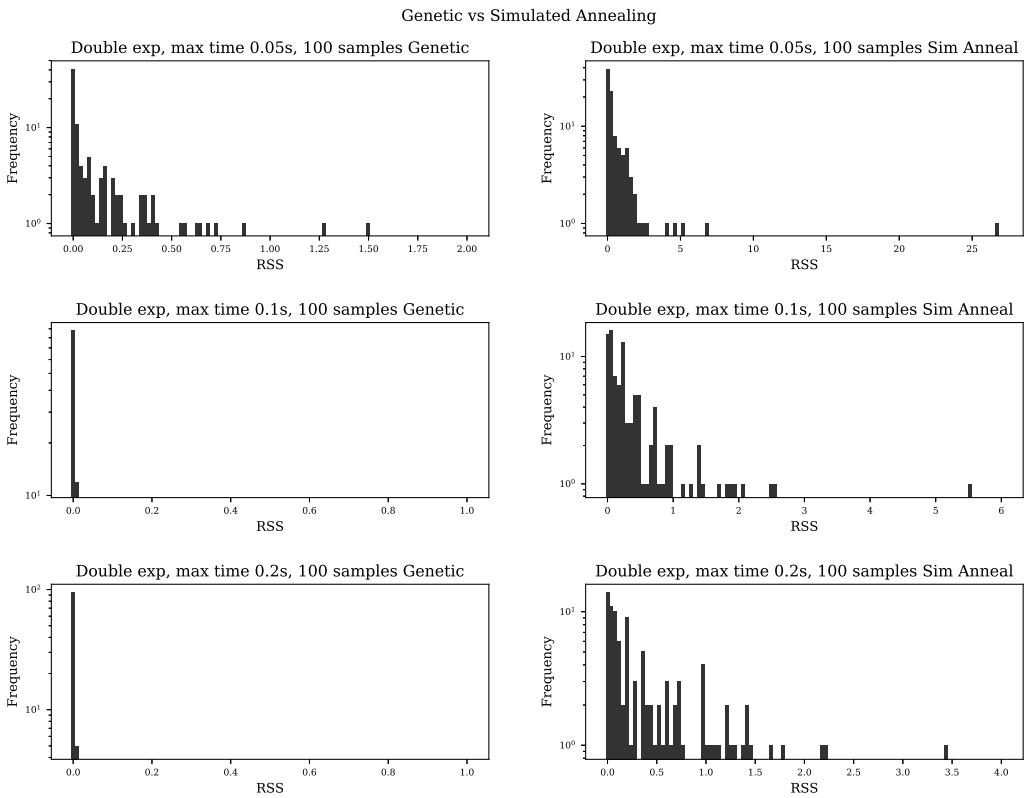


Figure B.3: Genetic Algorithm Population Size 32 vs Simulated Annealing

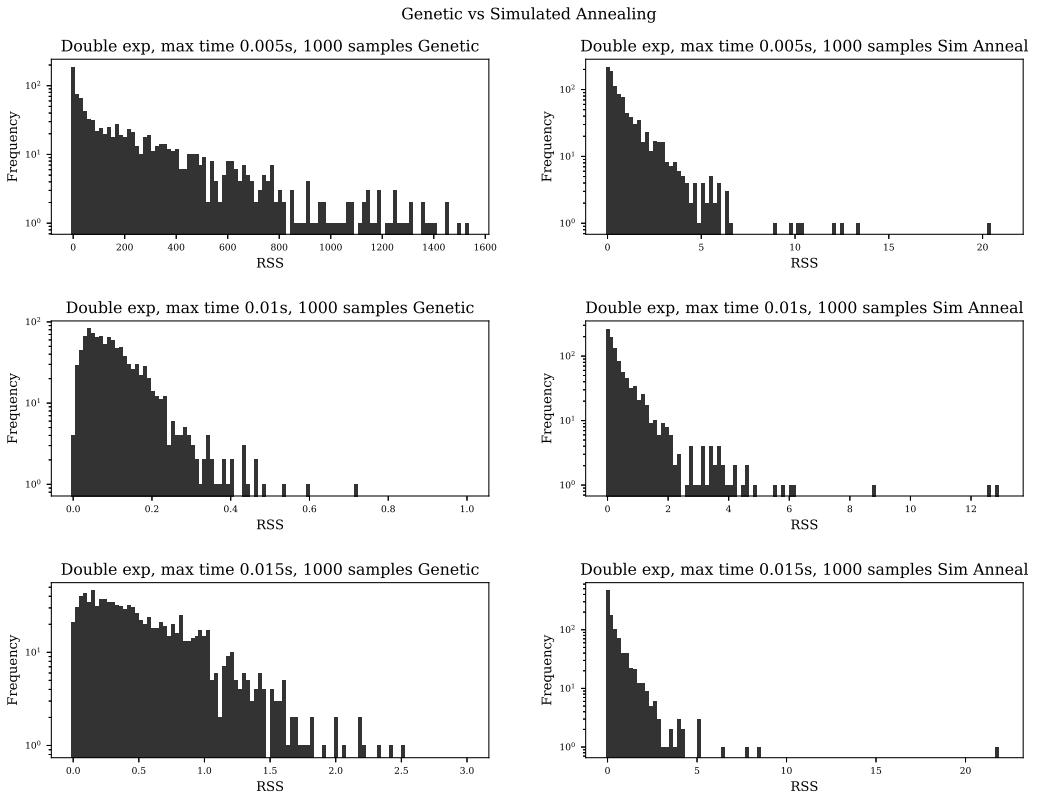


Figure B.4: Genetic Algorithm Population Size 64 vs Simulated Annealing

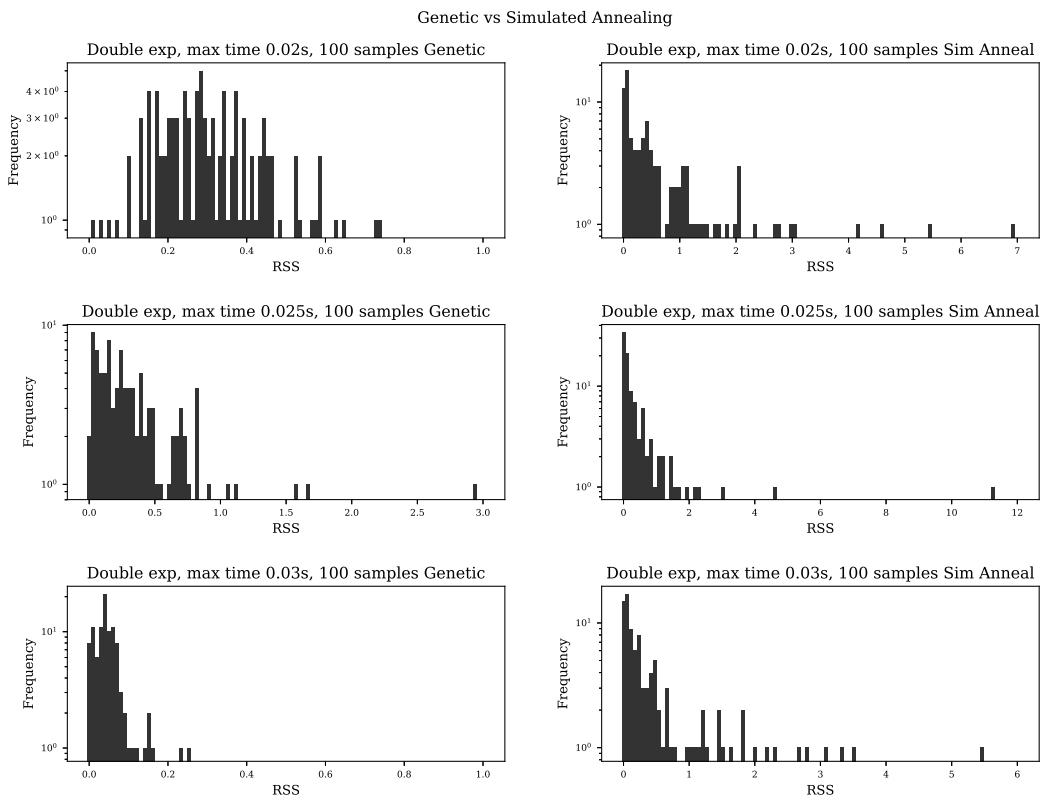


Figure B.5: Genetic Algorithm Population Size 64 vs Simulated Annealing

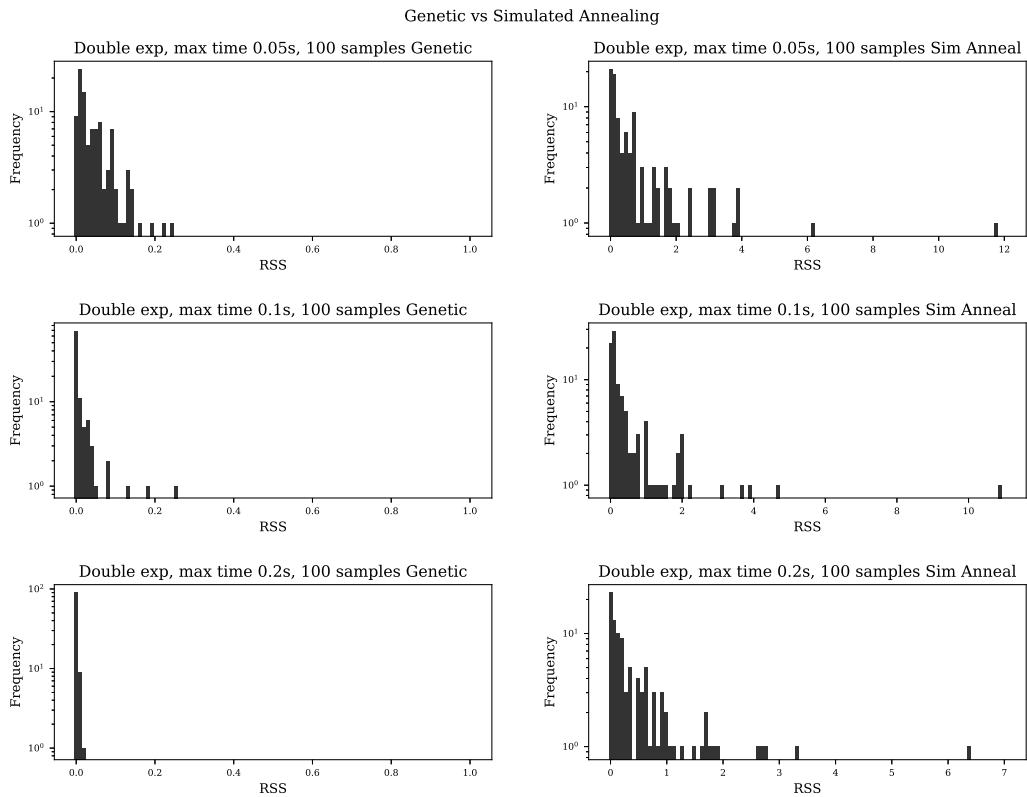


Figure B.6: Genetic Algorithm Population Size 64 vs Simulated Annealing

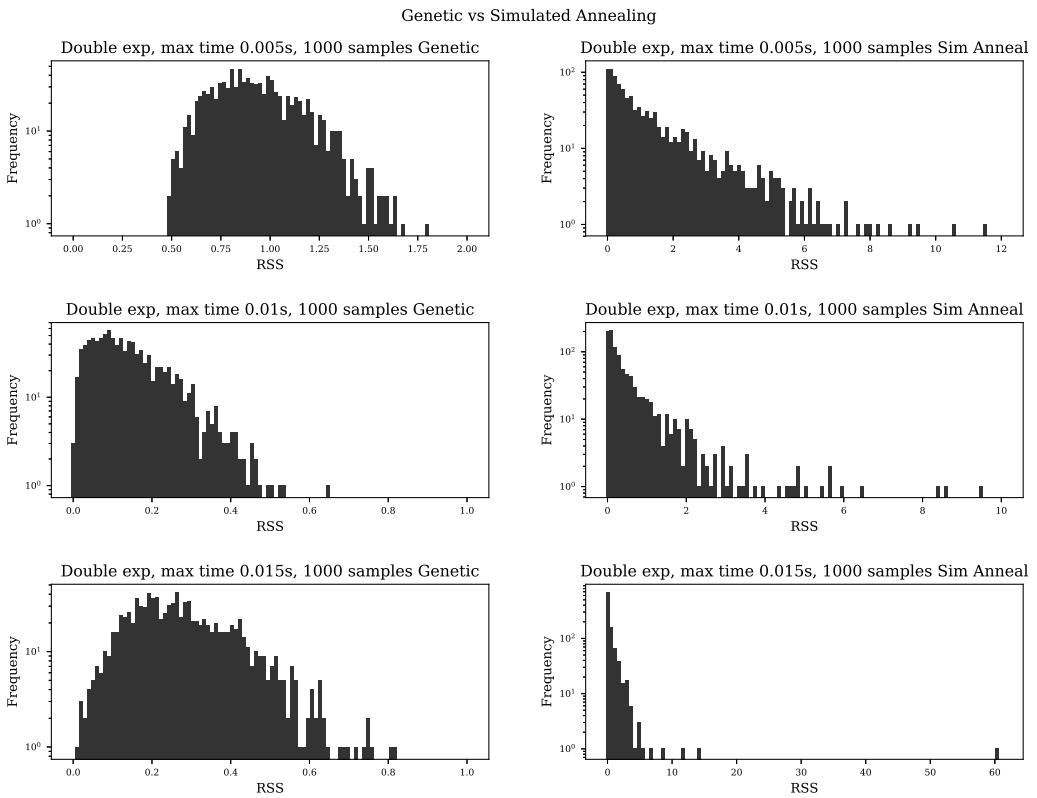


Figure B.7: Genetic Algorithm Population Size 512 vs Simulated Annealing

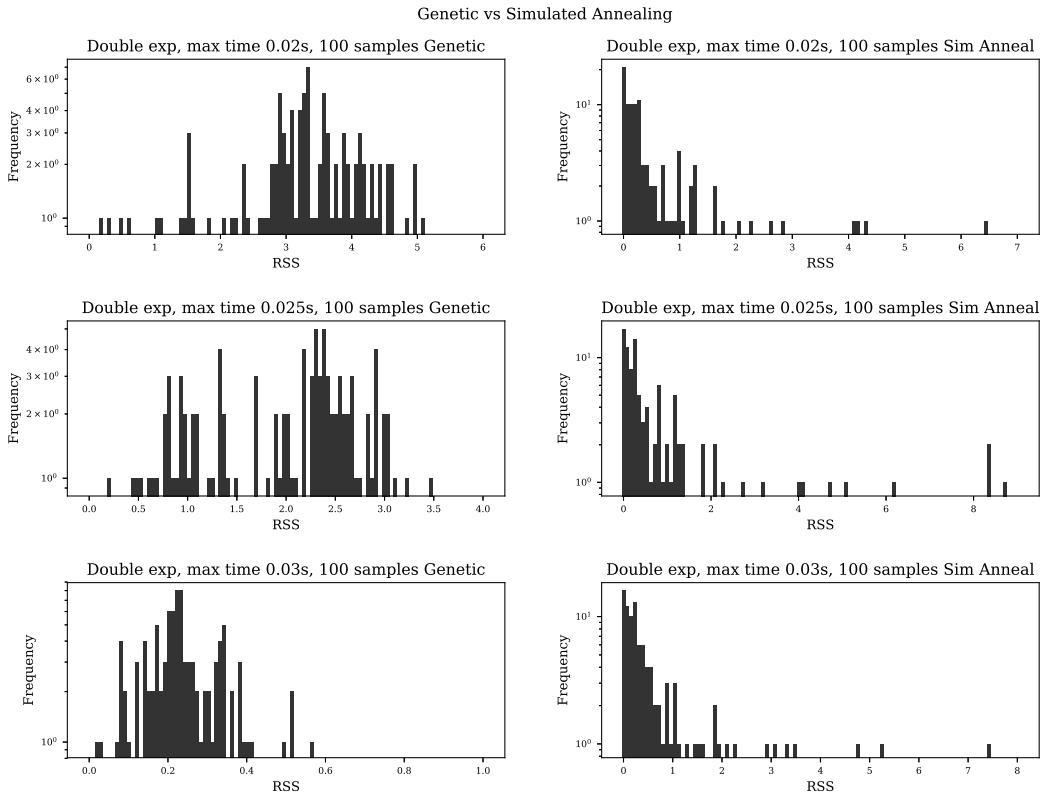


Figure B.8: Genetic Algorithm Population Size 512 vs Simulated Annealing

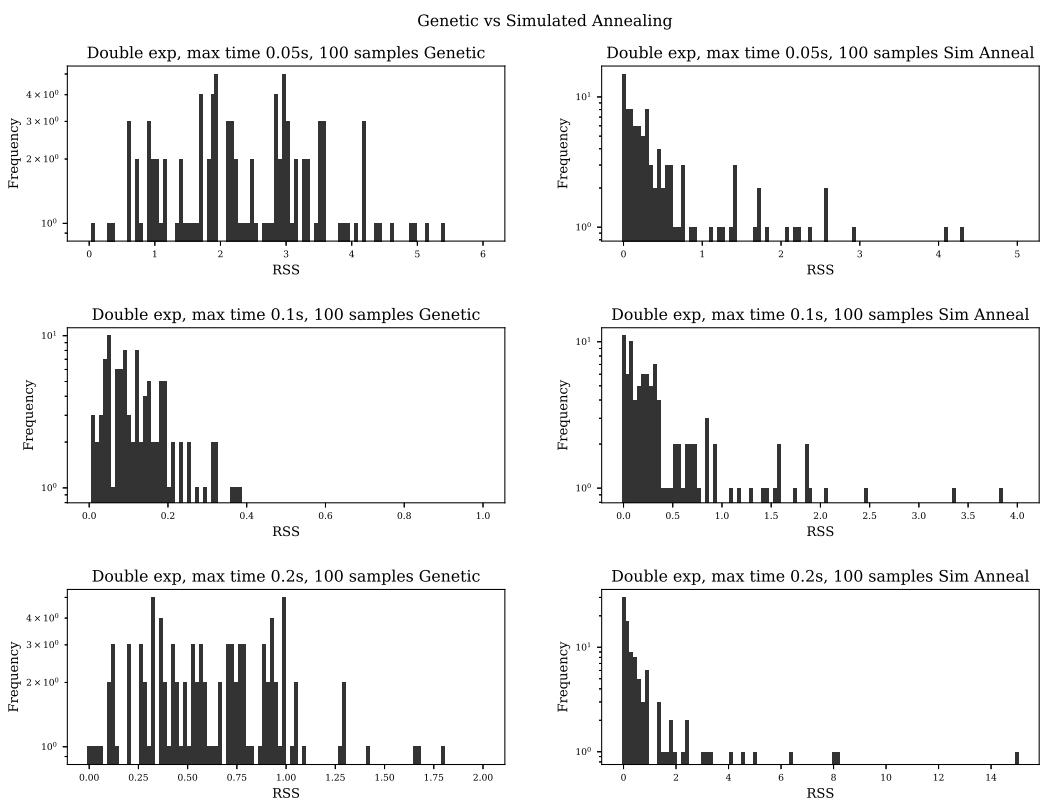


Figure B.9: Genetic Algorithm Population Size 512 vs Simulated Annealing

Bibliography

- [1] H. Khartabil. *Gen-4 Reactors*. 2013. URL: <http://www.gen-4.org/GIF/About/documents/25-Session2-3-Khartabil.pdf> (visited on 05/20/2013).
- [2] MIT Open Courseware. *Engineering of Nuclear Systems - Lecture 6A*. 2013. URL: http://ocw.mit.edu/courses/nuclear-engineering/22-06-engineering-of-nuclear-systems-fall-2010/lectures-and-readings/MIT22_06F10_lec06a.pdf (visited on 05/20/2013).
- [3] MIT Open Courseware. *Engineering of Nuclear Systems - Lecture 6B*. 2013. URL: http://ocw.mit.edu/courses/nuclear-engineering/22-06-engineering-of-nuclear-systems-fall-2010/lectures-and-readings/MIT22_06F10_lec06b.pdf (visited on 05/20/2013).
- [4] E. Malambu A. Orden D. Struwe P. Agostini S. Monti A. Alemberti J. Carlsson. “European lead fast reactor - ELSY”. In: *Nuclear Engineering and Design* 241 (2011), pp. 3470–3480.
- [5] N J Carron. *An Introduction to the Passage of Energetic Particles through Matter*. 2007. ISBN: 978-0-7503-0935-6.
- [6] G. Verdu J. Rodenas. “Analysis of nuclear reactions to determine the radionuclides generated and its activity in various devices”. In: *Radiation Physics and Chemistry* 167 (2020), pp. 1–6.
- [7] World Nuclear. 2018. URL: <http://www.world-nuclear.org/information-library/non-power-nuclear-applications/radioisotopes-research/research-reactors.aspx> (visited on 11/08/2018).
- [8] IAEA. 2018. URL: <https://nucleus.iaea.org/RRDB/RR/ReactorSearch.aspx> (visited on 11/08/2018).
- [9] ORNL. 2018. URL: <https://neutrons.ornl.gov/hfir/parameters> (visited on 11/10/2018).
- [10] P. Blaise M. Coste A. Courcelle T.D. Huynh C. Jouanne P. Leconte O. Litaize S. Mengelle G. Nogure J-M. Ruggiri O. Srot J. Tommasi C. Vaglio J-F. Vidal A. Santamarina D. Bernard. *The JEFF-3.1.1 Nuclear Data Library*. 2009. ISBN: 978-92-64-99074-6.
- [11] J. P. Biersack James F. Ziegler M. D. Ziegler. “SRIM - The stopping and range of ions in matter”. In: *Nuclear Instruments and Methods in Physics Research Section B* 268 (2010), pp. 1818–1823.
- [12] Graeme J. Ackland. “Two-band second moment model for transition metals and alloys”. In: *Condensed Materials* (2005).
- [13] J. B. Adams F. Ercolessi. “Interatomic Potentials from First-Principles Calculations: the Force-Matching Method”. In: *Europhys. Lett.* 26 (8 1994), pp. 583–588.
- [14] A. Cadier T. Fujita H. W. Sheng M. J. Kramer and M. W. Chen. “Highly Optimized Embedded-Atom-Method Potentials for Fourteen FCC Metals”. In: *Europhys. Lett.* 26 (8 1994), pp. 583–588.
- [15] F. D. Murnaghan. “The Compressibility of Media Under Extreme Pressures”. In: (1944).
- [16] F. Birch. “Finite elastic strain of cubic crystals”. In: (1947).
- [17] Michael J. Mehl. “Pressure dependence of the elastic moduli in aluminium-rich Al-Li compounds”. In: *Physical Review B* 47 (5 1993), pp. 2493–2500.

- [18] P. A. Korzhavyi B. Johansson P. Ravindran Lars Fast. “Density functional theory for calculation of elastic properties of orthorhombic crystals: applications to TiSi₂”. In: *Journal of Applied Physics* 84 (9 1998), pp. 4891–4904.
- [19] H. L. Marcus M. E. Fine L. D. Brown. “Elastic Constants Versus Melting Temperature in Metals”. In: *Scripta Metallurgica* 18 (1984), pp. 951–956.
- [20] Jiechao Cui Baoqin Fu Min Li Qing Hou. “Molecular dynamics simulations of tungsten bombardments on tungsten”. In: *Nuclear Inst. and Methods in Physics Research B* (2018), In Press.
- [21] H. Stehfest. “Algorithm 368: Numerical Inversion of Laplace Transform”. In: *Communications of the ACM* 13 (1970), pp. 47–49.
- [22] G. Dimitri Ngantso A. T. Raji A. L. Okana-Lomanga B. R. Malonda-Boungou. “Adsorption of Br₂ molecule on the Fe/W(110) substrate: Energetics, electronic and magnetic properties”. In: *Computational Condensed Matter* 23 (2020).
- [23] M. Almasi-Kashi A. E. Rezaee. “The influence of point defects on Na diffusion in black phosphorene: first principles study”. In: *Journal Pre-proof* () .
- [24] S. P. Sanyal N. Acharya. “Structural phase transition, electronic and superconducting properties of ScBi and YBi”. In: *Solid State Communications* 266 (2017), pp. 39–45.
- [25] R. K. Siripurapu A. Choudhary L. Malakkal. “First principles calculations of hydrogen storage on Cu and Pd-decorated graphene”. In: *International Journal of Hydrogen Energy* 41 (2016), pp. 17652–17656.
- [26] G. A. Adebayo O. E. Osafole P. O. Adebambo. “Elastic constants and observed ferromagnetism in inverse Heusler alloy Ti₂CoAs using kjpaw pseudopotentials: A first-principles approach”. In: *Journal of Alloys and Compounds* 722 (2017), pp. 207–211.